Centre de recherche sur l'environnement alpin (CREALP)

HydroCosmos SA

RS MINERVE

RS MINERVE - Technical Manual

Version 2.25, April 2020

ISSN 2673-2661

For Software version 2.8.7

Centre de recherche sur l'environnement alpin (CREALP)

HydroCosmos SA

Authors:

Javier GARCIA HERNANDEZ	CREALP
Alain FOEHN	CREALP
Javier FLUIXA SANMARTIN	CREALP
Bastien ROQUIER	CREALP
Tristan BRAUCHLI	CREALP
Javier PAREDES ARQUIOLA	UPV
Giovanni DE CESARE	EPFL

Please refer this technical manual as:

García Hernández, J., Foehn, A., Fluixá-Sanmartín, J., Roquier, B., Brauchli, T., Paredes Arquiola, J. and De Cesare, G. (2020). *RS MINERVE – Technical manual*, v2.25. Ed. CREALP, Switzerland. ISSN 2673-2661.

Index

Foreword

RS MINERVE is a software for the simulation of free surface run-off flow formation and propagation. It models complex hydrological and hydraulic networks according to a semi-distributed conceptual scheme. In addition to particular hydrological processes such as snowmelt, glacier melt, surface and underground flow, hydraulic control elements (e.g. gates, spillways, diversions, junctions, turbines and pumps) are also included.

The global analysis of a hydrologic-hydraulic network is essential in numerous decision-making situations such as the management or planning of water resources, the optimization of hydropower plant operations, the design and regulation of spillways or the development of appropriate flood protection concepts. RS MINERVE makes such analyses accessible to a broad public through its user-friendly interface and its valuable possibilities. In addition, thanks to its modular framework, the software can be developed and adapted to specific needs or issues.

RS MINERVE contains different hydrological models for **rainfall-runoff, such as GSM**, **SOCONT**, **SAC-SMA**, **GR4J** and **HBV**. The combination of hydraulic structure models (reservoirs, turbines, spillways,…) can also reproduce complex hydropower schemes. In addition, a **hydropower model** computes the net height and the linear pressure losses, providing energy production values and total income based on the turbine performance and on the sale price of energy. A **consumption model** calculates water deficits for consumptive uses of cities, industries and/or agriculture. A **structure efficiency model** computes discharge losses in a structure such a canal or a pipe by considering a simple efficiency coefficient.

The **Expert module**, specifically created for research or complex studies, enables in-depth evaluation of hydrologic and hydraulic results. **Time-slice simulation** facilitates the analysis of large data sets without overloading the computer memory. **Scenario simulation** introduced the possibility of simulating multiple weather scenarios or several sets of parameters and initial conditions to study the variability and sensitivity of the model results. The **automatic calibration** with different algorithms, such as the SCE-UA, calculates the best set of hydrological parameters depending on a user-defined objective function.

RS MINERVE program is **freely distributed** to interested users. Several projects and theses have used and are using this program for study basins in Switzerland, Spain, Peru, Brazil France and Nepal. In addition to the research center **CREALP** and the engineering office **HydroCosmos SA**, which currently develop RS MINERVE, two universities (**Ecole Polytechnique Fédérale de Lausanne** and **Universitat Politècnica de València**) collaborate to improve RS MINERVE and use it to support postgraduate courses in Civil Engineering and Environmental Sciences.

Chapter 1. Introduction

The Routing System II program was developed at the Laboratory of Hydraulic Constructions (LCH) at the Ecole Polytechnique Fédérale de Lausanne (EPFL) (Dubois et al., 2000; García Hernández et al., 2007).

The program presented hereafter, RS MINERVE, is based on the same concept than Routing System II. RS MINERVE is developed by the CREALP and HydroCosmos SA with the collaboration of the Laboratory of Hydraulic Constructions (LCH) at the Ecole Polytechnique Fédérale de Lausanne (EPFL) and the Universitat Politècnica de València (UPV).

1.1. Manual's structure

The presented Technical manual is organised in eight chapters:

1. Introduction
2. Hydrological and hydraulic models description
3. Performance indicators
4. Calibration algorithms
5. Visual basic scripts
6. Files formats
7. Database formats
8. GIS formats

For the RS MINERVE software utilisation, the reader can also use the RS MINERVE User's Manual (Foehn et al., 2020).

Chapter 2. Hydrological and hydraulics models description

This chapter presents the hydrological objects existing in RS MINERVE software.

2.1. List of objects

The parameters or required data (such as paired data) of all objects are presented hereafter. Not presented objects do not require data or require data from another object or from database.

- Hydrology:
  - Virtual Station (see 2.2)
  - Snow-SD model (see 2.3)
  - Runoff (SWMM) model (see 2.4)
  - GSM model (see 2.5)
  - SOCONT model (see 2.6)
  - HBV model (see 2.7)
  - GR4J model (see 2.8)
  - SAC (SACRAMENTO-SOIL MOISTURE ACCOUNT) model (see 2.9)
- Rivers
  - Channel routing description (see 2.10)
- Standard:
  - Junction (object without parameters or paired data)
  - Time Series (see 2.11)
  - Source (object without parameters or paired data, but only a link to the database)
  - Comparator (see 2.20)
  - Sub-model (object without any parameters or paired data)
  - Group Interface (object without any parameters or paired data)
- Structures:
  - Reservoir (see 2.12)
  - Level-Discharge relation HQ (see 2.13)
  - Turbine (see 2.14)
  - TurbineDB (object linked to the database; see 2.14)
  - Hydropower (see 2.15)
  - Diversion (see 2.16)
  - Consumer (see 2.17)
  - Structure Efficiency (see 2.18)
- Regulation objects:
  - Planner (see 2.19)

2.2. Virtual Station

The object « V-Station » (which is associated with the coordinates X, Y, Z) allows the spatial distribution of the meteorological variables (precipitation, temperature, ETP) from available measures or estimations of a database, with spatial reference in metric coordinates.

Table 1 List of parameters and initial conditions for the virtual station

Object	Name	Units	Description	Regular Range
Station	X, Y, Z	-	Coordinates of the virtual station	-
	Search Radius	m	Search radius of the virtual stations	>0
	No. min. of stations	-	Minimal number of stations used for interpolation (higher priority than “Search Radius”)	≥1
	Gradient P	1/m	Precipitation gradient	- [a]
	Gradient T	°C/m	Temperature gradient	-0.007 to -0.004
	Gradient ETP	1/m	Evapotranspiration gradient	- [a]
	Coeff P	-	Multiplying correction coefficient	0.5 to 2
	Coeff T	°C	Adding correction coefficient	-2 to 2
	Coeff ETP	-	Multiplying correction coefficient	0.5 to 2

[a] The precipitation and evapotranspiration gradients are function of the local conditions. Their regular ranges have to be estimated for each studied case.

The method chosen for the spatial distribution of the precipitation, the temperature and the ETP corresponds to the Thiessen and Shepard methods. The first method, Thiessen, searches the nearest meteorological station for each meteorological variable. The second one, Shepard, searches i stations being in a search radius and calculates the meteorological variable depending on inverse distance weighting.

Thiessen interpolation

The evaluation of a variable in a virtual station s from n meteorological stations is obtained by searching the nearest meteorological station k of the database to the virtual point s (normally referring to the gravity centre of a sub-catchment).

This method has been extended to take into account the evolution of certain meteorological variables as a function of the altitude. Thus, gradients and coefficients for precipitation, potential evapotranspiration or temperature is also included in the method for obtaining the final value at virtual station s, as presented in equations A.1 to A.3.

$$P_s = CoeffP_s \cdot \left(1 + GradP_s \cdot (z_s - z_k)\right) \cdot P_k \tag{A.1}$$

$$T_s = CoeffT_s + GradT_s \cdot (z_s - z_k) + T_k \tag{A.2}$$

$$ETP_s = CoeffETP_s \cdot \left(1 + GradETP_s \cdot (z_s - z_k)\right) \cdot ETP_k \tag{A.3}$$

with P_s: value of the precipitation in the virtual station s [I.U.]; T_s: value of the temperature in the virtual station s [I.U.]; ETP_s: value of the potential evapotranspiration in the virtual station s [I.U.]; P_k: value of the precipitation in the meteorological station k [I.U.]; T_k: value of the temperature in the meteorological station ki [I.U.]; ETP_k: value of the potential evapotranspiration in the meteorological station k [I.U.]; $CoeffP_s$: precipitation coefficient [-]; $CoeffT_s$: temperature coefficient [°C]; $CoeffETP_s$:

potential evapotranspiration coefficient [-];GradP$_s$: precipitation gradient (dP$_s$/dz) [1/m]; GradT$_s$: temperature gradient (dT$_s$/dz) [°C/m]; GradETP$_s$: potential evapotranspiration gradient (dETP$_s$/dz) [1/m]; z$_s$: altitude of the virtual station s [m a.s.l.]; z$_k$: position of the meteorological station k of the database [m a.s.l.].

In that case, the parameters rs (search radius) and No. min. of stations (minimal number of stations) are not used, since only the nearest meteorological station is used.

Shepard interpolation

The evaluation of a variable in a virtual station s from n meteorological stations localized at i=1,2,…,n is obtained by weighting according to the inverse square distance d$_{i,s}$ between the meteorological station i of the database and the virtual station s.

$$d_{\mathrm{i},s} = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2} \qquad \text{(A.4)}$$

with x$_i$, y$_i$: position of the meteorological station i of the database [m]; x$_s$, y$_s$: position of the virtual station s [m]; d$_{i,s}$: distance between the meteorological station i and the virtual station s [m].

The n meteorological stations for the spatial interpolation in the virtual station s are determined automatically respecting equation A.5. Hence, the number n of meteorological stations is variable for every pair (s, r$_s$). Nevertheless, a minimal number of stations used for interpolation can be fixed by the user whit the corresponding parameter.

$$d_{\mathrm{i},s} \leq r_s \qquad \text{(A.5)}$$

with rs: search radius of meteorological stations [m].

The Shepard method (1968) has been also extended to take into account the evolution of the meteorological variables as a function of the altitude. Gradients and coefficients for precipitation, potential evapotranspiration or temperature are also included in the method for obtaining the final value at virtual station s, as presented in equations A.6 to A.8.

$$P_s = CoeffP_s \cdot \left(\frac{\sum_{i=1}^{n} \left[\left(1 + GradP_s \cdot (z_s - z_i)\right) \cdot P_i \cdot \frac{1}{{d_{\mathrm{i,s}}}^2} \right]}{\sum_{i=1}^{n} \frac{1}{{d_{\mathrm{i,s}}}^2}} \right) \qquad \text{(A.6)}$$

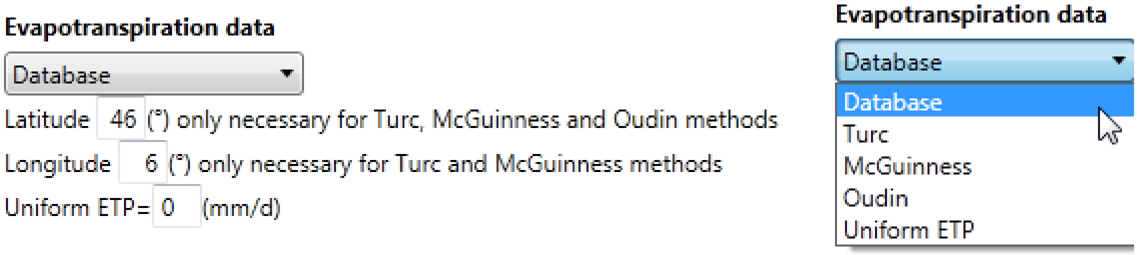
$$T_s = CoeffT_s + \left(\frac{\sum_{i=1}^{n} \left[\left(GradT_s \cdot (z_s - z_i) + T_i\right) \cdot \frac{1}{{d_{\mathrm{i,s}}}^2} \right]}{\sum_{i=1}^{n} \frac{1}{{d_{\mathrm{i,s}}}^2}} \right) \qquad \text{(A.7)}$$

$$ETP_s = CoeffETP_s \cdot \left(\frac{\sum_{i=1}^{n} \left[\left(1 + GradETP_s \cdot (z_s - z_i)\right) \cdot ETP_i \cdot \frac{1}{{d_{\mathrm{i,s}}}^2} \right]}{\sum_{i=1}^{n} \frac{1}{{d_{\mathrm{i,s}}}^2}} \right) \qquad \text{(A.8)}$$

with P$_s$: value of the precipitation in the virtual station s [I.U.]; T$_s$: value of the temperature in the virtual station s [I.U.]; ETP$_s$: value of the potential evapotranspiration in the virtual station s [I.U.]; P$_i$: value of the precipitation in the meteorological station i [I.U.]; T$_i$: value of the temperature in the meteorological station i [I.U.]; ETP$_i$: value of the potential evapotranspiration in the meteorological station i [I.U.]; $CoeffP_s$: precipitation coefficient [-]; $CoeffT_s$: temperature coefficient [°C]; $CoeffETP_s$: potential evapotranspiration coefficient [-]; $GradP_s$: precipitation gradient (dP_s/dz) [1/m]; $GradT_s$: temperature gradient (dT_s/dz) [°C/m]; $GradETP_s$: potential evapotranspiration gradient ($dETP_s/dz$) [1/m]; z_s: altitude of the virtual station s [m a.s.l.]; z_i: position of the meteorological station i of the database [m a.s.l.].

Complementary calculations for the Potential Evapotranspiration (ETP)

If no ETP values are available in the database, RS MINERVE offers also the possibility of calculating the ETP from different methods directly at virtual station. These methods can be selected in the RS MINERVE Settings, on the Evapotranspiration frame (Figure 1).



Evapotranspiration data

Database

Latitude 46 (°) only necessary for Turc, McGuinness and Oudin methods

Longitude 6 (°) only necessary for Turc and McGuinness methods

Uniform ETP= 0 (mm/d)

Figure 1 Selection of the ETP calculation

The available methods are presented hereafter in detail.

a) Turc

The potential evapotranspiration proposed by Turc (1955, 1961) is presented in equation A.9:

$$\begin{aligned} ETP &= CoeffETP \cdot \mathrm{K} \cdot \frac{T}{T+15} \cdot (R_g + 50) && if\ \mathrm{T}{>}0 \\ ETP &= 0 && if\ T \leq 0 \end{aligned} \qquad \text{(A.9)}$$

with ETP : potential evapotranspiration [mm/month] ; T : air temperature [°C] ; Rg : global radiation [cal/cm^2/day] ; K : constant [-].

The constant K value is:

$$\begin{aligned} K &= 0.4 && if\ Month \neq February \\ K &= 0.37 && if\ Month = February \end{aligned} \qquad \text{(A.10)}$$

R_g value is a location dependent (latitude and longitude) monthly average of the global radiation.

The global radiation R_g is obtained in [kWh/m^2/day] from the *Global horizontal radiation* dataset provided by the Surface meteorological and Solar Energy (SSE) web portal, sponsored by the NASA's Applied Science Program (http://eosweb.larc.nasa.gov/sse). This data comes as a grid (latitude and longitude) and is composed of monthly averaged values.

R_g data takes into account 22 year monthly average (July 1983 - June 2005). The latitude and the longitude values indicate the lower left corner of a 1x1 degree region. Negative values are south and west; positive values are north and east. Boundaries of the -90/-180 region are -90 to -89 (south) and -180 to -179 (west). The last region, 89/180, is bounded by 89 to 90 (north) and 179 to 180 (east). The mid-point of the region is +0.5 added to the latitude/longitude value. These data are regional averages, not point data.

If the user introduces decimals to the latitude/longitude values, the RS MINERVE program calculates the nearest integer value for R_g calculations.

b) McGuinness

McGuinness et Bordne (1972) proposes next ETP calculation:

$$\begin{aligned} &ETP = CoeffETP \cdot \frac{R_g}{\lambda + \rho} \cdot \frac{T_a + 5}{68} && if \ \mathrm{T} > -5 \\ &ETP = 0 && if \ T \leq -5 \end{aligned} \tag{A.11}$$

with ETP: potential evapotranspiration [m/d]; R_g: global radiation [MJ/m^2/day]; T_a : air temperature [C]; ρ: water density, constant value of 1’000 [kg/m^3] ; λ: latent heat of vaporization, constant value of 2.26 [MJ/kg].

R_g value is a location dependent (latitude and longitude) monthly average of the global radiation.

The global radiation R_g is obtained in [kWh/m^2/day] from the *Global horizontal radiation* dataset provided by the Surface meteorological and Solar Energy (SSE) web portal, sponsored by the NASA’s Applied Science Program (http://eosweb.larc.nasa.gov/sse). This data comes as a grid (latitude and longitude) and is composed of monthly averaged values.

R_g data takes into account 22 year monthly average (July 1983 - June 2005). The latitude and the longitude values indicate the lower left corner of a 1x1 degree region. Negative values are south and west; positive values are north and east. Boundaries of the -90/-180 region are -90 to -89 (south) and -180 to -179 (west). The last region, 89/180, is bounded by 89 to 90 (north) and 179 to 180 (east). The mid-point of the region is +0.5 added to the latitude/longitude value. These data are regional averages; not point data.

If the user introduces decimals to the latitude/longitude values, the RS MINERVE program calculates the nearest integer value for R_g calculations.

c) Oudin

Oudin (2004) proposes following equation for the calculation of ETP:

$$\begin{aligned} &ETP = CoeffETP \cdot \frac{R_e}{\lambda \cdot \rho} \cdot \frac{T+5}{100} && if\ T > -5 \\ &ETP = 0 && if\ T \leq -5 \end{aligned} \qquad \text{(A.12)}$$

with ETP: potential evapotranspiration [m/d]; R_e : extra-terrestrial radiation [MJ m^{-2} d^{-1}] ; T: Air temperature [°C] ; ρ: water density, constant value of 1'000 [kg/m^3] ; λ: latent heat of vaporization, constant value of 2.26 [MJ/kg].

Oudin method coefficients (5 and 100) were optimized for the hydrological modelling, on the basis of a study realized on many worldwide watersheds (Oudin, 2004).

Latitude are only necessary for obtaining R_e values

The extra-terrestrial radiation R_e is calculated as follows:

$$R_e = 37.6 \cdot dr \cdot (\omega \cdot \sin(\psi) \cdot \sin(\delta) + \sin(\omega) \cdot \cos(\psi) \cdot \cos(\delta)) \qquad \text{(A.13)}$$

$$dr = 1 + 0.033 \cdot \cos(\frac{2 \cdot \pi \cdot J_d}{365}) \qquad \text{(A.14)}$$

$$\omega = \arccos(-\tan(\psi) \cdot \tan(\delta)) \qquad \text{(A.15)}$$

$$\delta = 0.409 \cdot \sin(\frac{2 \cdot \pi \cdot J}{365} - 1.39) \qquad \text{(A.16)}$$

$$\begin{aligned} &J_d = 275 \cdot \frac{month}{9} - 30 + Dm && if\quad month < 3 \\ &J_d = 275 \cdot \frac{month}{9} - 31 + Dm && if\quad month \geq 3\ and\ leap\ year = true \\ &J_d = 275 \cdot \frac{month}{9} - 32 + Dm && if\quad month \geq 3\ and\ leap\ year = false \end{aligned} \qquad \text{(A.17)}$$

with dr: relative distance Sun-Earth [-] ; δ : solar declination [rad] ; J_d : Julian day [-] ; ψ : latitude, negative in the south hemisphere [rad]; ω: hour angle of the sun [rad]; *month* : month of the year, 1 to 12 [-] ; Dm :day of the month [-].

d) Uniform ETP

The user can also set a uniform ETP for the whole simulation period and for the entire basin.

$$ETP = CoeffETP \cdot X \qquad \text{(A.18)}$$

with X: uniform ETP [mm/d].

2.3. Snow-SD model description

The Snow-SD (Snow model with a Seasonal Degree-day factor) model (Figure 2), inspired by Schaefli et al. 2005 and Hamdi et al. 2005, is composed of two sub-models. They simulate the transient evolution of the snow pack (accumulation and melt) as a function of the temperature (T) and precipitation (P) and produce an equivalent precipitation (P_{eq}) which can be used as an input variable for the SAC-SMA or GR4J model.

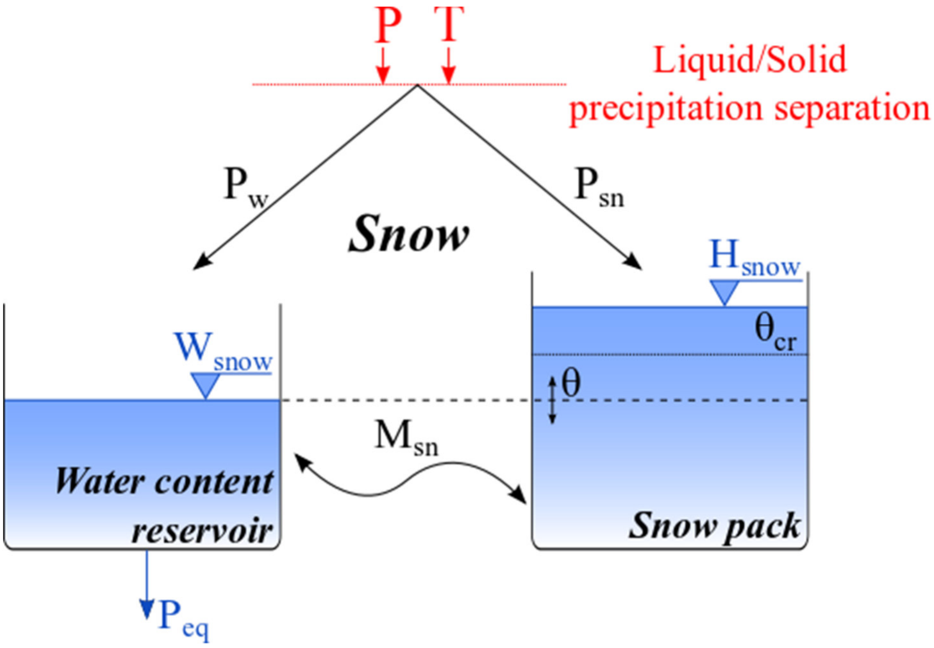


Figure 2 Snow-SD model

Table 2 List of parameters and initial conditions for the Snow-SD model

Object	Name	Units	Decription	Regular range
Snow-SD	S	mm/°C/d	Reference degree-day snowmelt coefficient	0.5 to 20
	SInt	mm/°C/d	Degree-day snowmelt interval	0 to 4
	SMin	mm/°C/d	Minimal degree-day snowmelt coefficient	≥0
	SPh	d	Phase shift of the sinusoidal function	1 to 365
	ThetaCri	-	Critical relative water content of the snow pack	0.1
	bp	d/mm	Melt coefficient due to liquid precipitation	0.0125
	Tcp1	°C	Minimum critical temperature for liquid precipitation	0
	Tcp2	°C	Maximum critical temperature for solid precipitation	4
	Tcf	°C	Critical snowmelt temperature	0
	CFR	-	Refreezing coefficient	0 to 1
	SWEIni	m	Initial snow water equivalent height	-
	ThetaIni	-	Initial relative water content in the snow pack	-

In a first step, precipitation is separated into solid (P_{sn}) and liquid precipitation (P_w) as a function of the temperature (*equations* B.*1* to B.*3*):

$$P_w = \alpha \cdot P \tag{B.1}$$

$$P_{sn} = (1-\alpha)\cdot P \tag{B.2}$$

$$\begin{aligned} &\alpha = 0 && if \quad T \le T_{cp1} \\ &\alpha = (T-T_{cp1})/(T_{cp2}-T_{cp1}) && if \quad T_{cp1} < T < T_{cp2} \\ &\alpha = 1 && if \quad T \ge T_{cp2} \end{aligned} \tag{B.3}$$

with P_w: liquid precipitation [L/T]; α: separation factor; P: precipitation [L/T]; P_{sn}: solid precipitation [L/T]; T: *te*mperature [°C]; T_{cp1}: minimum critical temperature for liquid precipitation [°C]; T_{cp2}: maximum critical temperature for solid precipitation [°C].

If the observed temperature is lower than T_{cp1}, only solid precipitation is generated. If the temperature is higher than T_{cp2}, only liquid precipitation (P_w) is present. If the observed temperature is between these two critical values, both phases (liquid and solid) are produced. Solid precipitation (P_{sn}) is used as input for the snow pack, varying its content as a function of melt or freezing. The snowmelt calculation is performed as follows, using a time-varying degree-day snowmelt coefficient with a lower bound as presented in Figure 3 (Griessinger et al. 2016, Magnusson et al. 2014, Slater and Clark 2005):

$$\mathrm{S}' = \max\left(S_{Min}\,;\, \mathrm{S} + \frac{S_{Int}}{2}\sin\left(2\pi\frac{\mathrm{n}-S_{Ph}}{365}\right)\right) \tag{B.4}$$

$$\begin{aligned} &M_{sn} = \mathrm{S}'\cdot(1+b_p\cdot P_w)\cdot(T-T_{cf}) && if \quad T > T_{cf} \\ &M_{sn} = \mathrm{S}'\cdot \mathrm{CFR}\cdot(T-T_{cf}) && if \quad T \le T_{cf} \end{aligned} \tag{B.5}$$

$$\begin{aligned} &dH_{snow}/dt = P_{sn} - M_{sn} \\ &M_{sn} \le P_{sn} + H_{snow}/dt \\ &M_{sn} \ge -W_{snow}/dt \end{aligned} \tag{B.6}$$

with S' (S Series in RS MINERVE): time-varying degree-day snowmelt coefficient [L/T/°C]; *S:* reference degree-day snowmelt coefficient [L/T/°C]; S_{Int}: degree-day snowmelt coefficient interval [L/T/°C]; n: day of the year [T]; S_{Min}: Minimal degree-day snowmelt coefficient [L/T/°C]; S_{Ph} Phase shift of the sinusoidal function [T]; M_{sn}: snowmelt or freezing [L/T]; b_p: melt coefficient due to liquid precipitation [T/L]; T_{cf}: critical snowmelt temperature [°C]; CFR: refreezing coefficient [-]; H_{Snow}: water content of the solid fraction of snow [L]; dt: time step [T]; W_{Snow}: water content of the liquid fraction of snow [L].

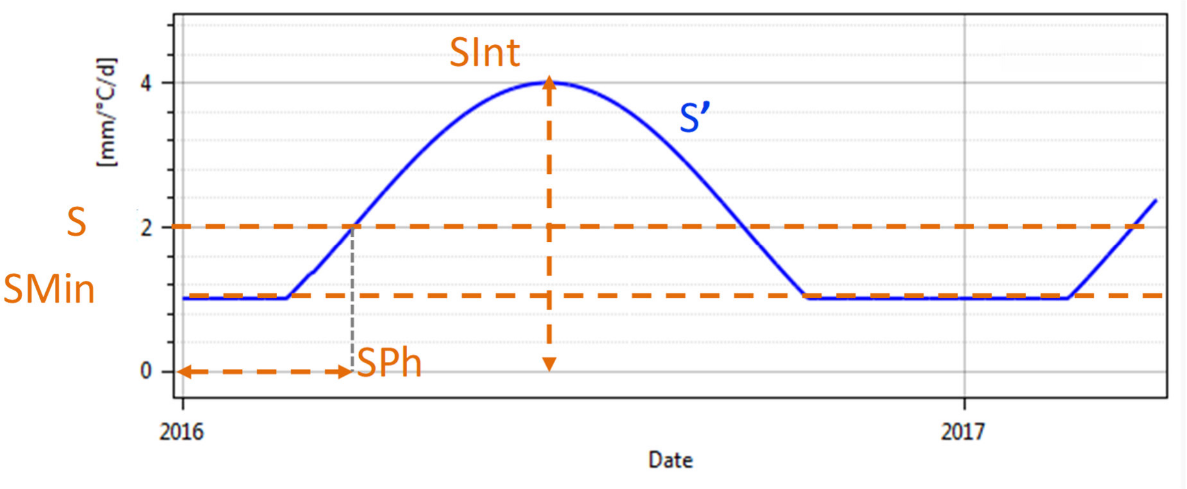


Figure 3 Time-varying degree-day snowmelt coefficient

S_{Min} allows to fix a lower limit to the S' value. The S_{Ph} parameter defines the (horizontal) phase shift of the sinusoidal curve with respect to the first day of the year.

The equivalent precipitation (P_{eq}) depends on the water content of the snow (*equations* B.*7* to B.*9*):

$$\theta = W_{snow}/H_{snow} \tag{B.7}$$

$$\begin{aligned} P_{eq} &= P_w + W_{snow}/dt && if \quad H_{snow} = 0 \\ P_{eq} &= 0 && if \quad H_{snow} > 0 \quad et \quad \theta \leq \theta_{cr} \\ P_{eq} &= (\,\theta - \,\theta_{cr}) \cdot H_{snow}/dt && if \quad H_{snow} > 0 \quad et \quad \theta > \theta_{cr} \end{aligned} \tag{B.8}$$

$$dW_{snow}/dt = P_w + M_{sn} - P_{eq} \tag{B.9}$$

with θ (*Theta* in RS MINERVE): relative water content in the snow pack [-]; θ_{cr} (*ThetaCri* in RS MINERVE): critical relative water content in the snow pack [-]; P_{eq}: equivalent precipitation [L/T].

The snow water equivalent is then the addition of H_{snow} and W_{snow} (*equation* B.*10*):

$$SWE = H_{snow} + W_{snow} \tag{B.10}$$

with SWE: snow water equivalent [L].

The variables for the initial situation associated to this model are θ (*Theta* in RS MINERVE) and SWE. The parameters to adjust are S, S_{Int} and CFR. The parameters S_{Phi}, b_p, θ_{cr}, T_{cp1}, T_{cp2} and T_{cf} can be assumed as constant ($b_p = 0.0125$, $\theta_{cr} = 0.1$, $T_{cp1} = 0$ °C, $T_{cp2} = 4$ °C, $T_{cf} = 0$ °C, $S_{Phi} = 80$ (corresponding to March 21st for the Northern hemisphere; use 264 for Southern hemisphere corresponding to September 21)) but can be also be calibrated for some cases.

The refreezing coefficient CFR is similar to the one found for example in HBV (see Section 2.7) and is available for the Snow-SD model in RS MINERVE program from the version 2.8.0. By default, the CFR is set to 1, recommended values from the literature are around 0.05.

The input variables of the model are precipitation (P) and temperature (T), the output value is the equivalent precipitation (P_{eq}).

2.4. Runoff (SWMM) model description

The SWMM (Storm Water Management Model) model presented hereafter was developed by Metcalf and Eddy (1971).

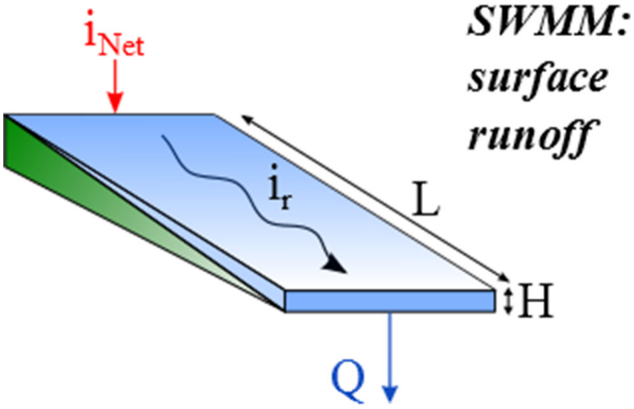


Figure 4 SWMM Runoff model

Table 3 List of parameters and initial conditions for the SWMM model

Object	Name	Units	Description	Regular Range
SWMM	A	m^2	Surface of runoff	>0
	L	m	Length of the plane	>0
	J0	-	Runoff slope	>0
	K	$m^{1/3}/s$	Strickler coefficient	0.1 to 90
	HIni	m	Initial water level downstream of the surface	-

The transfer of the net intensity to an impermeable surface is carried out by the help of a non-linear transfer reservoir Figure 4) depending on the *equations E.1 to E.3*:

$$\begin{aligned} & dH/dt = 2 \cdot (i_{Net} - i_r) \\ & H_r \geq 0 \end{aligned} \tag{E.1}$$

$$i_r = K \cdot \sqrt{J_o} \cdot H^{5/3} \cdot \frac{1}{L} \tag{E.2}$$

$$Q = i_r \cdot A \tag{E.3}$$

with H: runoff water level downstream of the surface [L]; i_{Net}: net intensity [L/T]; i_r: runoff intensity [L/T]; K: Strickler coefficient [$L^{1/3}$/T]; J_o: average slope of the plane [-]; L: length of the plane [L]; A: run-off surface [L^2].

The variable for the initial condition associated to the model is H_r. The parameter to adjust is K. The other parameters (J_o, L, A) are supposed to be constant.

The SWMM model, supplied by a hyetograph of net rainfall (i_{Net}), provides a hydrograph downstream of the surface (Q).

2.5. GSM model description

The GSM model (Figure 5) is composed of 5 sub-models, two corresponding to the Snow-SD model and the other three corresponding to the glacier model. The present model allows an easy construction of this kind of composition.

From the inputs of precipitation (P) and temperature (T), the snow model creates an equivalent precipitation (P_{eq}) which is transferred to the glacier model. The same accounts for the height of the snow (SWE) and the temperature (T).

In the glacier model the equivalent precipitation is transferred to the linear snow reservoir (R_{sn}) and finally to the outlet of the sub-catchment (Q_{snow}). Besides, the sub-model of the glacier melt creates a flow when the height of snow is zero (SWE=0). This glacier flow (P_{eqGL}) is transferred to the linear glacier reservoir (R_{gl}) and the resulting flow ($Q_{glacier}$) to the outlet of the sub-catchment.

The final flow (Q_{tot}) produced by the sub-catchment is the addition of the two flows ($Q_{glacier}$ and Q_{snow}).

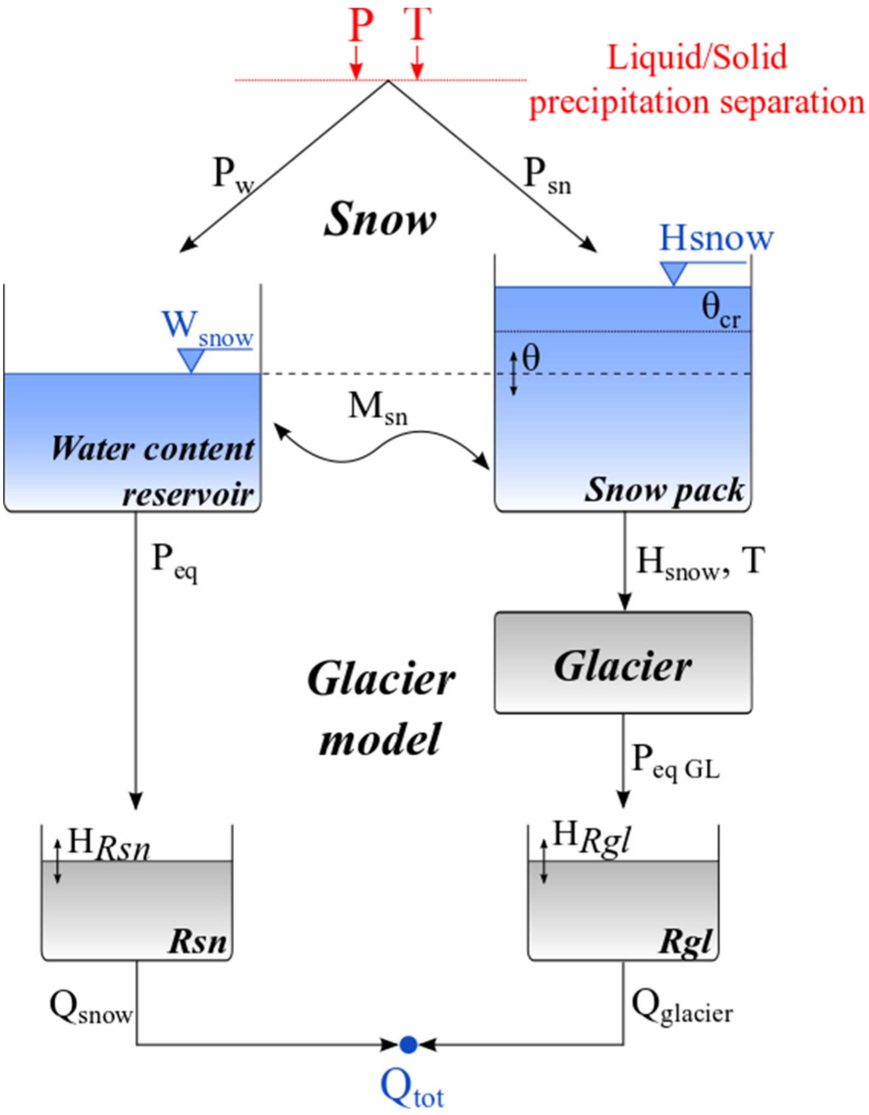


Figure 5 GSM model

Table 4 List of parameters and initial conditions for the GSM model

Object	Name	Units	Description	Regular Range
GSM	A	m^2	Surface of infiltration	>0
	S	mm/°C/d	Reference degree-day snowmelt coefficient	0.5 to 20
	SInt	mm/°C/d	Degree-day snowmelt interval	0 to 4
	SMin	mm/°C/d	Minimal degree-day snowmelt coefficient	≥0
	Sph	d	Phase shift of the sinusoidal function	1 to 365
	ThetaCri	-	Critical relative water content of the snow pack	0.1
	bp	d/mm	Melt coefficient due to liquid precipitation	0.0125
	Tcp1	°C	Minimum critical temperature for liquid precipitation	0
	Tcp2	°C	Maximum critical temperature for solid precipitation	4
	Tcf	°C	Critical snowmelt temperature	0
	G	mm/°C/d	Reference degree-day glacier melt coefficient	0.5 to 20
	GInt	mm/°C/d	Degree-day glacier melt interval	0 to 4
	GMin	mm/°C/d	Minimal degree-day glacier melt coefficient	≥0
	Tcg	°C	Critical glacier melt temperature	0
	Kgl	1/d	Release coefficient of glacier melt reservoir	0.1 to 5
	Ksn	1/d	Release coefficient of snowmelt reservoir	0.1 to 5
	CFR	-	Refreezing coefficient	0 to 1
	SWEIni	m	Initial snow water equivalent height	-
	ThetaIni	-	Initial relative water content in the snow pack	-
	QsnowIni	m^3/s	Initial outflow of linear snow reservoir	-
	QglacierIni	m^3/s	Initial outflow of linear glacier reservoir	-

In a first step, the precipitation is divided into a solid precipitation (P_{sn}) and into a liquid precipitation (P_w) as a function of the temperature (*equations F.1* to F.*3*):

$$P_w = \alpha \cdot P \tag{F.1}$$

$$P_{sn} = (1-\alpha) \cdot P \tag{F.2}$$

$$\begin{array}{ll} \alpha = 0 & if \;\; T \leq T_{cp1} \\ \alpha = (T - T_{cp1})/(T_{cp2} - T_{cp1}) & if \;\; T_{cp1} < T < T_{cp2} \\ \alpha = 1 & if \;\; T \geq T_{cp2} \end{array} \tag{F.3}$$

with P_w: liquid precipitation [L/T]; α: separation factor; P: precipitation [L/T]; P_{sn}: solid precipitation [L/T]; T*: te*mperature [°C]; T_{cp1}: minimum critical temperature for liquid precipitation [°C]; T_{cp2}: maximum critical temperature for solid precipitation [°C].

If the observed temperature is lower than T_{cp1} only solid precipitation is produced. If the temperature is higher than T_{cp2} only liquid precipitation (P_w) is produced. If the temperature observed is found between these two critical values liquid and solid precipitation are produced. The solid precipitation (P_{sn}) is used as input for the snow pack, varying its content as a function of melt or freezing. The snowmelt calculation is performed as follows, using a time-varying degree-day snowmelt coefficient (Figure 3) (Magnusson et al. 2014, Slater and Clark 2005)

$$S' = \max\left(S_{Min} \; ; \; \mathrm{S} + \frac{S_{Int}}{2}\sin\left(2\pi\frac{\mathrm{n} - S_{Ph}}{365}\right)\right) \tag{F.4}$$

$$M_{sn} = \mathrm{S}' \cdot (1 + b_p \cdot P_w) \cdot (T - T_{cf}) \qquad if \quad T > T_{cf}$$
$$M_{sn} = \mathrm{S}' \cdot \mathrm{CFR} \cdot (T - T_{cf}) \qquad if \quad T \leq T_{cf} \tag{F.5}$$

$$dH_{snow}/dt = P_{sn} - M_{sn}$$
$$M_{sn} \leq P_{sn} + H_{snow}/dt \tag{F.6}$$
$$M_{sn} \geq -W_{snow}/dt$$

with S' (SSeries in RS MINERVE): time-varying degree-day snowmelt coefficient [L/T/°C]; *S:* reference degree-day snowmelt coefficient [L/T/°C]; S_{Int}: degree-day snowmelt coefficient interval [L/T/°C]; n: day of the year [T]; S_{Min}: Minimal degree-day snowmelt coefficient [L/T/°C]; S_{Ph} Phase shift of the sinusoidal function [T]; M_{sn}: snowmelt or freezing [L/T]; b_p: melt coefficient due to liquid precipitation [T/L]; T_{cf}: critical snowmelt temperature [°C]; CFR: refreezing coefficient [-]; H_{snow}: water content of the solid fraction of snow [L]; dt: time step [T]; W_{Snow}: water content of the liquid fraction of snow [L].

The S_{ph} parameter defines the phase shift of the sinusoidal curve with respect to the first day of the year.

The equivalent precipitation (P_{eq}) is produced by the water content of the snow (*equations F.7* to *F.9*):

$$\theta = W_{snow}/H_{snow} \tag{F.7}$$

$$P_{eq} = P_w + W_{snow}/dt \qquad if \quad H_{snow} = 0$$
$$P_{eq} = 0 \qquad if \quad H_{snow} > 0 \quad et \quad \theta \leq \theta_{cr} \tag{F.8}$$
$$P_{eq} = (\,\theta - \,\theta_{cr}) \cdot H_{snow}/dt \qquad if \quad H_{snow} > 0 \quad et \quad \theta > \theta_{cr}$$

$$dW_{snow}/dt = P_w + M_{sn} - P_{eq} \tag{F.9}$$

with θ (*Theta* in RS MINERVE): relative water content in the snow pack [-]; θ_{cr} (*ThetaCri* in RS MINERVE): critical relative water content in the snow pack [-]; P_{eq}: equivalent precipitation [L/T].

The equivalent precipitation P_{eq} is then transferred to the linear snow reservoir (R_{sn}), generating an outflow according to *equations F.10 and F.11.*

$$dH_{Rsn}/dt = P_{eq} - K_{sn} \cdot H_{Rsn} \tag{F.10}$$

$$Q_{snow} = K_{sn} \cdot H_{Rsn} \cdot A \tag{F.11}$$

with P_{eq}: equivalent precipitation [L/T]; H_{Rsn}: level in linear snow reservoir [L]; K_{sn}: release coefficient of linear snow reservoir [1/T]; Q_{snow}: outflow of linear snow reservoir [L^3/T]; A: glacier surface [L^2].

The snow water equivalent is then the addition of H_{snow} and W_{snow} (*equation F.12*):

$$SWE = H_{snow} + W_{snow} \tag{F.12}$$

with SWE: snow water equivalent [L].

The glacier melt sub-model only provides a discharge when the snow level is zero (H_{snow}=0). The water intensity produced by the glacier melt (P_{eqGL}) is transferred to the linear glacier reservoir (R_{gl}) and the resulting discharge ($Q_{glacier}$) at the outlet of the sub-catchment.

The glacier melt $Q_{glacier}$ is defined according to *equations F.13 to F.16:*

$$G' = \max\left(G_{Min}\,;\, G + \frac{G_{Int}}{2}\sin\left(2\pi\frac{\mathrm{n} - S_{Ph}}{365}\right)\right) \tag{F.13}$$

$$P_{eqGL} = 0 \qquad if \quad T \leq T_{cg} \quad or \ H_{snow} > 0$$
$$P_{eqGL} = \mathrm{G} \cdot (T - T_{cr}) \qquad if \quad T > T_{cg} \quad and \ H_{snow} = 0 \tag{F.14}$$

$$dH_{Rgl}/dt = P_{eqGL} - K_{gl} \cdot H_{Rgl} \tag{F.15}$$

$$Q_{glacier} = K_{gl} \cdot H_{Rgl} \cdot A \tag{F.16}$$

with P_{eqGL}: glacier melt [L/T];]; T_{cg}: critical glacier melt temperature [°C]; G: degree-day glacier melt coefficient [L/T/ºC]; G_{Min}: Minimal degree-day glacier melt coefficient [L/T/°C]; H_{Rgl}: level of glacier melt reservoir [L]; K_{gl}: coefficient of linear glacier reservoir [1/T]; $Q_{glacier}$: outflow of linear glacier reservoir [L^3/T].

And the total discharge is the addition of $Q_{glacier}$ and Q_{snow}:

$$Q = Q_{snow} + Q_{glacier} \tag{F.17}$$

with Q: total outflow [L^3/T];

The variables for the initial situation associated to this model are θ (*Theta* in RS MINERVE), SWE, H_{Rsn}, and H_{Rgl}.

The parameters to adjust are S, S_{Int}, CFR, $θ_{cr}$, G, G_{Int} , K_{gl} and K_{sn}. The parameters S_{ph}, b_p, $θ_{cr}$, T_{cp1}, T_{cp2},T_{cf} and T_{cg} can be assumed as constant ($b_p = 0.0125$, $θ_{cr} = 0.1$, T_{cp1} = 0 °C, T_{cp2} = 4 °C, T_{cf} = 0 °C, T_{cg} = 0 °C, $S_{Ph} = 80$ (corresponding to March 21st for the Northern hemisphere; use 264 for Southern hemisphere corresponding to September 21)). The parameter A is supposed to be constant.

The refreezing coefficient CFR is similar to the one found for example in HBV (see Section 2.7) and is available for the GSM model in the RS MINERVE program from its version 2.8.0. By default, the CFR is set to 1, recommended values from the literature are around 0.05.

The input variables of the model are the precipitation (P) and the temperature (T). The output is the total discharge (Q_{tot}) at the model outlet.

2.6. SOCONT model description

In the SOCONT model (Figure 6), the Snow-SD model simulates the transient evolution of the snow pack (melt and accumulation) as a function of the temperature (T) and the precipitation (P), thus providing an equivalent precipitation (P_{eq}) that is used as input by the GR3 model. The GR3 model also takes into account the potential evapotranspiration (ETP) and provides the net intensity to the SWMM model.

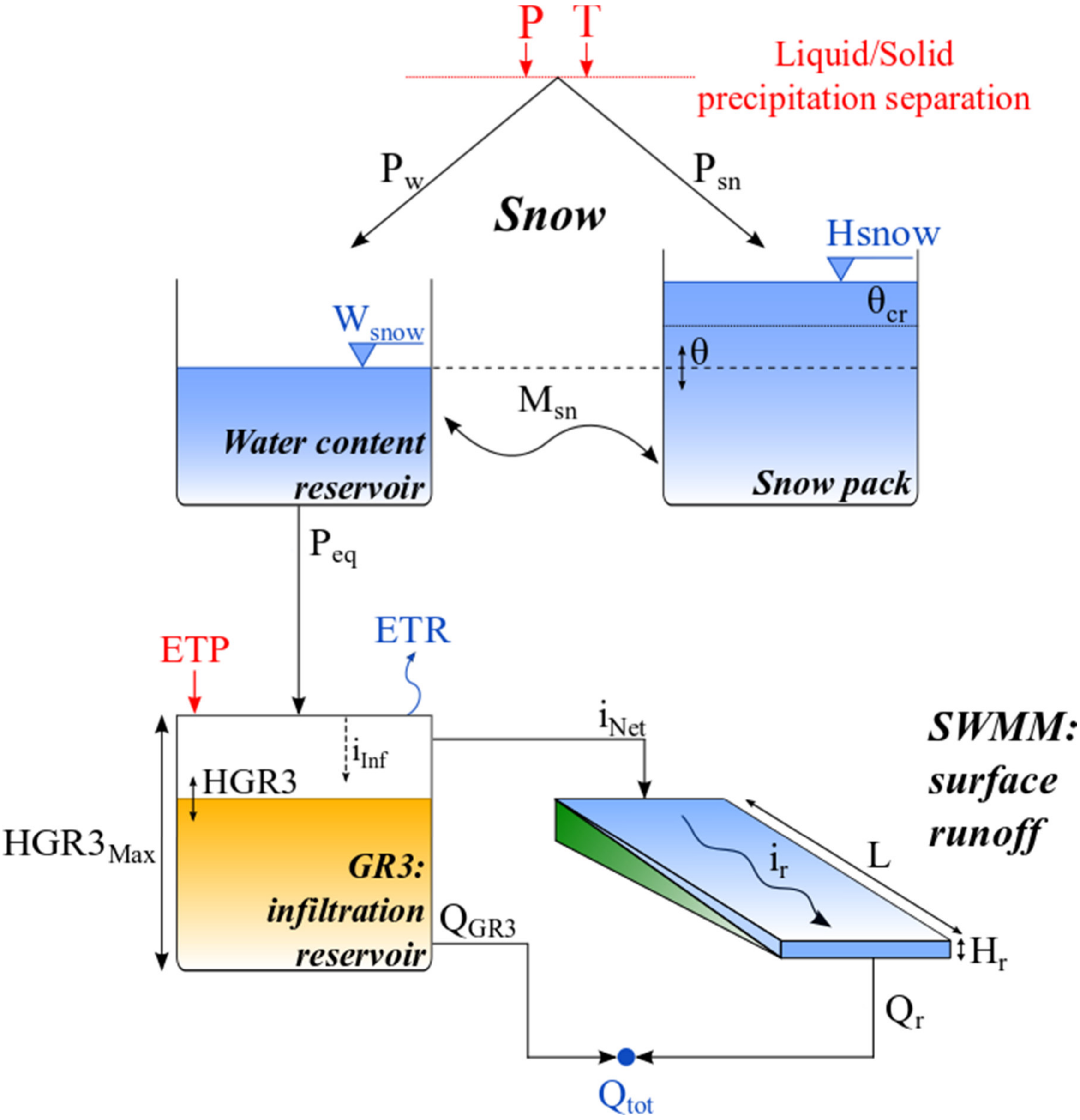


Figure 6 SOCONT Model

Table 5 List of parameters and initial conditions for the SOCONT model

Object	Name	Units	Description	Regular Range
SOCONT	A	m^2	Surface	>0
	S	mm/°C/d	Reference degree-day snowmelt coefficient	0.5 to 20
	SInt	mm/°C/d	Degree-day snowmelt interval	0 to 4
	SMin	mm/°C/d	Minimal degree-day snowmelt coefficient	≥0
	SPh	d	Phase shift of the sinusoidal function	1 to 365
	ThetaCri	-	Critical relative water content of the snow pack	0.1
	bp	d/mm	Melt coefficient due to liquid precipitation	0.0125
	Tcp1	°C	Minimum critical temperature for liquid precipitation	0
	Tcp2	°C	Maximum critical temperature for solid precipitation	4
	Tcf	°C	Critical snowmelt temperature	0
	HGR3Max	m	Maximum height of infiltration reservoir	0 to 2
	KGR3	1/s	Release coefficient of infiltration reservoir	0.00025 to 0.1
	L	m	Length of the plane	>0
	J0	-	Runoff slope	>0
	Kr	$m^{1/3}/s$	Strickler coefficient	0.1 to 90
	CFR	-	Refreezing coefficient	0 to 1
	SWEIni	m	Initial snow water equivalent height	-
	HGR3Ini	m	Initial level in infiltration reservoir	-
	HrIni	m	Initial runoff water level downstream of the surface	-
	ThetaIni	-	Initial relative water content in the snow pack	-

In a first step, the precipitation is divided into a solid precipitation (P_{sn}) and into a liquid precipitation (P_w) as a function of the temperature (*equations G.1* to G.*3*):

$$P_w = \alpha \cdot P \tag{G.1}$$

$$P_{sn} = (1 - \alpha) \cdot P \tag{G.2}$$

$$\begin{array}{ll} \alpha = 0 & if \;\; T \leq T_{cp1} \\ \alpha = (T - T_{cp1})/(T_{cp2} - T_{cp1}) & if \;\; T_{cp1} < T < T_{cp2} \\ \alpha = 1 & if \;\; T \geq T_{cp2} \end{array} \tag{G.3}$$

with P_w*: liquid* precipitation [L/T]; α: separation factor; P: precipitation [L/T]; P_{sn}: solid precipitation [L/T]; T*: te*mperature [°C]; T_{cp1}: minimum critical temperature for liquid precipitation [°C]; T_{cp2}: maximum critical temperature for solid precipitation [°C].

If the observed temperature is lower than T_{cp1} only solid precipitation is produced. If the temperature is higher than T_{cp2} only liquid precipitation (P_w) is produced. If the temperature observed is found between these two critical values liquid and solid precipitation are produced. The solid precipitation (P_{sn}) is used as input for the snow pack, varying its content as a function of melt or freezing. The snowmelt calculation is performed as follows, using a time-varying degree-day snowmelt coefficient (Figure 3) (Magnusson et al. 2014, Slater and Clark 2005):

$$S' = \max\left(S_{Min} \,;\, S + \frac{S_{Int}}{2}\sin\left(2\pi\frac{n - S_{Ph}}{365}\right)\right) \tag{G.4}$$

$$\begin{aligned} M_{sn} &= \mathrm{S}' \cdot (1 + b_p \cdot P_w) \cdot (T - T_{cf}) && if \;\; T > T_{cf} \\ M_{sn} &= \mathrm{S}' \cdot \mathrm{CFR} \cdot (T - T_{cf}) && if \;\; T \leq T_{cf} \end{aligned} \tag{G.5}$$

$$\begin{aligned} dH_{snow}/dt &= P_{sn} - M_{sn} \\ M_{sn} &\leq P_{sn} + H_{snow}/dt \\ M_{sn} &\geq -W_{snow}/dt \end{aligned} \tag{G.6}$$

with S' (SSeries in RS MINERVE): time-varying degree-day snowmelt coefficient [L/T/°C]; *S:* reference degree-day snowmelt coefficient [L/T/°C]; $\mathrm{S_{Int}}$: degree-day snowmelt coefficient interval [L/T/°C]; n: day of the year [T]; $\mathrm{S_{Min}}$: Minimal degree-day snowmelt coefficient [L/T/°C]; $\mathrm{S_{Ph}}$ Phase shift of the sinusoidal function [T]; $\mathrm{M_{sn}}$: snowmelt or freezing [L/T]; b_p: melt coefficient due to liquid precipitation [T/L]; $\mathrm{T_{cf}}$: critical snowmelt temperature [°C]; CFR: refreezing coefficient [-]; $\mathrm{H_{snow}}$: water content of the solid fraction of snow [L]; dt: time step [T]; $\mathrm{W_{Snow}}$: water content of the liquid fraction of snow [L].

The $\mathrm{S_{Ph}}$ parameter defines the phase shift of the sinusoidal curve with respect to the first day of the year.

The equivalent precipitation ($\mathrm{P_{eq}}$) is produced by the water content of the snow (*equations G*.7 *to G*.9):

$$\theta = W_{snow}/H_{snow} \tag{G.7}$$

$$\begin{aligned} P_{eq} &= P_w + W_{snow}/dt && if \;\; H_{snow} = 0 \\ P_{eq} &= 0 && if \;\; H_{snow} > 0 \;\; et \;\; \theta \leq \theta_{cr} \\ P_{eq} &= (\,\theta - \,\theta_{cr}) \cdot H_{snow}/dt && if \;\; H_{snow} > 0 \;\; et \;\; \theta > \theta_{cr} \end{aligned} \tag{G.8}$$

$$dW_{snow}/dt = P_w + M_{sn} - P_{eq} \tag{G.9}$$

with θ (*Theta* in RS MINERVE): relative water content in the snow pack [-]; θ_{cr} (*ThetaCri* in RS MINERVE): critical relative water content in the snow pack [-]; $\mathrm{P_{eq}}$: equivalent precipitation [L/T].

The snow water equivalent is then the addition of $\mathrm{H_{snow}}$ and $\mathrm{W_{snow}}$ (*equation G*.10):

$$SWE = H_{snow} + W_{snow} \tag{G.10}$$

with SWE: snow water equivalent [L].

Next, the infiltration reservoir is computed as follows:

$$\begin{aligned} i_{Inf} &= P_{eq} \cdot (1 - (H_{GR3}/H_{GR3Max})^2) && if \;\; H_{GR3} \leq H_{GR3Max} \\ i_{Inf} &= 0 && if \;\; H_{GR3} > H_{GR3Max} \end{aligned} \tag{G.11}$$

$$\begin{aligned} ETR &= ETP \cdot \sqrt{H_{GR3}/H_{GR3Max}} && if \;\; H_{GR3} \leq H_{GR3Max} \\ ETR &= ETP && if \;\; H_{GR3} > H_{GR3Max} \end{aligned} \tag{G.12}$$

$$i_{Net} = P_{eq} - i_{Inf} \tag{G.13}$$

$$\begin{aligned} Q &= K_{GR3} \cdot H_{GR3} \cdot A && if \;\; H_{GR3} \leq H_{GR3Max} \\ Q &= K_{GR3} \cdot H_{GR3Max} \cdot A && if \;\; H_{GR3} > H_{GR3Max} \end{aligned} \tag{G.14}$$

$$dH_{GR3}/dt = i_{Inf} - ETR - Q/A \tag{G.15}$$

with ETP: potential evapotranspiration [L/T]; $\mathrm{i_{Inf}}$: infiltration intensity [L/T]; $\mathrm{P_{eq}}$: equivalent precipitation [L/T]; H: level in infiltration reservoir [L]; $\mathrm{H_{GR3Max}}$: capacity of infiltration reservoir [L]; ETR: real evapotranspiration [L/T]; Q: base discharge [$\mathrm{L^3}$/T]; $\mathrm{K_{GR3}}$: release coefficient of infiltration reservoir [1/T]; A: surface [$\mathrm{L^2}$]; $\mathrm{i_{Net}}$: net intensity [L/T].

The transfer of the net intensity to an impermeable surface is carried out by the help of a non-linear transfer reservoir depending on the *equations G.16 to G.18*:

$$dH_r/dt = 2 \cdot (i_{Net} - i_r) \qquad \text{(G.16)}$$
$$H_r \geq 0$$

$$i_r = K_r \cdot \sqrt{J_o} \cdot H_r^{\frac{5}{3}} \cdot \frac{1}{L} \qquad \text{(G.17)}$$

$$Q = i_r \cdot A \qquad \text{(G.18)}$$

with H_r: runoff water level downstream of the surface [L]; i_{Net}: net intensity [L/T]; i_r: runoff intensity [L/T]; K_r: Strickler coefficient [$L^{1/3}$/T]; J_o: average slope of the plane [-]; L: length of the plane [L].

The variables for the initial situation associated to this model are θ (*Theta* in RS MINERVE), SWE, H_{GR3} and H_r. The parameters to adjust are S, S_{Int}, b_p, CFR, θ_{cr}, K_{GR3}, H_{GR3Max} and K_r. The parameters S_{Phi}, b_p, θ_{cr}, T_{cp1}, T_{cp2} and T_{cf} can be assumed as constant ($b_p = 0.0125$, $\theta_{cr} = 0.1$, $T_{cp1} = 0$ C, T_{cp2} = 4 °C, T_{cf} = 0 °C, $S_{Phi} = 80$ (corresponding to March 21st for the Northern hemisphere; use 264 for Southern hemisphere corresponding to September 21)). The other parameters (J_o, L, A) are supposed to be constant.

The refreezing coefficient CFR is similar to the one found for example in HBV (see Section 2.7) and is available for the SOCONT model in the RS MINERVE program from its version 2.8.0. By default, the CFR is set to 1; recommended values from the literature are around 0.05.

The input variables of the model are the precipitation (P), the temperature (T) and the potential evapotranspiration (ETP). The output value is the equivalent precipitation (P_{eq}).

The outflow discharge Q_{tot}, composed by the base discharge (Q_{GR3}) and the run-off discharge (Q_r), is finally transferred to the outlet of the sub-catchment.

2.7. HBV model description

The integrated rainfall-runoff model HBV (Bergström, 1976, 1992) is composed of a snow function, a humidity reservoir and two (upper and lower) soil storage reservoirs. The structure of the implemented model is presented in the Figure 7.

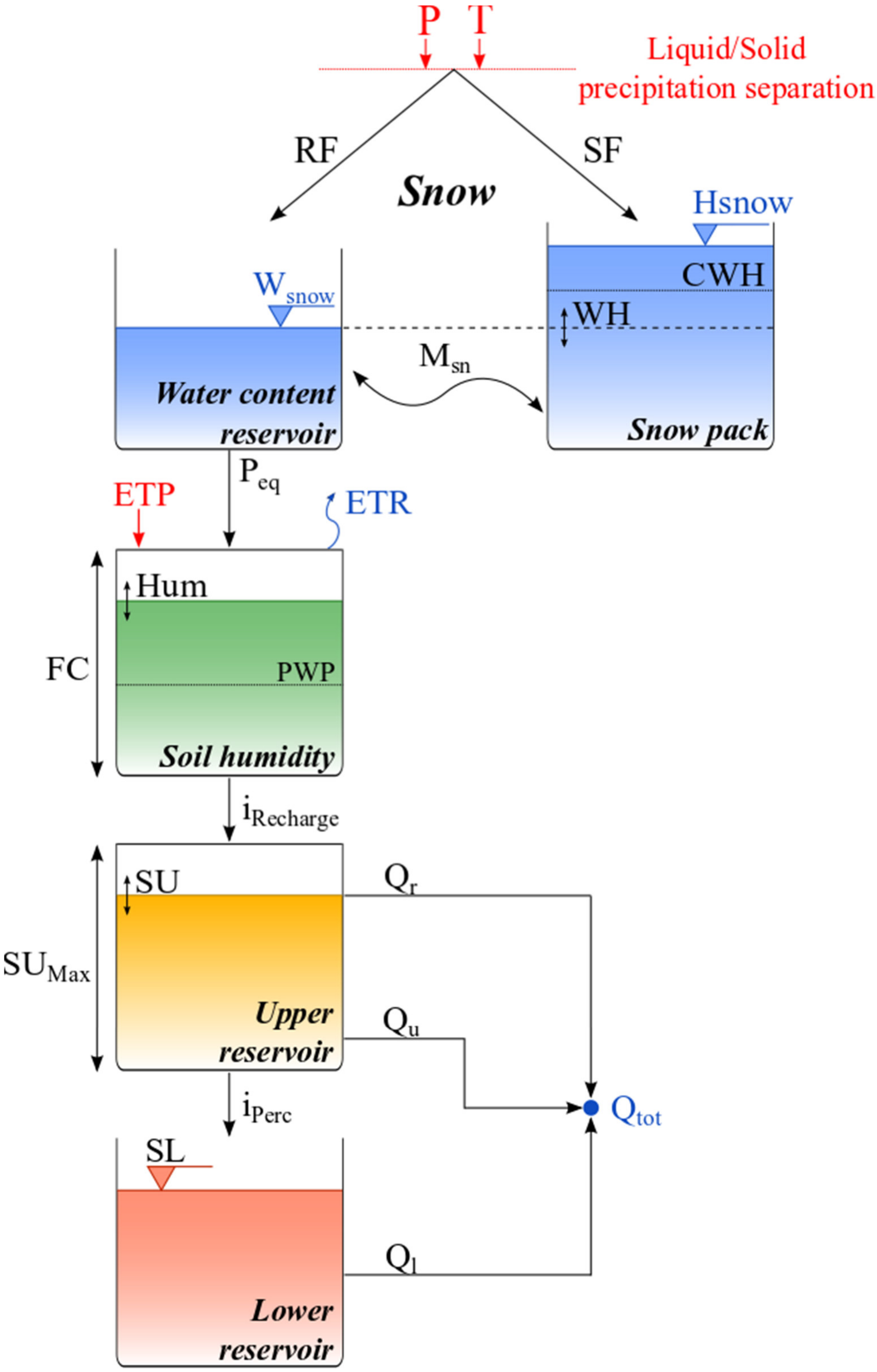


Figure 7 HBV model

Table 6 List of parameters and initial conditions for the HBV model

Object	Name	Units	Description	Regular Range
HBV	A	m^2	Surface of the basin	>0
	CFMax	mm/°C/d	Melting factor	0.5 to 20
	CFR	-	Refreezing factor	0.05
	CWH	-	Critical relative water content of the snow pack	0.1
	TT	°C	Threshold temperature of rain/snow	0 to 3
	TTInt	°C	Temperature interval for rain/snow mixing	0 to 3
	TTSM	°C	Threshold temperature for snow melt	0
	Beta	-	Model parameter (shape coefficient)	1 to 5
	FC	m	Maximum soil storage capacity	0.050 to 0.65
	PWP	-	Soil permanent wilting point threshold	0.030 to 1
	SUMax	m	Upper reservoir water level threshold	0 to 0.10
	Kr	1/d	Near surface flow storage coefficient	0.05 to 0.5
	Ku	1/d	Interflow storage coefficient	0.01 to 0.4
	Kl	1/d	Baseflow storage coefficient	0 to 0.15
	Kperc	1/d	Percolation storage coefficient	0 to 0.8
	SWEIni	m	Initial snow water equivalent height	-
	WHIni	-	Initial relative water content in the snow pack	-
	HumIni	m	Initial humidity	-
	SUIni	m	Initial upper reservoir water level	-
	SLIni	m	Initial lower reservoir water level	-

The precipitation is first divided into snowfall (SF) and rainfall (RF) as a function of the temperature (*equations* H.*1* to H.*3*). If the observed temperature is lower than *TT*-*TTInt*/2, only solid snowfall is produced. If the temperature is higher than *TT*+*TTInt*/2, only rainfall is produced. If the observed temperature is found between these values, both rainfall and snowfall are produced.

$$RF = \alpha \cdot P \qquad \text{(H.1)}$$

$$SF = (1 - \alpha) \cdot P \qquad \text{(H.2)}$$

$$\begin{aligned} &\alpha = 0 && if \quad T < TT - TTInt/2 \\ &\alpha = \frac{T - (TT - TTInt/2)}{TTInt} && if \quad TT - TTInt/2 < T < TT + TTInt/2 \\ &\alpha = 1 && if \quad T > TT + TTInt/2 \end{aligned} \qquad \text{(H.3)}$$

with RF: rainfall [L/T]; *α*: separation factor; *P*: precipitation [L/T]; *SF*: snowfall [L/T]; *T*: temperature [°C]; *TT*: threshold temperature for rain/snow [°C]; *TTInt*: temperature interval for rain/snow mixing [°C].

The snowfall (*SF*) is used as input for the snow pack, varying its content as a function of melt or freezing. The snowmelt calculation is performed as follows:

$$\begin{aligned} &M_{sn} = CFMax \cdot (T - TTSM) && if \quad T > TTSM \\ &M_{sn} = CFR \cdot CFMax \cdot (T - TTSM) && if \quad T \leq TTSM \end{aligned} \qquad \text{(H.4)}$$

$$\begin{aligned} &dH_{snow}/dt = SF - M_{sn} \\ &M_{sn} \leq SF + H_{snow}/dt \\ &M_{sn} \geq -W_{snow}/dt \end{aligned} \qquad \text{(H.5)}$$

with M_{sn}: snowmelt or freezing [L/T]; *CFMax:* degree-day melting factor [L/T/°C]; *CFR*: refreezing factor [-]; *TTSM*: critical snowmelt temperature [°C]; *H_{snow}*: snow height [L]; *W_{snow}*: water content [L]; *dt*: time step [T].

The equivalent precipitation (*P_{eq}*) is produced by the water content of the snow (*equations* H.6 to H.8):

$$WH = W_{snow}/H_{snow} \tag{H.6}$$

$$\begin{aligned} P_{eq} &= RF + W_{snow}/dt && if \;\; H_{snow} = 0 \\ P_{eq} &= 0 && if \;\; H_{snow} > 0 \;\; et \;\; WH \leq CWH \\ P_{eq} &= (\,WH - \,CWH) \cdot H_{snow}/dt && if \;\; H_{snow} > 0 \;\; et \;\; WH > CWH \end{aligned} \tag{H.7}$$

$$dW_{snow}/dt = RF + M_{sn} - P_{eq} \tag{H.8}$$

with WH: relative water content in the snow pack [-]; *CWH*: critical relative water content in the snow pack [-]; *P_{eq}*: equivalent precipitation [L/T].

The snow water equivalent is then the addition of $\mathrm{H_{snow}}$ and $\mathrm{W_{snow}}$ (*equation H.9*):

$$SWE = H_{snow} + W_{snow} \tag{H.9}$$

with SWE: snow water equivalent [L].

The calculation of the recharge is carried out depending on a model parameter *Beta*, as presented in equation H.10. *ETR* is calculated as shown in equation H.11. Finally, the humidity of the soil (*Hum*) is performed taking into account the input (Equivalent precipitation, $\mathrm{P_{eq}}$) and outputs (Recharge intensity and *ETR*) as presented in equation H.12. Additionally, and based on Seibert (1997), parameter PWP is a rate related to parameter FC. Thus, the height of the soil permanent wilting point threshold is calculated multiplying PWP by FC.

$$\mathrm{i}_{\mathrm{Recharge}} = \left(\frac{Hum}{FC}\right)^{Beta} \cdot P_{eq} \tag{H.10}$$

$$\begin{aligned} ETR &= ETP \cdot \frac{Hum}{(PWP \cdot \mathrm{FC})} && if\; Hum < (PWP \cdot \mathrm{FC}) \\ ETR &= ETP && if\; Hum \geq (PWP \cdot \mathrm{FC}) \end{aligned} \tag{H.11}$$

$$\begin{aligned} &dHum/dt = (P_{eq} - i_{\mathrm{Recharge}}) - ETR \\ &Hum \geq 0 \end{aligned} \tag{H.12}$$

with $i_{Recharge}$: Reservoir recharge intensity [L/T] ; *Hum*: Humidity [L]; *FC*: Maximum soil storage capacity [L] ; *Beta*: Model parameter (shape coefficient) [-]; *P_{eq}*: Equivalent precipitation [L/T]; *ETR*: Evapotranspiration [L/T] ; *ETP*: Potential evapotranspiration [L/T] ; *PWP*: Soil permanent wilting point threshold [-].

Then, near surface (or run-off) flow is calculated depending on the water level in the *Upper reservoir* (*SU*) and its threshold, as well as on a flow storage coefficient *K_r*.

$$\begin{aligned} Q_r &= K_r \cdot (SU - SU_{Max}) \;\cdot A && if\; SU > SU_{Max} \\ Q_r &= 0 && if\; SU \leq SU_{Max} \end{aligned} \tag{H.13}$$

with Q_r: Near surface flow (or run-off flow) [L^3/T]; *K_r*: Near surface flow storage coefficient [1/T]; *SU*: Upper reservoir water level [L]; *SU_{Max}*: Upper reservoir water level threshold [L]; *A*: Basin surface [L^2].

The Upper reservoir (or interflow reservoir), corresponding to the upper soil storage and producing the interflow, is computed as follows:

$$dSU/dt = i_{\text{Recharge}} - \left(K_{perc} + K_u\right) \cdot SU - Q_r/A \\ SU \geq 0 \tag{H.14}$$

$$\text{i}_{\text{Perc}} = K_{perc} \cdot SU \tag{H.15}$$

$$Q_u = K_u \cdot SU \cdot A \tag{H.16}$$

with K_{perc}: Percolation storage coefficient [1/T] ; *K_u*: Interflow storage coefficient [1/T] ; *i_{Perc}*: Percolation intensity [L/T] ; *Q_u*: Interflow [L^3/T].

Afterwards, the lower reservoir (or baseflow reservoir), corresponding to the lower soil storage, is calculated as presented in equations H.17 and H.18

$$dSL/dt = i_{\text{Perc}} - K_l \cdot SL \\ SL \geq 0 \tag{H.17}$$

$$Q_l = K_l \cdot SL \cdot A \tag{H.18}$$

with SL: Lower reservoir water level [L]; *K_l*: Baseflow storage coefficient [1/T]; *Q_l*: Baseflow [L^3/T].

And finally the total outflow is:

$$Q_{tot} = Q_r + Q_u + Q_l \tag{H.19}$$

with Q_{tot}: Total outflow [L^3/T].

The initial conditions associated to this model are SWE_{Ini}, WH_{Ini}, Hum_{Ini}, SU_{Ini} and SL_{Ini}. The parameters to adjust are CFMax, CFR, CWH, TT, TTInt, TTSM, Beta, FC, PWP, SU_{max}, K_r, K_u, Kl and K_{perc}. The parameter A is supposed to be constant.

The model inputs are the precipitation (P), the temperature (T) and the potential evapotranspiration (ETP). The output is the total discharge at the model outlet (Q_{tot}).

2.8. GR4J model description

The GR4J model is a global hydrological model with four parameters developed by Perrin et al. (2003). It is an empirical model (Figure 8), but its structure is similar to the conceptual models. It takes into account the humidity and contains two reservoirs (production and routing). Unit hydrographs are also associated for the hydrological behaviour of the basin.

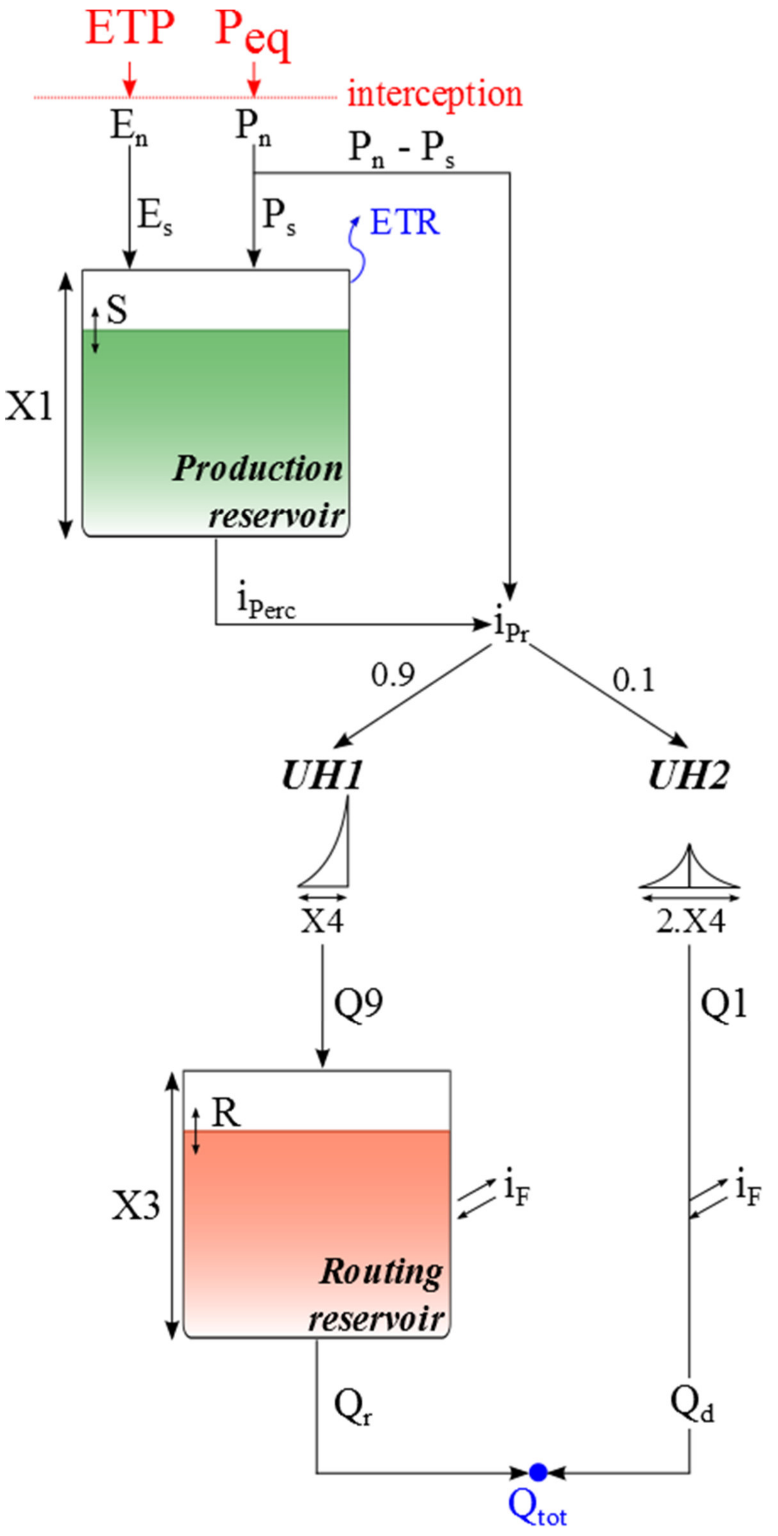


Figure 8 GR4J model

Table 7 List of parameters and initial conditions for the GR4J model

Object	Name	Units	Description	Regular Range
GR4J	A	m^2	Surface of the basin	>0
	X1	m	Capacity of production store	0.01 to 1.2
	X2	m	Water exchange coefficient	-0.005 to 0.003
	X3	m	Capacity of routing store	0.01 to 0.5
	X4	d	UH time base	-0.5 to 1
	SIni	m	Initial water content in the production reservoir	-
	RIni	m	Initial water level in the routing reservoir	-

The first calculation is the neutralization of Peq by ETP for determining a net intensity (Pn) and a net evapotranspiration (En) according to equations I.1.

$$\begin{array}{ll} Pn = Peq - ETP & \\ En = 0 & if \quad P \geq ETP \\ & \\ Pn = 0 & \\ En = ETP - Peq & if \quad P < ETP \end{array} \tag{I.1}$$

with Pn : net rain intensity [L/T] ; Peq : Equivalent precipitation [L/T] ; ETP : potential evapotranspiration [L/T] ; En : net evapotranspiration [L/T].

When Pn is not zero, a part (Ps) of Pn feed the production reservoir (S) as presented in equation I.2. Similarly, when En is not zero, an evapotranspiration (Es) from the production reservoir is calculated depending on the water level in the production reservoir as described in equation I.3.

$$Ps = \frac{d}{dt}\frac{X_1 \cdot \left(1 - \left(\frac{S}{X_1}\right)^2\right) \cdot \tanh\left(\frac{Pn \cdot dt}{X_1}\right)}{1 + \left(\frac{S}{X_1}\right) \cdot \tanh\left(\frac{Pn \cdot dt}{X_1}\right)} \tag{I.2}$$

$$Es = \frac{d}{dt}\frac{S \cdot \left(2 - \left(\frac{S}{X_1}\right)\right) \cdot \tanh\left(\frac{En \cdot dt}{X_1}\right)}{1 + \left(1 - \frac{S}{X_1}\right) \cdot \tanh\left(\frac{En \cdot dt}{X_1}\right)} \tag{I.3}$$

with Ps: rain intensity feeding the production reservoir [L/T]; X1 : maximum capacity of production reservoir S [L] ; S : water content in the production reservoir [L]; Es : evapotranspiration from the production reservoir [L/T].

The percolation (iPerc) flowing from the production reservoir is then calculated as follows:

$$iPerc = \frac{d}{dt}(S + (Ps - Es) \cdot dt) \cdot \left(1 - \left(1 + \left(\frac{4}{9} \cdot \frac{(S + (Ps - Es) \cdot dt)}{X_1}\right)^4\right)^{-\frac{1}{4}}\right) \tag{I.4}$$

with iPerc : percolation intensity [L/T].

Finally, the variation in the production reservoir S is obtained according to equation I.5 :

$$\begin{array}{l} dS/dt = Ps - Es - iPerc \\ S \geq 0 \end{array} \tag{I.5}$$

The quantity of water (iPr) which goes to the routing reservoir is provided by :

$$iPr = iPerc + (Pn - Ps) \tag{I.6}$$

with iPr : intensity flowing to the routing reservoir [L/T].

Afterwards, for the calculation of unit hydrographs, iPr is divided in two flow components: 90% are routed by a unit hydrograph UH1 and a routing reservoir and 10% by a symmetrical unit hydrograph UH2. UH1 and UH2 depend on parameter X4, which is the base time.

The ordinates of the hydrographs are calculated from curves SH1 and SH2, which correspond to the cumulated hydrograph. SH1 and SH2 are defined depending on the time step t as presented in equation I.7 and I.8. The ordinates of UH1 and UH2 are then calculated.

$$\begin{aligned} SH1_t &= 0 \qquad if\ t \leq 0 \\ SH1_t &= \left(\frac{t}{X_4}\right)^{\frac{5}{2}} \quad if 0\ < t < X_4 \\ SH1_t &= 1 \qquad if\ t \geq X_4 \end{aligned} \tag{I.7}$$

$$\begin{aligned} SH2_t &= 0 \qquad if\ t \leq 0 \\ SH2_t &= \frac{1}{2} \cdot \left(\frac{t}{X_4}\right)^{\frac{5}{2}} \qquad if 0\ < t < X_4 \\ SH2_t &= 1 - \frac{1}{2} \cdot \left(2 - \frac{t}{X_4}\right)^{\frac{5}{2}} \quad if\ X_4\ < t < 2 \cdot X_4 \\ SH2_t &= 1 \qquad if\ t \geq 2 \cdot X_4 \end{aligned} \tag{I.8}$$

$$UH1_j = SH1_j - SH1_{j-1} \tag{I.9}$$

$$UH2_j = SH2_j - SH2_{j-1} \tag{I.10}$$

with SH1 : hydrograph 1 from S [-]; SH2 : hydrograph 2 from S [-]; t: time step [T]; X4: base time [T]; UH1 : hydrograph from SH1 [-]; UH2 : hydrograph from SH2 [-].

Then, at each time step k, outflows Q9 and Q1 from the two hydrographs correspond to the convolution of previous intensities according to the repartition provided by the discretized hydrograph and calculated as presented in equations I.11 and I.12.

$$Q9 = 0.9 \cdot \sum_{j=1}^{l} \left(UH1_j \cdot i\mathrm{Pr}_{k-j-l}\right) \tag{I.11}$$

$$Q1 = 0.1 \cdot \sum_{j=1}^{m} \left(UH2_j \cdot i\mathrm{Pr}_{k-j-l}\right) \tag{I.12}$$

with Q9: Unitary flow from the UH1 hydrograph; [L/T]; Q1: Unitary flow from the UH2 hydrograph [L/T]; M j : day (integer) [T]; l = integer value of X4 + 1 [T] ; m = integer value of 2*X4+1 [T].

Then, an exchange function (iF) with the non-atmospheric outside produces an underground water exchange according to I.13 (physical interpretation of this function is not direct). Then, the routing reservoir (R) is emptied by a routing discharge (Qr’) as described in equation I.14.

$$iF = \frac{d}{dt} X_2 \cdot \left(\frac{R}{X_3}\right)^{\frac{7}{2}} \tag{I.13}$$

$$Qr' = \frac{d}{dt}(R + (Q9 + iF) \cdot dt) \cdot \left(1 - \left(1 + \left(\frac{R + (Q9 + iF) \cdot dt}{X_3}\right)^4\right)^{-\frac{1}{4}}\right) \qquad \text{(I.14)}$$

with iF : exchange function [L/T] ; X2 : water exchange coefficient, positive for contributions, negative for losses due to infiltration or zero when no exchange is produced [L]; R : water level in the routing reservoir [L] ; X3 : one day capacity of routing reservoir [L]; Qr': unitary routing reservoir outflow [L/T].

The variation of the water level in the routing reservoir is:

$$\begin{aligned} &dR/dt = Q9 - iF - Qr' \\ &R \geq 0 \end{aligned} \qquad \text{(I.15)}$$

The outflow Q1 from hydrograph UH2 has also the same exchange for providing the complementary flow Qd':

$$Qd' = Q1 + iF \qquad \text{(I.16)}$$

with Qd': unitary complementary outflow [L/T].

Finally, the model outflows are calculated as presented in equations I.17 and I.18 and the total flow at the outlet, Qtot, as presented in equation I.19:

$$Qr = Qr' \ \cdot \ A \qquad \text{(I.17)}$$

$$Qd = Qd' \ \cdot \ A \qquad \text{(I.18)}$$

$$Qtot = Qr + Qd \qquad \text{(I.19)}$$

with Qr: outflow from the routing reservoir [L^3/T]; A: Basin surface [L^2]; Qd: complementary outflow [L^3/T]; Qtot : total outflow [L^3/T].

The variables associated to this model are S_{Ini} and R_{Ini}. The parameters to adjust are X1, X2, X3 and X4. The parameter A is supposed to be constant after its calculation.

The model inputs are the equivalent precipitation (P_{eq}) and the potential evapotranspiration (ETP). The output is the total discharge at the model outlet (Q_{tot}).

2.9. SAC-SMA model description

The SAC-SMA (Figure 9) or SACRAMENTO hydrological model was developed in the 70's (Burnash et al., 1973; Burnash, 1995) to optimize humidity characteristics into the soil, distributed into different level, with rational percolation characteristics, for an efficient simulation of discharges. This model calculates the total discharge from the precipitation (P) and the potential evapotranspiration (ETP) depending on the parameters and initial conditions presented in Table 8.

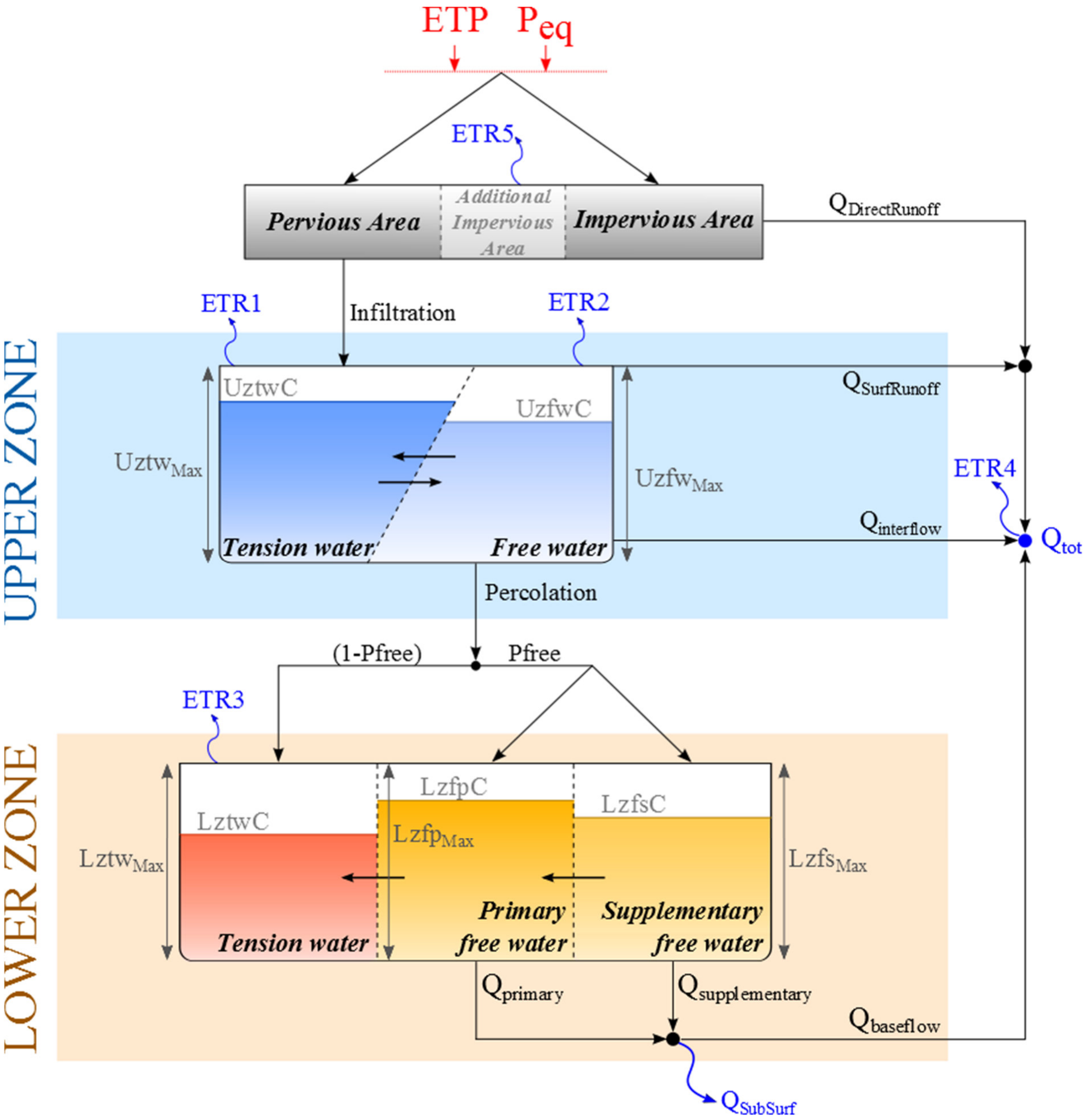


Figure 9 SAC-SMA model

Table 8 List of parameters and initial conditions for the SAC-SMA model

Object	Name	Units	Description	Regular Range
SAC-SMA	A	m^2	Surface of the basin	>0
	Adimp	-	Maximum fraction of an additional impervious area due to saturation	0 to 0.2
	Pctim	-	Permanent impervious area fraction	0 to 0.05
	Riva	-	Riparian vegetarian area fraction	0 to 0.2
	UztwMax	m	Upper Zone Tension Water capacity	0.01 to 0.15
	UzfwMax	m	Upper Zone Free Water capacity	0.005 to 0.10
	Uzk	1/d	Interflow depletion rate from the Upper Zone Free Water storage	0.10 to 0.75
	Zperc	-	Ratio of maximum and minimum percolation rates	10 to 350
	Rexp	-	Shape parameter of the percolation curve	1 to 4
	Pfree	-	Percolation fraction that goes directly to the Lower Zone Free Water storages	0 to0.6
	LztwMax	m	The Lower Zone Tension Water capacity	0.05 to 0.40
	LzfpMax	m	The Lower Zone primary Free Water capacity	0.03 to 0.80
	LzfsMax	m	The Lower Zone supplementary Free Water capacity	0.01 to 0.40
	Rserv	-	Fraction of Lower Zone Free Water not transferable to Lower Zone Tension Water	0 to 1
	Lzpk	1/d	Depletion rate of the Lower Zone primary Free Water storage	0.001 to 0.03
	Lzsk	1/d	Depletion rate of the Lower Zone supplemental Free Water storage	0.02 to 0.3
	Side	-	Ratio of deep percolation from Lower Zone Free Water storages	0 to 0.5
	AdimIni	m	Initial Tension Water content of the Adimp area	-
	UztwIni	m	Initial Upper Zone Tension Water content	-
	UzfwIni	m	Initial Upper Zone Free Water content	-
	LztwIni	m	Initial Lower Zone Tension Water content	-
	LzfpIni	m	Initial Lower Zone Free supplementary content	-
	LzfsIni	m	Initial Lower Zone Free primary content	-

When considering rainfall over a basin, two basic basin areas are taken into account:

- An impervious portion of the soil mantle covered by streams, lake surfaces, marshes and other impervious material directly linked to the streamflow network. This area produces impervious and direct runoff $Q_{DirectRunoff}$ from any rainfall.
- A pervious portion of the soil mantle which distributes rainwater to the next storages. In addition, the pervious area can produce runoff when rainfall rates are sufficiently heavy.

The basin may present permanent or temporarily impervious areas, determined by:

- The permanent impervious area fraction (**Pctim**).

- The maximum fraction of an additional impervious area due to saturation (**Adimp**). It represents the maximum portion of reservoirs and temporary watersheds contained in the basin.

Thus, the permanent pervious area fraction of the basin is calculated following equation J.1

$$Parea = 1 - (Pctim + Adimp) \tag{J.1}$$

with Parea : permanent pervious area fraction of the basin [-] ; Pctim : permanent impervious area fraction [-] ; Adimp : maximum fraction of an additional impervious area due to saturation [-].

It is worth mentioning that the behaviour of the additional impervious area might be purely pervious, purely impervious or mixed, depending on the storage of the permanent pervious and impervious areas. The pervious capacity (or fraction) of this area will decrease as the rainfall grows.

As shown in the Figure 9, the SAC model divides the soil in 2 zones:

- An **Upper Zone**, sub-divided in two areas that receive the infiltrated water from rainfall.
- A **Lower Zone**, sub-divided in 3 reservoirs, where the input is the percolation from the Upper Zone.

The main components of the model and their flow transfers are described below.

Upper Zone Tension Water storage

This storage represents the water bound by adhesion and cohesion in between the soil pores as well as the water intercepted by vegetation. The water can be only consumed by evapotranspiration.

The water that reaches this reservoir comes from the rain felt into the pervious area of the basin, but also from transfers from the Upper Zone Free Water storage.

The following concepts are defined for this reservoir:

- The Upper Zone *Tension Water Maximum* capacity (**UztwMax**) is the maximum Tension Water storage admissible in the Upper Zone.
- The Upper Zone *Tension Water Content* (**UztwC**) is the Tension Water stored in this Upper Zone at any given time.

If the relative content of Tension Water (ratio between its content and its maximum storage) is smaller than the relative content of Free Water, a transfer is produced from the Free Water storage to the Tension Water storage. The expressions of these ratios and the infiltration transfer are shown in equations J.2 to J.5:

$$T = \frac{UztwC}{UztwMax} \tag{J.2}$$

$$F = \frac{UzfwC}{UzfwMax} \tag{J.3}$$

$$Uztr = d\left(\frac{UzfwC \cdot UztwMax - UztwC \cdot UzfwMax}{UzfwMax + UztwMax}\right)/dt \qquad \text{if } T < F \tag{J.4}$$

$$Uztr = 0 \qquad \text{if } T \geq F \tag{J.5}$$

with T: relative content of Upper Zone Tension Water [-]; F: relative content of Upper Zone Free Water [-]; Uztr: infiltration transfer from the Free Water storage to the Tension Water storage [L/T]; UztwC: the Upper Zone Tension Water storage [L]; UztwMax: the Upper Zone Tension Water capacity [L]; UzfwC: the Upper Zone Free Water storage [L]; UzfwMax: the Upper Zone Free Water capacity [L].

Although free water is present during filling of Upper Zone Tension Water (by infiltration), this free water is rapidly transformed into Upper Zone Tension Water until tension water relative content requirements are satisfied.[1]

In addition, when the Upper Zone Tension Water volume has been filled, excess moisture above the Upper Zone Tension Water capacity is accumulated in the Upper Zone Free Water.

Upper Zone Free Water storage

This storage represents the water not attached to the soil particles and free to move under gravitational forces. The water that reaches this reservoir comes from the infiltration of the Upper Zone Tension Water storage

The following concepts are defined:

- The Upper Zone *Free Water Maximum* capacity (**UzfwMax**) is the maximum Free Water storage admissible in the Upper Zone.
- The Upper Zone *Free Water Content* (**UzfwC**) is the Free Water stored in the Upper Zone at any given time.
- The *Interflow depletion rate* (**Uzk**) represents the portion of the Upper Zone Free Water that is transferred outside as interflow.

The water may be depleted by evapotranspiration, percolation to the Lower Zone or horizontal flow (surface runoff and interflow).

On the one hand, the percolation to the Lower Zone is computed prior to the interflow computation as it is its preferred path. It depends on the deficiency of the Lower Zone moisture volumes, on the soil properties, and on the water relative storage in this reservoir. The first 2 factors define the percolation demand from the Lower Zone (DDA). This demand represents the percolation in case of total percolation availability on the Upper Zone Free Water storage.

$$PBase = LzfpMax \cdot Lzpk + LzfsMax \cdot Lzsk \tag{J.6}$$

$$DEFR = \frac{(LztwMax - LztwC) + (LzfpMax - LzfpC) + (LzfsMax - LzfsC)}{LztwMax + LzfpMax + LzfsMax} \tag{J.7}$$

$$DDA = PBase \cdot (1 + Zperc \cdot DEFR^{Rexp}) \tag{J.8}$$

with PBase: the continuing percolation rate under saturated condition (maximal percolation) [L/T]; DDA: Lower Zone maximum percolation demand [L/T]; Zperc: ratio of maximum and minimum

[1] http://www.nws.noaa.gov/oh/hrl/nwsrfs/users_manual/part2/_pdf/23sacsma.pdf

percolation rates [-]; DEFR: relative Lower Zone water deficiency [-]; Rexp: shape parameter of the percolation curve [-].

The effective percolation depends on the percolation demand and the ratio between the Upper Zone Free Water content and the Upper Zone Free Water capacity as shown in equation J.9. In addition, the effective percolation is limited by the humidity content in the Upper Zone and also by the deficiency in the Lower Zone (equations J.10 and J.11 respectively):

$$PERC = DDA \cdot \frac{UzfwC}{UzfwMax} \tag{J.9}$$

$$PERC \leq d(UzfwC)/dt \tag{J.10}$$

$$PERC \leq d[(LztwMax - LztwC) + (LzfpMax - LzfpC) + (LzfsMax - LzfsC)]/dt \tag{J.11}$$

with PERC: real percolation [L/T]; DDA: Lower Zone maximum percolation demand [L/T]; UzfwC: Upper Zone Free Water content [L]; UzfwMax: Upper Zone Free Water capacity [L].

On the other hand, the interflow occurs only when the rate of precipitation exceed the rate at which downward motion can occur from the Upper Zone Free Water. It depends on the water content and on its interflow depletion rate Uzk, as shown in equation J.12:

$$Q_{interflow} = Uzk \cdot UzfwC \cdot Parea \cdot A \tag{J.12}$$

with $Q_{interflow}$: interflow [L^3/T]; Uzk: interflow depletion rate from the Upper Zone Free water storage [T^{-1}]; UzfwC: Upper Zone Free Water content [L]; Parea : pervious area fraction of the basin [-]; A: surface of the basin [L^2].

Moreover, when the Upper Zone Free Water storage is completely full and the precipitation intensity exceeds both the percolation rate and the maximum interflow drainage capacity, the excess precipitation results in surface runoff $Q_{SurfRunoff}$.

Lower Zone Tension Water storage

This storage represents the semi-saturated zone. It characterizes the volume of moisture in the lower soils which will be claimed by dry soil particles when moisture from a wetting front reaches that depth. The water is consumed through evapotranspiration, proportionally to the remaining ETP of the Upper Zone Tension Water storage.

The water reaching this reservoir comes from the Upper Zone percolation. One fraction (Pfree) of this percolation goes directly to the Free Water storages, even if the Tension Water storage is not yet full, and the other fraction (1-Pfree) goes to the Tension Water storage of the Lower Zone [2].The water also reaches this storage from transfers between the other Lower Zone Free Water storages.

The following concepts are defined:

- The Lower Zone *Tension Water Maximum* capacity (**LztwMax**) is the maximum Tension Water storage admissible in the Lower Zone.

[2] If the Tension Water storage of the Lower Zone is completely full, then percolation goes entirely to the 2 Free Water storages.

- The Lower Zone *Tension Water Content* (**LztwC**) is the Tension Water stored in the Lower Zone at any given time.

If the relative content of Tension Water (ratio between its content and its maximum storage) is smaller than the relative content of Free Water, the incoming excess is transferred from Free Water to Tension Water. The expressions of both ratios are shown in equations J.13 to J.15:

$$RFW = Rserv \cdot (LzfpMax + LzfsMax) \tag{J.13}$$

$$T = \frac{LztwC}{LztwMax} \tag{J.14}$$

$$F = \frac{LzfpC + LzfsC - RFW}{LzfpMax + LzfsMax - RFW} \tag{J.15}$$

with T: relative content of Lower Zone Tension Water [-]; F: relative content of Lower Zone Free Water [-]; RFW: Lower Zone Free Water not transferable to Lower Zone Tension Water, susceptible to become baseflow [L]; Rserv: fraction of Lower Zone Free Water not transferable to Lower Zone Tension Water[-].

If the relative content of the Lower Zone Tension Water is smaller than the relative fullness of the global Lower Zone, a water transfer DEL occurs from the Lower Zone Free Supplementary reservoir to the Lower Zone Tension Water reservoir, as described by the following equations:

$$R = \frac{LzfpC + LzfsC + LztwC - RFW}{LzfpMax + LzfsMax + LztwMax - RFW} \tag{J.16}$$

$$DEL = LztwMax \cdot d(R - T)/dt \tag{J.17}$$

with DEL: the Free-Tension Water transfer in the Lower Zone [L/T]; R: ratio between the available water for evapotranspiration and the total water content, in the whole Lower Zone [-]; RFW: Lower Zone Free Water not transferable to Lower Zone Tension Water, susceptible to become baseflow [L].

Lower Zone Primary and Supplementary Free Water storages

This storage represents the saturated zone of the subsoil, i.e. the aquifer. There are two types of Lower Zone Free Water: a primary type with a very slow draining providing baseflow over long periods of time, and a supplementary type which supplements the baseflow after a period of relatively recent rainfall.

The water that reaches these storages comes from the *Pfree* fraction of the percolation of the Upper Zone Free Water. This fraction is distributed into the primary and supplementary storages according to their deficiency of water.

The following concepts are defined:

- The Lower Zone *primary Free Water Maximum* capacity (**LzfpMax**) is the maximum primary water storage admissible in the Lower Zone.
- The Lower Zone *primary Free Water Content* (**LzfpC**) is the primary Free Water stored in the Lower Zone at any given time.
- The *Depletion rate of the Lower Zone primary Free Water storage* (**Lzpk**) is the portion of primary Free Water that drains as baseflow per day.

- The Lower Zone *supplementary Free Water Maximum* capacity (**LzfsMax**) is the maximum supplementary water storage admissible in the Lower Zone.
- The Lower Zone *supplementary Free Water Content* (**LzfsC**) is the supplementary Free Water stored in the Lower Zone at any given time.
- The *Depletion rate of the Lower Zone supplementary Free Water storage* (**Lzsk**) is the portion of supplementary Free Water that drains as baseflow per day.

The drained water leaving these two storages (primary flow $Q_{primary}$ and supplementary flow $Q_{supplementary}$) follows the Darcy's law and forms the baseflow of the Lower Zone $Q_{Lower\ Zone}$ as shown in equations J.18 to J.20:

$$Q_{primary} = Lzpk \cdot LzfpC \cdot Parea \cdot A \tag{J.18}$$

$$Q_{supplementary} = Lzsk \cdot LzfsC \cdot Parea \cdot A \tag{J.19}$$

$$Q_{Lower\ Zone} = Q_{primary} + Q_{supplementary} \tag{J.20}$$

with $Q_{primary}$: primary baseflow [L^3/T]; $Q_{supplementary}$: supplementary baseflow [L^3/T]; $Q_{Lower\ Zone}$: total baseflow produced in the Lower Zone [L^3/T]; Lzpk: depletion rate of the Lower Zone primary Free Water storage [T^{-1}]; LzfpC: Lower Zone primary Free Water Content [L]; Lzsk: depletion rate of the Lower Zone supplementary Free Water storage [T^{-1}]; LzfsC: Lower Zone supplementary Free Water Content [L]; Parea : pervious area fraction of the basin [-]; A: surface of the basin [L^2].

The model allows including baseflow losses, called subsurface flow $Q_{SubSurf}$, due to the existence of geological faults, fractured rocks, etc. that lead to filtrations throughout the aquifer. The *Side* parameter captures these losses as shown in equation J.21:

$$Q_{SubSurf} = Side \cdot Q_{baseflow} \tag{J.21}$$

with Side: ratio of deep percolation from Lower Zone Free Water storages [-], $Q_{baseflow}$: Lower Zone Free Water volume actually integrated to the channel outflow hydrographs [L^3/T]; $Q_{SubSurf}$: discharge lost into the aquifer [L^3/T].

And so this *Side* parameter is used to correct the baseflow as follows:

$$Q_{baseflow} = \frac{Q_{Lower\ Zone}}{1 + Side} \tag{J.22}$$

$$Q_{SubSurf} = Side \cdot \frac{Q_{Lower\ Zone}}{1 + Side} \tag{J.23}$$

Evapotranspiration

The real evapotranspirations are obtained from each Tension Water storage, from the transfers between Free Water and Tension Water storages, from riverside vegetation and from impervious areas. They are described hereafter.

- $ETR1$: evapotranspiration from the Upper Zone Tension Water reservoir. If this reservoir is not full, the evapotranspiration is proportional to its content (equation J.24):

$$ETR1 = ETP \cdot \frac{UztwC}{UztwMax} \leq UztwC \tag{J.24}$$

with ETP: potential evapotranspiration [L/T].

The remaining evapotranspiration demand RED and the updated content of water in the Upper Zone Tension Water reservoir UztwC are then calculated:

$$RED = ETP - ETR1 \tag{J.25}$$

$$d(UztwC)/dt = -ETR1 \tag{J.26}$$

- $ETR2$: evapotranspiration from the Upper Zone Free Water reservoir. This evapotranspiration is equal to the remaining RED left by the Tension Water reservoir, but it is limited by the Upper Zone Free Water content UzfwC:

  $$ETR2 = \min\{RED; d(UzfwC)/dt\} \tag{J.27}$$

  Then, the remaining evapotranspiration REDLz from the Lower Zone is:

  $$REDLz = RED - ETR2 \tag{J.28}$$

  The variation of the water content UzfwC in the Upper Zone Free Water reservoir is defined as follows:

  $$d(UzfwC)/dt = -ETR2 \tag{J.29}$$

- $ETR3$: evapotranspiration from the Lower Zone Tension Water reservoir. The evapotranspiration availability corresponds to the remaining evapotranspiration demand in the Lower Zone. Evapotranspiration in this reservoir is proportional to the remaining evapotranspiration and to the Lower Zone Tension Water content, and inversely proportional to the Tension Water reservoirs' capacity of the Upper Zone and the Lower Zone:

  $$ETR3 = REDLz \cdot \frac{LztwC}{UztwMax + LztwMax} \leq d(LztwC)/dt \tag{J.30}$$

- $ETR4$: evapotranspiration from the basin covered by riverside vegetation. Riverside vegetation absorbs the deficiency of potential evapotranspiration from the Upper Zone and the Lower Zone:

  $$ETR4 = ETP - ETR1 - ETR2 - ETR3 \tag{J.31}$$

- $ETR5$: evapotranspiration from the impervious fraction of the basin.

  $$ETR5 = ETR1 + (RED + ETR2) \cdot \frac{AdimC - UztwC - V_{ETR1}}{UztwMax + LztwMax} \tag{J.32}$$

  $$V_{ETR1} = \int_{t1}^{t2} ETR1.dt \tag{J.33}$$

  with AdimC: sum of the Upper Zone and Lower Zone Tension Water Content in the *Adimp* area [-]; V_{ETR1}: evapotranspirated water from the Upper Zone Tension Water reservoir in a time step [L].

Each evapotranspiration value is weighted based on the portion of the basin area in which it is produced. The total evapotranspiration is finally:

$$ETRtotal = [(ETR1 + ETR2 + ETR3) \cdot Parea + ETR4 \cdot Riva + ETR5 \cdot Adimp] \cdot A \qquad \text{(J.34)}$$

with ETRtotal: total evapotranspiration [L^3/T]; Parea: pervious area fraction of the basin [-]; Riva: riparian vegetarian area fraction [-]; Pctim: permanent impervious area fraction of the basin [-]; Adimp: maximum fraction of an additional impervious area due to saturation [-]; A: surface of the basin [L^2].

A more detailed guide about the SAC-SMA equations can be found on the following National Weather Service link: http://www.nws.noaa.gov/iao/iao_SAC_SMA.php

2.10. Channel routing description

The channel routing can be solved by the St-Venant, Muskingum-Cunge or Kinematic wave. The three possibilities are presented hereafter and their parameters in Table 9.

Table 9 List of parameters for the channel routing

Object	Name	Units	Description	Regular Range
Reach	L	m	Length	>0
	B0	m	Width of the channel base	>0
	m	-	Side bank relation coefficient (1H/mV)	0.1 to 1
	J0	-	Slope	>0
	K	$m^{1/3}/s$	Strickler coefficient	10 to 90
	N	-	Number of sections (not for Lag-Time)	>0
	Lag	min	Lag time (only for Lag-Time)	≥0
	QIni	m^3/s	Initial discharge	-

St. Venant routing

The St. Venant equations solving the 1D unsteady flow are:

$$\frac{\partial A}{\partial t}+\frac{\partial Q}{\partial x}=0 \qquad \text{(K.1)}$$

$$\frac{\partial Q}{\partial t}+\frac{\partial}{\partial x}\left(\frac{Q^2}{A}+g\cdot I_1\right)=g\cdot A\cdot\left(J_0-J_f\right)+g\cdot I_2 \qquad \text{(K.2)}$$

with A: cross sectional flow area [L^2]; Q: discharge [L^3/T]; J_0: bottom slope; J_f: friction slope [-]; I_1: profile coefficient [L^3]; I_2: coefficient for cross sectional variation [L^2].

Equation K.1 expresses the mass conservation while equation K.2 ensures the conservation of momentum. The term I_1 takes into account the shape of the transversal profile and is calculated as follows:

$$I_1=\int_0^h (h-\eta)\cdot b(\eta)\,d\eta \qquad \text{(K.3)}$$

The term b represents the cross sectional variation for the level η and constitutes an integration variable according to Figure 10. St. Venant equations can be solved in RS MINERVE for a trapezoidal profile (Figure 11):

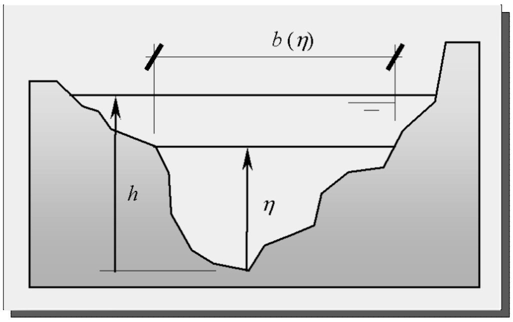


Figure 10 Descriptive sketch for parameters used in the calculation of I_1

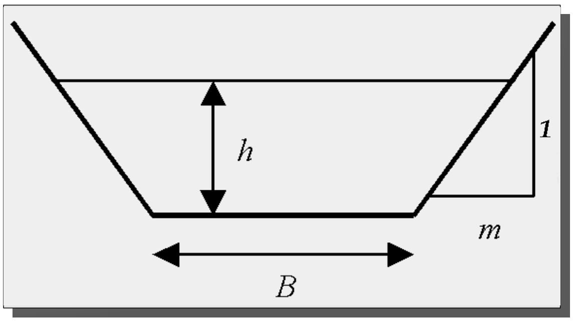


Figure 11 Transversal profile available for the computation of channel routing

For this trapezoidal section, the equation defining I_1 is reduced to:

$$I_1 = \frac{B \cdot h^2}{2} + \frac{m \cdot h^3}{3} \tag{K.4}$$

with B: width of the base of the transversal profile [L]; h: water level [L]; m: side bank relation coefficient (1 vertical / m horizontal) [-].

The friction slope J_f is calculated according to Manning-Strickler:

$$J_f = \frac{Q \cdot |Q|}{A^2 \cdot K^2 \cdot R_h^{4/3}} \tag{K.5}$$

$$R_h = \frac{A}{L_p} \tag{K.6}$$

with J_f: friction slope [-]; K: Strickler coefficient [$L^{1/3}/T$]; R_h : hydraulic radius [L]; A: flow area [L^2]; L_p : wetted perimeter [L].

The term I_2 takes into account the variation of the section along the channel. In the case of a prismatic channel I_2 is equal to zero. In general, I_2 is:

$$I_2 = \int_0^h (h - \eta) \left.\frac{\partial b}{\partial x}\right|_{h=\eta} \tag{K.7}$$

For a prismatic channel, equations K.1 and K.2 are solved by the Euler method (first order) as follows:

$$A_{j+1}{}^{n+1} = A_{j+1}{}^{n} - \frac{\Delta t}{\Delta x}(Q_{j+1}{}^{n} - Q_j{}^{n}) \tag{K.8}$$

$$\begin{aligned} Q_{j+1}{}^{n+1} &= Q_{j+1}{}^{n} - \frac{\Delta t}{\Delta x}\left(\frac{(Q_{j+1}{}^{n})^2}{A_{j+1}{}^{n}} - \frac{(Q_j{}^{n})^2}{A_j{}^{n}} + g \cdot I_{1,j+1}{}^{n} - g \cdot I_{1,j}{}^{n}\right) + \\ &+\Delta t \cdot g \cdot \frac{(A_j{}^{n} + A_{j+1}{}^{n})}{2} \cdot J_0 - \Delta t \cdot g \cdot A_{j+1}{}^{n} \cdot J_{f\,j+1}{}^{n} \end{aligned} \tag{K.9}$$

with index j and j+1 representing the spatial position; exponent n and n+1 representing the time increment; g: gravity constant [L^2/T]; Δx: longitudinal increment [L].

The downstream boundary condition used by RS MINERVE is the normal flow depth.

In practice, according to the physical situation to be simulated, some terms of the complete dynamic equations can be eliminated in order to get simplified expressions without losing precision. Applicable solutions are the diffusive and kinematic waves which are presented in the following.

Muskingum-Cunge routing

Ruling out the first two terms of equation K.2 yields:

$$\frac{\partial I_1}{\partial x} = A \cdot (J_0 - J_f) + I_2 \tag{K.10}$$

This new equation corresponds to the approximation of the diffusive wave. With the supplementary hypothesis of a prismatic channel (Cunge, 1991) it is possible to express equation K.2 as follows:

$$\frac{\partial Q}{\partial t} + \left(\frac{Q}{BD}\frac{dD}{dh}\right) \cdot \frac{\partial Q}{\partial x} - \frac{D^2}{2 \cdot B \cdot |Q|} \cdot \frac{\partial^2 Q}{\partial x^2} = 0 \tag{K.11}$$

with B: width of the bottom of the transversal profile [L]; D: discharge rate [L^3/T].

The discharge rate is the capacity of a cross section of a channel to transport a certain flow and is defined as:

$$Q = D \cdot {J_0}^{1/2} \tag{K.12}$$

Equation K.11 is an equation with partial derivatives of parabolic type which represents the convection and the diffusion of the variable Q. Hence, the flow transported with a velocity c (equation K.13) and diffused with a diffusion coefficient δ (equation K.14):

$$c = \frac{Q}{B \cdot D} \cdot \frac{dD}{dh} \tag{K.13}$$

$$\delta = \frac{D^2}{2B \cdot |Q|} \tag{K.14}$$

Based on the hypothesis of a clearly defined relation between the flow Q and the water level h, equation K.11 is reduced to:

$$\frac{\partial Q}{\partial t} + \left(\frac{dQ}{dA}\right)_{x_0} \cdot \frac{\partial Q}{\partial x} = 0 \tag{K.15}$$

This equation is called « equation of the kinematic wave » and describes the simple convection of the flow with a velocity c according to equation K.13. It can be solved by the following numerical finite difference scheme:

$$\frac{\partial Q}{\partial t} \approx \frac{X \cdot \left({Q_j}^{n+1} - {Q_j}^{n}\right) + (1 + X) \cdot \left({Q_{j+1}}^{n+1} - {Q_{j+1}}^{n}\right)}{\Delta t} \tag{K.16}$$

$$\frac{\partial Q}{\partial x} \approx \frac{\frac{1}{2} \cdot \left({Q_{j+1}}^{n+1} - {Q_j}^{n+1}\right) + \frac{1}{2} \cdot \left({Q_{j+1}}^{n} - {Q_j}^{n}\right)}{\Delta x} \tag{K.17}$$

Applying this scheme to equationK.15 yields:

$$\frac{X \cdot \left({Q_j}^{n+1} - {Q_j}^{n}\right) + (1 + X) \cdot \left({Q_{j+1}}^{n+1} - {Q_{j+1}}^{n}\right)}{c\Delta t} + \frac{\frac{1}{2} \cdot \left({Q_{j+1}}^{n+1} - {Q_j}^{n+1}\right) + \frac{1}{2} \cdot \left({Q_{j+1}}^{n} - {Q_j}^{n}\right)}{\Delta x} = 0 \tag{K.18}$$

We can express the solution of this equation as a function of the unknown variable Q_{j+1}^{n+1} namely:

$$Q_{j+1}{}^{n+1} = C_1 \cdot Q_j{}^{n+1} + C_2 \cdot Q_j{}^{n} + C_3 \cdot Q_{j+1}{}^{n} \tag{K.19}$$

with:

$$\begin{aligned} C_1 &= -\frac{KX - \frac{\Delta t}{2}}{K(1-X) + \frac{\Delta t}{2}} \\ C_2 &= \frac{KX + \frac{\Delta t}{2}}{K(1-X) + \frac{\Delta t}{2}} \\ C_3 &= \frac{K(1-X) - \frac{\Delta t}{2}}{K(1-X) + \frac{\Delta t}{2}} \end{aligned} \tag{K.20}$$

$$K = \frac{\Delta x}{c} \tag{K.21}$$

$$c = \frac{Q_{j+1}{}^{n} - Q_j{}^{n}}{A_{j+1}{}^{n} - A_j{}^{n}} \tag{K.22}$$

Here, hydraulic engineers might recognize the equation of Muskingum (Boillat, 1980) corresponding to the name of the river localized in the United States where the method was employed for the first time. The Muskingum method represents an approximation by finite differences of the equation for the kinematic wave. This is not only an appearance since developing the terms of equation K.18 in terms of a Taylor series around the point (j, n) assuming Δx/Δt = c and neglecting the quadratic terms (Δx2) the equation can be written as follows:

$$\frac{\partial Q}{\partial t} + c\frac{\partial Q}{\partial x} - c \cdot B\frac{\partial^2 Q}{\partial x^2} = 0 \tag{K.23}$$

$$B = \Delta x \cdot \left(\frac{1}{2} - X\right) \tag{K.24}$$

According to this analysis proposed by Cunge (1969) it can be recognized that the Muskingum equation is a solution in terms of finite differences of the equation of the diffusive wave (K.11) under the condition of correctly introducing the value of the parameters K and X. K is defined by equation K.21 and according to the celerity c (equation K.24) X corresponds to:

$$X = \frac{1}{2} - \frac{D^3}{2 \cdot \Delta x \cdot |Q| \cdot Q \cdot \frac{dD}{dh}} \tag{K.25}$$

This function of the diffusive wave implemented at present in RS MINERVE is capable of solving the Muskingum-Cunge equation for the trapezoidal geometry of a transversal profile according to Figure 11.

Kinematic Wave routing

In the kinematic wave model the terms of inertia and pressure of the St. Venant equations are supposed to be negligible. As a consequence, the cinematic hypothesis supposes that the gravity forces are identical, though with an opposite sign, to the friction forces. This implies that there is an explicit relationship between the flow and the water level (measured normal water depth).

The equation of the cinematic wave as presented in the previous chapter is presented as follows:

$$\frac{\partial Q}{\partial t} + \left(\frac{dQ}{dA}\right)_{x_0} \cdot \frac{\partial Q}{\partial x} = 0 \qquad \text{(K.26)}$$

This is a simple equation of convection which indicates that the flow Q is transported downstream with a celerity c which is defined as:

$$c = \frac{\partial Q}{\partial A} \qquad \text{(K.27)}$$

This rather simple model transports each point of the hydrograph from upstream to downstream with a velocity c. Since no diffusive term appears in the equation the peak discharge remains constant and is not reduced. On the contrary, the general behaviour of a flood is modified, since high discharges are transferred downstream more rapidly than small ones.

The initial parameters are identical to those of the model of the diffusive wave. The geometry for the transversal profiles also corresponds to the same than for the Muskingum-Cunge method (trapezoidal channels, Figure 11). Opposite to the model of the diffusive wave, no reduction of the flood is produced as mentioned previously. The solution is performed according to the following equations:

$$c = \frac{{Q_{j+1}}^{n} - {Q_j}^{n}}{{A_{j+1}}^{n} - {A_j}^{n}} \qquad \text{(K.28)}$$

$$\alpha = c \cdot \frac{\Delta t}{\Delta L} \qquad \text{(K.29)}$$

$$\begin{aligned} {Q_{j+1}}^{n+1} &= \alpha \cdot {Q_j}^{n} + (1-\alpha) \cdot {Q_{j+1}}^{n} && si \quad \alpha \leq 1 \\ {Q_{j+1}}^{n+1} &= \frac{1}{\alpha} \cdot {Q_j}^{n} + (1-\frac{1}{\alpha}) \cdot {Q_j}^{n+1} && si \quad \alpha > 1 \end{aligned} \qquad \text{(K.30)}$$

Lag-time

The lag-time model is the simpler routing model where upstream and downstream flows are delayed by a fixed lag time called *Lag* (in minutes).

2.11. Time series

In this object, temporary series of flow, precipitation, temperature or ETP can be directly introduced. The time is incorporated in seconds and the associated values in their corresponding units, in tabular form « t [s] – value [depending on the series] ».

Table 10 Time series required data

Object	Name	Units	Description
Time Series	Series (paired data)	s - (depending on the series)	Time – Value series

2.12. Reservoir

The transient evolution of a water volume ∀ in a reservoir is described by the following retention equation:

$$\frac{d\forall}{dt} = Q_e - Q_s \tag{L.1}$$

with ∀: volume in the reservoir [L^3]; Qe: inflow in the reservoir [L^3/T]; Qs: outflow [L^3/T].

Usually, reservoirs are equipped with turbines, pumps and spillways depending on the water level in the reservoir. To solve equation L.1, it is necessary to know the outflow as a function of the water volume ∀ in the reservoir. This operation is possible if relation between the water level and the water volume is known (Table 11).

Table 11 Reservoir paired data and initial condition required

Object	Name	Units	Description
Reservoir	H-V (paired data)	masl - m^3	Level - Volume relation
	Hini	masl	Initial level in the reservoir

2.13. Level-Discharge Relation (HQ)

The HQ object provides an outflow depending on a level in a reservoir. The outflow is calculated by the help of a « Level – Discharge » relation (Table 12). Hence, the discharge in then calculated as follows:

$$
\begin{aligned}
&{Q_{outflow}}^{n} = 0 && if \;\; H^{n} < H_{1} \\
&{Q_{outflow}}^{n} = Q_{dev,i} + (H_{i+1} - H^{n}) \cdot \frac{Q_{outflow,i+1} - Q_{outflow,i}}{H_{i+1} - H_{i}} && if \;\; H_{i} < H^{n} < H_{i+1}
\end{aligned}
\tag{M.1}
$$

with $Q_{outflow}^{n}$: discharge at instant n [L^3/T]; H^n: water level at instant n [L]; $Q_{outflow,i}$: discharge flow for a water level H_i [L^3/T]; H_i: reservoir water level [L].

Table 12 HQ paired data required

Object	Name	Units	Description
HQ	H-Q (paired data)	masl - m^3/s	Level - Discharge relation

2.14. Turbine

In this object, temporary series of flow can be directly introduced to provide an outflow in a reservoir. The time is incorporated in seconds and the associated values in their corresponding units, in tabular form « t [s] – discharge». The final discharge also depends on the level reservoir as presented in next equations:

$$\begin{aligned} &Q_{turbine}{}^{n} = Q_{wanted}{}^{n} && if \quad (H^n > H_{on})\ or\ (\,H_{off} \leq\ H^n \leq H_{on}\ and\ IsOperating = 1) \\ &Q_{turbine}{}^{n} = 0 && if \quad \left(H^n < H_{off}\right) or\ (\,H_{off} \leq\ H^n \leq H_{on}\ and\ IsOperating = 0) \end{aligned} \qquad \text{(N.1)}$$

with $Q_{turbine}{}^{n}$: discharge at instant n [L^3/T]; H^n: water level at instant n [L]; $Q_{owanted}{}^{n}$: discharge flow wanted at instant n [L^3/T]; Hon: Threshold in the level of the reservoir to start the turbine cycle [L]; Hoff: Threshold in the level of the reservoir to stop the turbine cycle [L]; IsOperating: Planned turbine cycle at instant t [0/1].

Table 13 Time-Q paired data, parameters and initial conditions required

Object	Name	Units	Description
HQ	H-Q (paired data)	masl - m^3/s	Level - Discharge relation
	Hon	masl	Threshold in the level of the reservoir to start the turbine cycle (turbine starts when the threshold is exceeded)
	Hoff	masl	Threshold in the level of the reservoir to stop the turbine cycle (turbine stops when level go below the threshold)
	IsOperatingIni	0/1	First suggested value for the turbine cycle (0 = not turbine; 1 = turbine). Only taken into account if *Hoff > H > Hon*

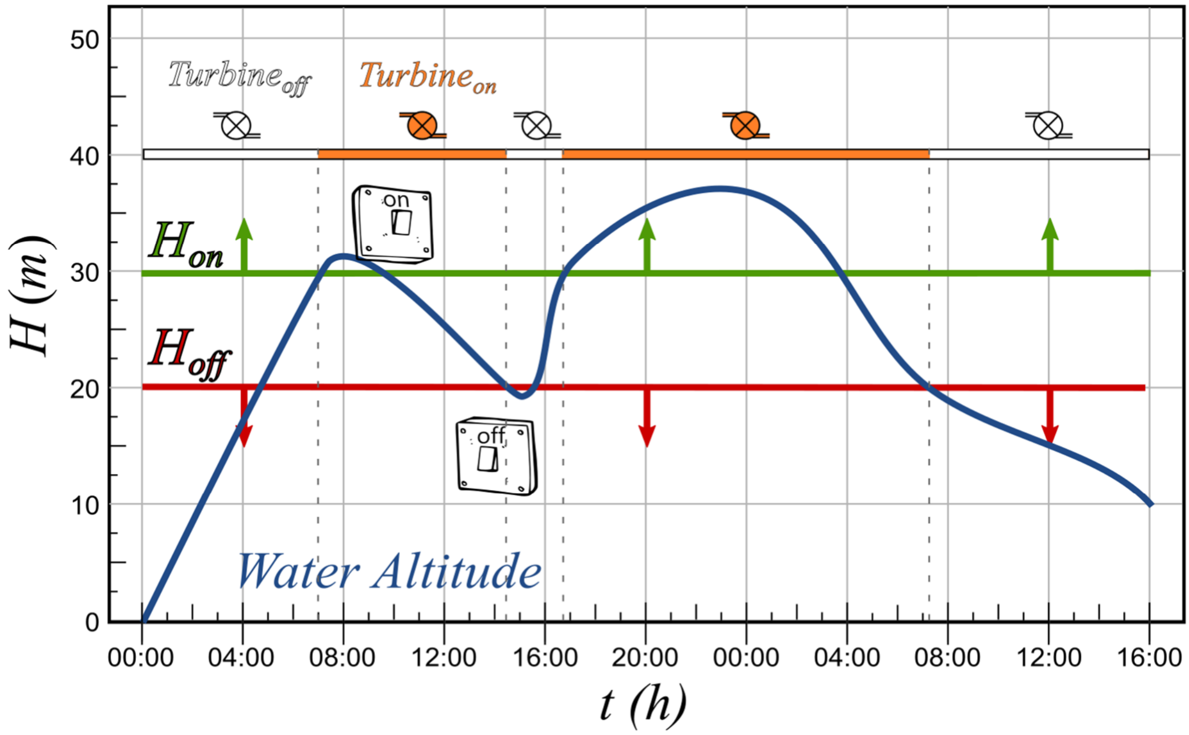


Figure 12 Time-H, tubine start and stop operation.

2.15. Hydropower

The inputs of this object are a reservoir level and a discharge. It also needs the paired data relation « Discharge (Q) – Performance (η) » for the turbine, the altitude ($Z_{central}$) of the hydropower plant, the length (L), the diameter (D) and the roughness (K) of the pipe as well as the kinematic viscosity of the fluid (υ).

As outputs, the object calculates not only the power and the revenue for each time step, as presented in equations N.1 to N.3, but also the total energy produced and the total obtained revenue. For this calculation, it takes into account the head loss in terms of volumetric flow rate in a full-flowing circular pipe from the Darcy-Weisbach equation (Darcy, 1857; Simmons, 2008).

$$Z_{net}{}^{n} = (Z_{water}{}^{n} - Z_{central}) - f \cdot \frac{8 \cdot L \cdot Q^{n^2}}{g \cdot \pi^2 \cdot D^5} \tag{O.1}$$

$$Power^{n} = \eta^{n} \cdot 1000 \cdot Q^{n} \cdot g \cdot Z_{net}{}^{n} \tag{O.2}$$

$$\mathrm{Revenue}^{n} = \mathrm{Power}^{n} \cdot \mathrm{Price}^{n} \tag{O.3}$$

with $H_{net}{}^{n}$: net height at instant n [L]; $Z_{water}{}^{n}$: water height in the reservoir at instant n [L]; $Z_{central}$: hydropower plant altitude [L]; f: friction factor [-]; L: length of the pipe [L]; Q^{n}: discharge at instant n [L^3/T]; g: gravity, 9.81 [L^2/T]; D: diameter [L]; $Power^{n}$: power at instant n [Watt]; η: performance of the turbine at instant n [%]; $Revenue^{n}$: Revenue of the turbine at instant n [€/Kwh]; $Price^{n}$: Price of the energy at instant n [€/Kwh].

If the discharge actually provided to the object Hydropower is higher than the range proposed in the relation Q-η of this object, the maximum discharge of the paired data Q-η is taken for the energy production calculations.

Table 14 Hydropower paired data and parameters required

Object	Name	Units	Description
Hydropower	Q-η (paired data)	m^3/s - %	Discharge-Performance relation
	Zplant	masl	Hydropower plant altitude
	L	m	Length of the pipe
	D	m	Diameter of the pipe
	K	m	Roughness
	ν	m^2/s	Kinematic viscosity
	Default Price	euro/Kwh	Default price, only used if no data exists in the database

A default price can be introduced for first approximation of the revenue. It is used only if the object does not have a price series in the database.

For the calculation of the friction factor f of the Darcy-Weisbach equation, the equation of Colebrook-White (Colebrook and White, 1937; Colebrook, 1939) is used. It is presented in equation N.4, where λ represents f. In addition, the Reynolds number, Re, is presented in equation N.5.

$$\frac{1}{\sqrt{\lambda}} = -2 \cdot \log_{10}\left(\frac{k/D}{3.7} + \frac{2.51}{\mathrm{Re} \cdot \sqrt{\lambda}}\right) \tag{O.4}$$

$$\mathrm{Re} = v \cdot \frac{D}{v} \tag{O.5}$$

with λ: friction factor [-]; k: roughness [L]; Re: Reynolds number [-];v: velocity [L/T]; υ: kinematic viscosity [L^2/T]

The equation N.4 is solved for the range of discharges of the paired data Q-η, providing the paired data relation « Discharge (Q) – Friction factor (λ) » for user information.

2.16. Diversion

This object needs the paired data relation « Inflow – Diverted flow » as information as well as the incoming hydrograph. The relation describes the behaviour of the diversion and is generated by the user, who performs a calculation for the behaviour of the diversion and then creates the relation « Inflow – Diverted flow ».

With this information, the Diversion calculates the diverted hydrograph and the downstream hydrograph, as presented in equation M.1:

$$Q_{up}{}^{n} = Q_{down}{}^{n} + Q_{diverted}{}^{n} \tag{P.1}$$

with $Q_{up}{}^{n}$: total flow upstream at instant n [L^3/T]; $Q_{diverted}{}^{n}$: diverted flow at instant n [L^3/T]; $Q_{down}{}^{n}$: downstream flow at instant n [L^3/T].

Table 15 Diversion paired data required

Object	Name	Units	Description
Diversion	Q_{up}-$Q_{diverted}$ (paired data)	m^3/s - m^3/s	Upstream flow - Diverted flow relation

2.17. Consumer

The Consumer object is used to take into account a consumption site (agriculture, cities, etc.). A series in the database is used as the demand of the Consumer object. If no information about consummation exists in the database, the parameter “Default QDemand” is used for the whole period of the simulation as uniform demand.

The LossRate parameter provides the distribution losses of a demand site (physical leaks, clandestine connections, etc.). If the input discharge (Qup) is sufficient, the delivered flow (Qdelivered) is equal to the Qdemand and the supply requirement (Qsupplied) is equal to the Qdemand divided by (1-LossRate), otherwise the supply requirement is equal to the input discharge (Qup). Then, the ConsumptionRate parameter is used to determine the discharge consumed by the demand site and which is lost (evaporation, embodied products, etc). The discharge remainder (Qreturn) is returned to the main system. The output flow is equal to the input flow minus the losses and the effective consumption (Qconsumed).

The Consumer object calculates the losses, the consumed discharge as well as the downstream hydrograph, as presented in next equations. At the same time, the object also calculates the unmet demand and the demand coverage (%) series during the simulation period.

$$Q_{supplied}{}^{n} = min\left(\frac{Q_{demand}{}^{n}}{1 - LossRate}, Q_{up}{}^{n}\right) \tag{Q.1}$$

$$Q_{stream}{}^{n} = Q_{up}{}^{n} - Q_{supplied}{}^{n} \tag{Q.2}$$

$$Q_{losses}{}^{n} = Q_{supplied}{}^{n} \cdot LossRate \tag{Q.3}$$

$$Q_{delivered}{}^{n} = Q_{supplied}{}^{n} \cdot (1 - LossRate) \tag{Q.4}$$

$$Q_{UnmetDemand}{}^{n} = Q_{demand}{}^{n} - Q_{delivered}{}^{n} \tag{Q.5}$$

$$DemandCoverage^{n} = \frac{Q_{demand}{}^{n} - Q_{delivered}{}^{n}}{Q_{demand}{}^{n}} \cdot 100 \tag{Q.6}$$

$$Q_{consumed}{}^{n} = Q_{delivered}{}^{n} \cdot ConsumptionRate \tag{Q.7}$$

$$Q_{return}{}^{n} = Q_{delivered}{}^{n} \cdot (1 - ConsumptionRate) \tag{Q.8}$$

$$Q_{down}{}^{n} = Q_{up}{}^{n} - Q_{losses}{}^{n} - Q_{consumed}{}^{n} \tag{Q.9}$$

with $Q_{demand}{}^{n}$: demanded consumption at instant n [L^3/T]; $Q_{up}{}^{n}$: total flow upstream at instant n [L^3/T]; LossRate: fraction of the $Q_{supplied}$ lost at the demand site [-]; $Q_{supplied}{}^{n}$: supplied discharge at instant n [L^3/T]; $Q_{stream}{}^{n}$: discharge that flows downstream after the initial supply at instant n [L^3/T]; $Q_{losses}{}^{n}$: part of the $Q_{supplied}$ discharge that is lost at instant n [L^3/T]; LossRate: fraction of the $Q_{supplied}$ discharge that will be lost before its delivery [-]; $Q_{delivered}{}^{n}$: part of the $Q_{supplied}$ discharge that is not lost at instant n [L^3/T]; $Q_{UnmetDemand}{}^{n}$: discharge shortfall at the consumption site at instant n [L^3/T]; DemandCoverage: percentage of the demand covered by the delivered discharge at instant n [%]; ConsumptionRate: fraction of the $Q_{delivered}$ ultimately consumed [-]; $Q_{consumed}{}^{n}$: discharge consumed at the consumption site at instant n [L^3/T]; $Q_{return}{}^{n}$: discharge returning downstream from the consumption site at instant n [L^3/T]; $Q_{down}{}^{n}$: downstream flow at instant n [L^3/T].

Table 16 Consumer parameters required

Object	Name	Units	Description
Consumer	LossRate	-	Fraction of the $Q_{supplied}$ lost at the demand site
Consumer	ConsumptionRate	-	Fraction of the $Q_{delivered}$ ultimately consumed
Consumer	Default QDemand	m^3/s	Default demand of consummation, only used if no data exists in the database

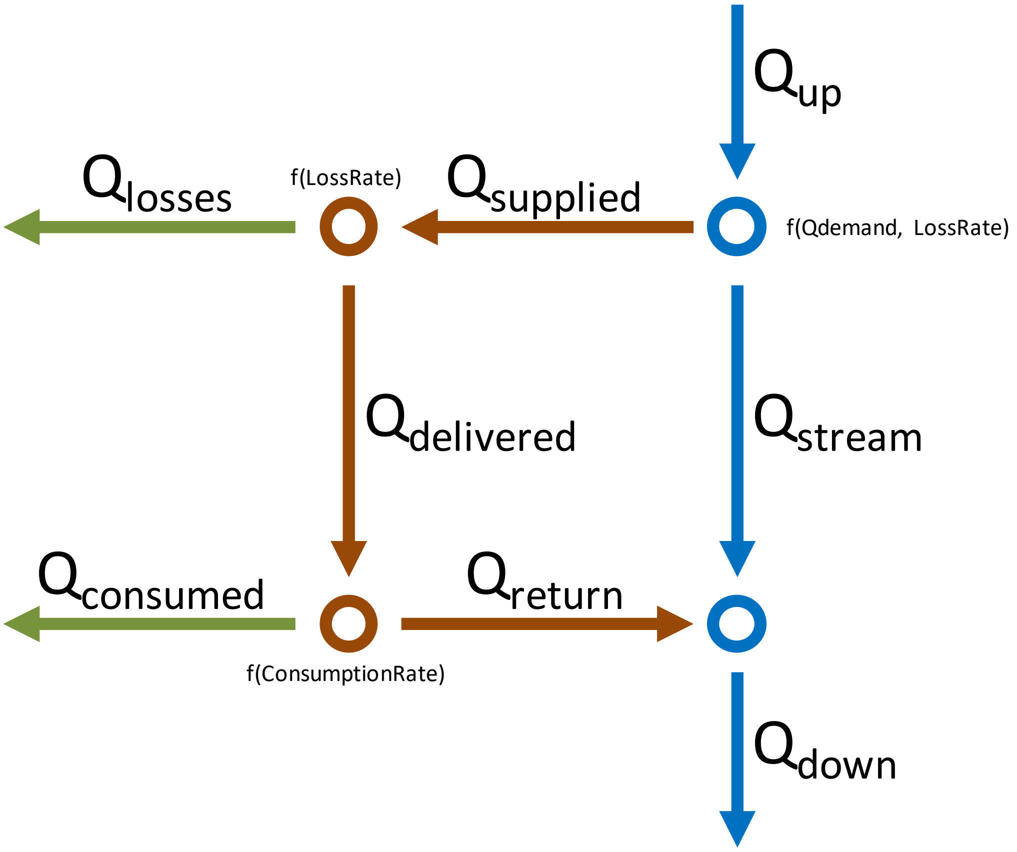


Figure 13 Distribution of discharges in the Consumer object.

2.18. Structure efficiency

This object needs an efficiency coefficient as information to describe the efficiency of a structure such a canal or a pipe, as well as the incoming hydrograph.

An efficiency of 1 provides an output flow equivalent to the input hydrograph. An efficiency of 0 generates a complete loss of the input.

The downstream hydrograph is calculated as presented in equation Q.1 and Q.2:

$$Q_{down}{}^{n} = Q_{up}{}^{n} \cdot Efficiency \tag{R.1}$$

$$Q_{lost}{}^{n} = Q_{up}{}^{n} \cdot (1 - Efficiency) \tag{R.2}$$

with $Q_{up}{}^{n}$: total flow upstream at instant n [L^3/T]; $Q_{lost}{}^{n}$: lost flow at instant n [L^3/T]; $Q_{down}{}^{n}$: downstream flow at instant n [L^3/T].

Table 17 Structure efficiency parameter required

Object	Name	Units	Description
Structure Efficiency	Efficiency	-	Efficiency of the structure

2.19. Planner

When a planner object is selected, a manage system is generated. Inside this object, several Rules can be created. All the rules will be computed, from first one to last one. In each rule define several Conditions is possible and also create combinations between them (with AND or OR operators). The rule will be applied if the combined condition (or an individual condition) is satisfied.

Table 18 Operators to combine rule conditions.

Operator	Combined condition is satisfied when
OR	Left condition OR Right condition are satisfied
AND	Left condition AND Right condition are satisfied

A typical example of planner is the implementation of a turbine / pump law as a function of the water level in the reservoir. Thus, some Rules should be created with different conditions regarding to reservoir levels or, in some cases, to results from other objects.

Table 19 Planner required data

Object	Name	Units	Description
Planner	Rule	-	Rule to be applied regarding at least one condition
Rules	Condition	-	Expression to be evaluated and if it is satisfied the parent rule will be applied

The possible operators to use inside the conditions are presented in Table 19.

Table 20 Operators for rule conditions.

Operator	Definition
>	Bigger than
>=	Bigger than or equal to
=	Equal to
< =	Smaller than or equal to
<	Smaller than

2.20. Comparator

The Comparator object calculates ten performance indicators by comparing a simulated and a reference (observed) timeseries. The warm up period parameter (in days) is used for the initialization of the model state variables. Note that this time period starts at the beginning of and must be shorter than the simulation period and is not used to compute the performance indicators. The description of each indicator is presented in Chapter 3.

Moreover, for the calculation of threshold-based indicators (Peirce Skill Score and Overall Accuracy), the definition of both the reference and the simulation thresholds is required (Table 21).

Table 21 Comparator's parameters required

Object	Name	Units	Description
Comparator	Reference threshold	(depends on the variable)	Threshold used by threshold-based indicators (returns TRUE if reference value > threshold)
	Simulation threshold	(depends on the variable)	Threshold used by threshold-based indicators (returns TRUE if simulation value > threshold)
	Warm up period	Days	Warm up period, not used for indicators computation

Chapter 3. Performance indicators

The indicators calculated by the Comparator object are based on observed and simulated values. Since the Comparator object is able to compare variables of type flow, height, altitude, power and True/False, the descriptions below only refer to these 2 variables.

3.1. Nash coefficient

The Nash-Sutcliffe criteria (Nash and Sutcliffe, 1970) is used to assess the predictive power of hydrological models (Ajami et al., 2004; Schaefli and al, 2005; Jordan, 2007; Viviroli et al., 2009; García Hernández et al., 2011). It is defined as presented in Eq. IND.1.

$$Nash = 1 - \frac{\sum_{t=t_i}^{t_f}(X_{sim,t} - X_{ref,t})^2}{\sum_{t=t_i}^{t_f}(X_{ref,t} - \overline{X}_{ref})^2} \qquad \text{IND.1}$$

with Nash: Nash-Sutcliffe model efficiency coefficient [-]; $X_{sim,t}$: simulated variable (discharge [L^3/T] or height [L]) at time t; $X_{ref,t}$: observed variable (discharge [L^3/T] or height [L]) at time t; $\overline{X}_{ref}$: average observed variable (discharge [L^3/T] or height [L]) for the considered period.

It varies from -∞ to 1, with 1 representing the best performance of the model and zero the same performance than assuming the average of all the observations at each time step.

3.2. Nash coefficient for logarithm values

The Nash-Sutcliffe coefficient for logarithm flow values (Nash-ln) is used to assess the hydrological models performance for low values (Krause et al., 2005; Nóbrega et al., 2011). It is defined as presented in Eq. IND.2.

$$Nash\text{-}ln = 1 - \frac{\sum_{t=t_i}^{t_f}(\ln(X_{sim,t}) - \ln(X_{ref,t}))^2}{\sum_{t=t_i}^{t_f}(\ln(X_{ref,t}) - \ln(\overline{X}_{ref}))^2} \qquad \text{IND.2}$$

with Nash-ln: Nash-Sutcliffe coefficient for log values [-].

It varies from -∞ to 1, with 1 representing the best performance of the model.

3.3. Pearson Correlation Coefficient

The Pearson correlation coefficient shows the covariability of the simulated and observed values without penalizing for bias (AghaKouchak and Habib, 2010; Wang et al., 2011). It is defined as presented in Eq. IND.3.

$$Pearson = \frac{\sum_{t=t_i}^{t_f}(X_{sim,t} - \overline{X}_{sim}) \cdot (X_{ref,t} - \overline{X}_{ref})}{\sqrt{\sum_{t=t_i}^{t_f}(X_{sim,t} - \overline{X}_{sim})^2 \cdot \sum_{t=t_i}^{t_f}(X_{ref,t} - \overline{X}_{ref})^2}} \qquad \text{IND.3}$$

with Pearson: Pearson Correlation Coefficient [-];$\overline{X}_{sim}$: average simulated variable (discharge [L^3/T] or height [L]) for the considered period.

It varies from -1 to 1, with 1 representing the best performance of the model.

3.4. Kling-Gupta Efficiency

The Kling-Gupta efficiency (Gupta et al., 2009) provides an indicator which facilitates the global analysis based on different components (correlation, bias and variability) for hydrological modelling issues.

Kling et al. (2012) proposed a revised version of this indicator, to ensure that the bias and variability ratios are not cross-correlated. This update is proposed as indicator in RS MINERVE (Eq. IND.4):

$$KGE' = 1 - \sqrt{(\mathrm{r}-1)^2 + (\upbeta - 1)^2 + (\upgamma - 1)^2} \qquad \text{IND.4}$$

$$\beta = \frac{\mu_s}{\mu_o} \qquad \text{IND.5}$$

$$\gamma = \frac{CV_s}{CV_o} = \frac{\sigma_s/\mu_s}{\sigma_o/\mu_o} \qquad \text{IND.6}$$

with KGE': modified KGE-statistic [-]; r: correlation coefficient between simulated and reference values [-]; β: bias ratio; γ: variability ratio [-]; μ: mean discharge [L^3/T]; CV: coefficient of variation [-]; σ: standard deviation of discharge [L^3/T]; the indices s and o indicate respectively simulated and observed discharge values.

It varies from 0 to 1, with 1 representing the best performance.

3.5. Bias Score

The Bias Score (BS) is a symmetric estimation of the match between the average simulation and average observation (Wang et al., 2011). It is defined as presented in Eq. IND.7.

$$BS = 1 - \left[\max\left(\frac{\overline{X}_{sim}}{\overline{X}_{ref}}, \frac{\overline{X}_{ref}}{\overline{X}_{sim}}\right) - 1\right]^2 \qquad \text{IND.7}$$

with BS: Bias Score [-].

It varies from -∞ to 1, with 1 representing the best performance of the model.

3.6. Relative Root Mean Square Error

The Relative Root Mean Square Error (RRMSE) is defined as the RMSE normalized to the mean of the observed values (Feyen et al., 2000; El-Nasr et al., 2005; Heppner et al., 2006) and is presented in Eq. IND.8.

$$RRMSE = \frac{\sqrt{\frac{\sum_{t=t_i}^{t_f}(X_{sim,t} - X_{ref,t})^2}{n}}}{\overline{X}_{ref}} \qquad \text{IND.8}$$

with RRMSE: relative RMSE [-]; n: number of values [-].

It varies from 0 to +∞. The smaller RRMSE, the better the model performance is.

3.7. Relative Volume Bias

The Relative Volume Bias (RVB), sometimes called differently, corresponds in this case to the relative error between the simulated and the observed volumes during the studied period (Ajami and al, 2004; Schaefli and al, 2005; Moriasi et al., 2007; AghaKouchak and Habib, 2010) according to Eq. IND.9. This indicator is envisaged for the comparison between observed and simulated discharges.

$$RVB = \frac{\sum_{t=t_i}^{t_f}(X_{sim,t} - X_{ref,t})}{\sum_{t=t_i}^{t_f}(X_{ref,t})} \qquad \text{IND.9}$$

with RVB: relative volume bias between forecast and observation for the considered period [-]; X usually corresponding to the discharge variable.

The RVB varies from -1 to +∞. An index near to zero indicates a good performance of the simulation. Negative values are returned when simulated variable is, in average, smaller than the average of the observed one (deficit model), while positive values mean the opposite (overage model).

3.8. Normalized Peak Error

The Normalized Peak Error (NPE) indicates the relative error between the simulated and the observed maximum values (Masmoudi and Habaieb, 1993; Sun and al, 2000; Ajami and al, 2004; Gabellani and al, 2007). It is computed according to IND.10 TO IND.12.

$$NPE = \frac{S_{\max} - R_{\max}}{R_{\max}} \qquad \text{IND.10}$$

$$S_{\max} = \bigvee_{t=t_i}^{t_f} Q_{sim,t} \qquad \text{IND.11}$$

$$R_{\max} = \bigvee_{t=t_i}^{t_f} Q_{ref,t} \qquad \text{IND.12}$$

with NPE: relative error between simulated and observed peak value [-]; S_{max} : maximum simulated value (discharge [L^3/T] or height [L]) for the studied period; R_{max} : maximum observed value (discharge [L^3/T] or height [L]) for the studied period.

The NPE varies from -1 to +∞. Negative values are returned when maximum simulated value is below the observed one, while positive values mean the opposite. Values near to zero indicate a good performance of simulated peaks regarding observed ones.

Warning : The indicator is computed over the entire simulation period and the absolute maximum of the simulated and the observed peaks are considered! This indicator should therefore be used with care when simulating over long periods of time.

3.9. Peirce Skill Score

The Peirce Skill Score (PSS) indicates the performance of the model to reproduce the overrun of a threshold (Peirce, 1884; Manzato, 2007). Based on a contingency table definging the number of cases where the simulation and the obseration exceed or not the threshold, the PSS is computed according to IND.13.

$$PSS = \frac{\mathrm{ad} - \mathrm{bc}}{(\mathrm{a} + \mathrm{c})(\mathrm{b} + \mathrm{d})} \qquad \text{IND.13}$$

with a the number of cases when both simulation and observation exceed the threshold defined in the Comparator (event), b the number of cases when the simulation exceeds the threshold but not the obseration (false), c the number of cases when the observation exceeds the threshold but not the simulation (miss) and d the number of cases when both simulation and obseration are below the threshold (nonevent).

Remark : If the denominerator equals 0 (division by 0), a value of 0 is returned for the PSS.

3.10. Overall Accuracy

The Overall Accurary (OA) indicates the performance of the model to reproduce the overrun of a threshold (Parajka and Blöschl, 2008). Based on the same contingency table as the Pierce Skill Score, the OA is computed according to IND.14.

$$OA = \frac{\mathrm{a} + \mathrm{d}}{\mathrm{a} + \mathrm{b} + \ \mathrm{c} + \mathrm{d}} \qquad \text{IND.14}$$

Chapter 4. Expert module - Calibration algorithms

4.1. Introduction to the calibration module

The module *Calibrator* of the *RS Expert* frame has been implemented for calibrating the parameters of the hydrological model. This module uses an objective function defined by the user and different algorithms to solve it.

The first algorithm, the Shuffled Complex Evolution – University of Arizona (SCE-UA), is a global optimization method (Duan et al., 1992, 1993) based on a synthesis of the best features from several existing algorithms, including the genetic algorithm, and introduces the concept of complex information exchange, so-called complex shuffling. The SCE-UA method was designed for solving problems encountered in conceptual watershed model calibration (Hapuarachchi H.A.P. et al., 2001; Ajami et al., 2004; Muttil and Liong, 2004; Blasone et al., 2007), but has also been satisfyingly used in water resources management (Zhu et al., 2006; Lin et al., 2008; Wang et al., 2010).

The second algorithm is a variation of the Adaptive Markov Chain Monte Carlo, used since it can be interesting for solving complex problems in high dimensional spaces (Gilks et al., 1996; Liu, 2001). It has been modified to an Uniform Adaptative Monte Carlo (UAMC) in this program to adjust the solution space after a defined group of simulations up to the convergence of the optimization. Variations of the Monte Carlo method are usually used in hydrological problem for parameterization optimization (Vrugt and al., 2003; Jeremiah and al., 2012).

The third and last algorithm used in RS MINERVE is the Coupled Latin Hypercube and Rosenbrock (CLHR) It couples the Latin Hypercube algorithm (McKay et al., 1979) with the Rosenbrock algorithm (Rosenbrock, 1960), generating a powerful tool for optimization of complex problems. The latin hypercube algorithm has been usually used in hydrology for sampling the initial parameter space, combined the with other methods (van Griensven et al, 2006; Kamali et al., 2013). Rosenbrock algorithm has been also used for hydrological parameters optimisation (Abbot et Refsgaard, 1996) or optimization of numerical functions (Kang et al., 2011).

4.2. Objective function

A flexible objective function (OF) has been developed for the module of calibration aiming to be adapted to the user's requirements. The indicators presented in Chapter 4 are used in this OF, each one weighted with a value defined by the user (Table 22).

Table 22 Weights of the indicators for the objective function

Indicator	Weight	Range of Values	Ideal value
Nash	w_1	-∞ to 1	1
Nash-ln	w_2	-∞ to 1	1
Pearson Correlation Coefficient	w_3	-1 to 1	1
Kling-Gupta Efficiency (KGE)	w_4	-∞ to 1	1
Bias Score (BS)	w_5	-∞ to 1	1
Relative Root Mean Square Error (RRMSE)	w_6	0 to +∞	0
Relative Volume Bias (RVB)	w_7	-∞ to +∞	0
Normalized Peak Error (NPE)	w_8	-∞ to +∞	0
Peirce Skill Score (PSS)	w_9	-1 to 1	1
Overall Accuracy (OA)	w_{10}	0 to 1	1

The OF is presented in Eq. OF.1 and takes into account the ideal values of each indicator. Thus, the OF searches to maximize the first five and last two indicators (Nash, Nash-ln, Pearson, Kling-Gupta, BS, PSS and OA) since their ideal value is equal to the maximum possible value and, at the same time, to minimize the value or the absolute value for the three other indicators (RRMSE, RVB, NPE) since their ideal value corresponds to zero.

$$OF = \max(\text{Nash} \cdot w_1 + \text{Nashln} \cdot w_2 + \text{Pearson} \cdot w_3 + \text{KGE} \cdot w_4 + \text{BS} \cdot w_5 - \\ -R\text{RMSE} \cdot w_6 - |\text{RVB} \cdot w_7| - |\text{NPE} \cdot w_8| + \text{PSS} \cdot w_9 + \text{OA} \cdot w_{10}) \qquad \text{OF.1}$$

4.3. Shuffled Complex Evoluation – University of Arizona

Model architecture

The Shuffled Complex Evolution – University of Arizona (SCE-UA) method was developed to obtain the traditional best parameter set and its underlying posterior distribution within a single optimization run. The goal is to find a single best parameter set in the feasible space. It starts with a random sample of points distributed throughout the feasible parameter space, and uses an adaptation of the Simplex Downhill search scheme (Nelder and Mead, 1965) to continuously evolve the population toward better solutions in the search space, progressively relinquishing occupation of regions with lower posterior probability (Mariani et al., 2011).

A general description of the steps of the SCE-UA method is given below (Duan et al., 1994) and illustrated in Figure 14.

Step 1

Generate sample: Sample NPT points in the feasible parameter space and compute the criterion value at each point. In the absence of prior information on the location of the global optimum, use a uniform probability distribution to generate a sample.

Step 2

Rank points: Sort the NPT points to increase criterion value so that the first point represents the point with the lowest criterion value and the last the one with the highest criterion value (assuming that the goal is to minimize the criterion value).

Step 3

Partition into complexes: Partition the NPT points into NGS complexes, each containing NPG points. The complexes are partitioned in such a way that the first complex contains every NGS·(k-1)+1 ranked point, the second complex contains every NGS*(k-1)+2 ranked point, and so on, where k = 1,2,...,NPG.

Step 4

Evolve each complex: Evolve each complex independently by taking NSPL evolution steps, according to the Competitive Complex Evolution (CCE) algorithm. Figure 16 illustrates how each evolution step is taken.

Step 5

Shuffle complexes: Combine the points in the evolved complexes into a single sample population; sort the sample population in order of increasing criterion value; re-partition or shuffle the sample population into NGS complexes according to the procedure specified in the third step.

Step 6

Check convergence: If any of the pre-specified convergence criteria are satisfied, stop; otherwise, continue.

Step 7

Check complex number reduction: If MINGS (the minimum number of complexes) < NGS, remove the complex with the lowest ranked points; set NGS=NGS-1 and NPT=NGS*NPG; and return to Step 4. If MINGS=NGS, return to Step 4.

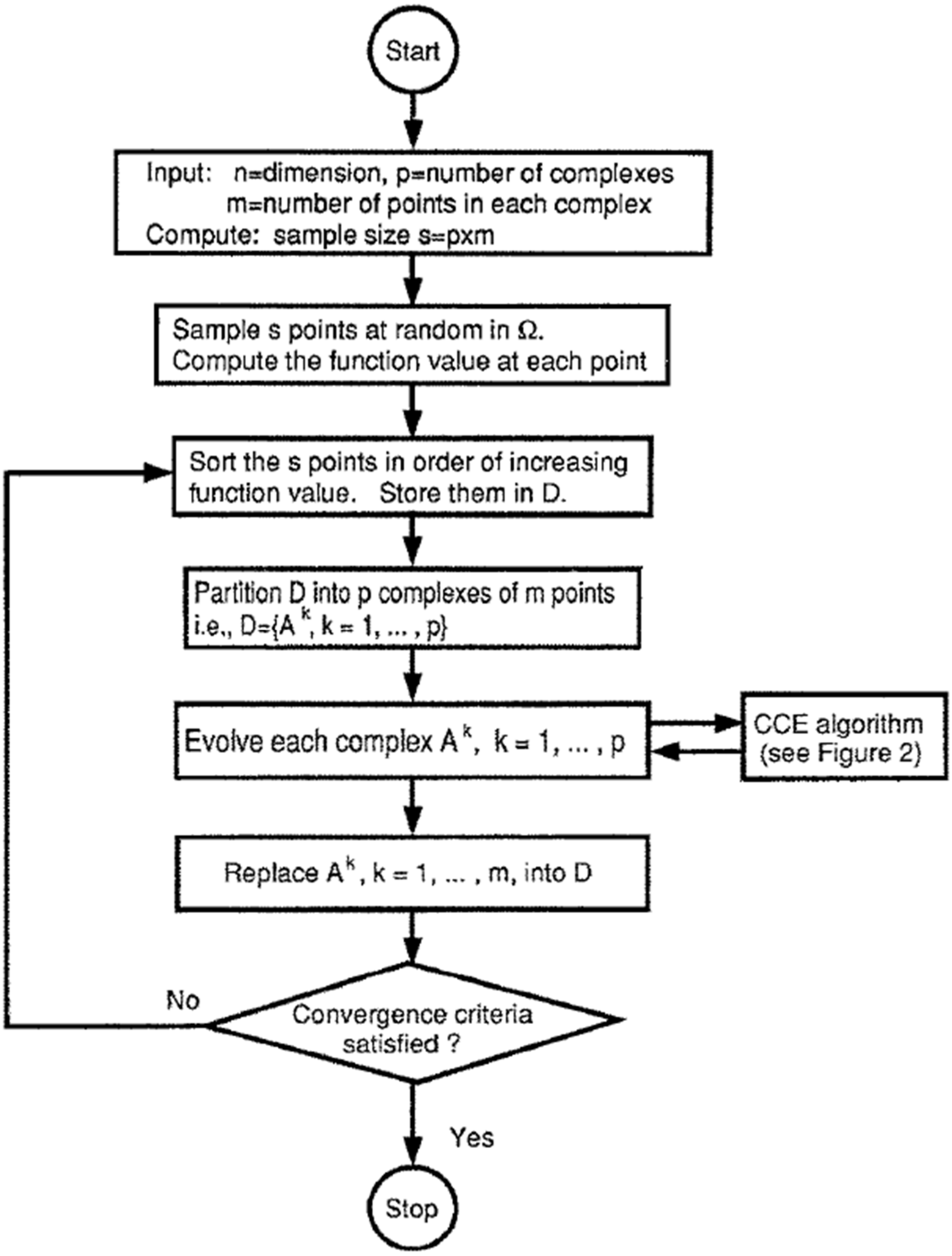

Start
Input: n=dimension, p=number of complexes
m=number of points in each complex
Compute: sample size s=pxm
Sample s points at random in Ω.
Compute the function value at each point
Sort the s points in order of increasing function value. Store them in D.
Partition D into p complexes of m points
i.e., D={A k, k = 1, ... , p}
Evolve each complex A k, k = 1, ... , p
CCE algorithm
(see Figure 2)
Replace A k, k = 1, ... , m, into D
Convergence criteria satisfied ?
No
Yes
Stop

Figure 14 Flow chart of the shuffled complex evolution method (from Duan et al., 1993), with V=n, NGS=p, NPG=m and NPT=s

The SCE-UA method is explained in Figure 15 and Figure 16 for a two dimensional case (Duan et al., 1994). The contour lines in Figure 15 and Figure 16 represent a function surface having a global optimum located at (4,2) and a local optimum located at (1,2). Figure 15a shows that a sample population containing NPT (=10) points is divided into NGS (=2) complexes. Each complex contains NPG (=5) points which are marked by • and * respectively. Figure 15b shows the locations of the points in the two independently evolved complexes at the end of the first cycle of evolution. It can be seen that one complex (marked by *) is converging towards the local optimum, while the other (marked by •) is converging toward the global optimum. The two evolved complexes are shuffled according to step 5. Figure 15c displays the new membership of the two evolved complexes after shuffling.

Figure 15d illustrates the two complexes at the end of the second cycle of evolution. It is clear that both complexes are now converging to the global optimum at the end of second cycle.

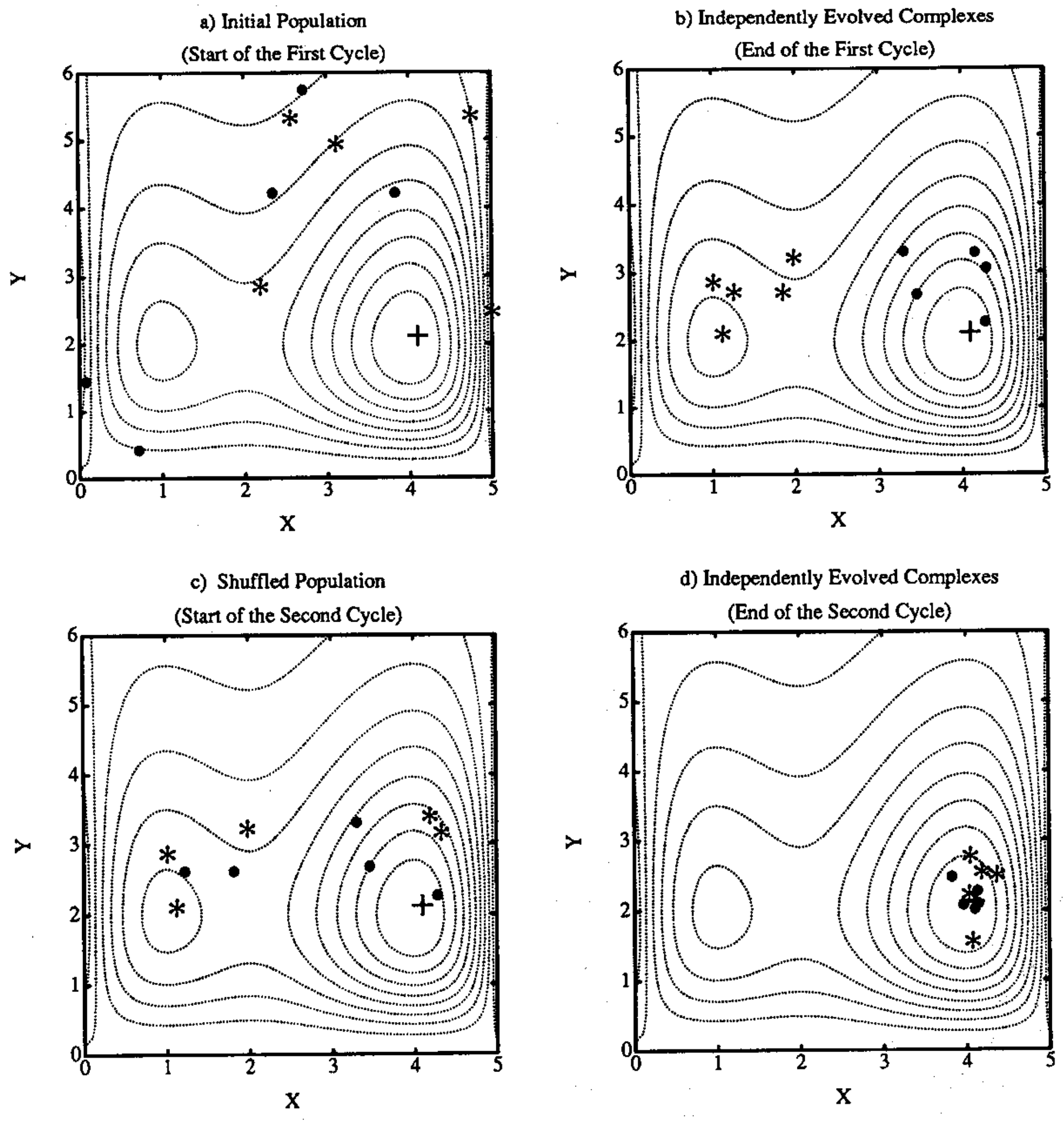


Figure 15 Illustration of the shuffled complex evolution (SCE-UA) method (from Duan et al., 1994).

The CCE algorithm is graphically illustrated in Figure 16. The black dots (•) indicate the locations of the points in a complex before the evolution step is taken. A sub-complex containing NPS (=3, i.e. forms a triangle in this case) points is selected according to a pre-specified probability distribution to initiate an evolution step.

The probability distribution is specified such that the better points have a higher chance of being chosen to form the sub-complex than the worse points. The symbol (*) represents the new points generated by the evolution steps. There are three types of evolution steps: reflection, contraction and mutation.

Figure 16a, Figure 16b and Figure 16d illustrate the "reflection" step, which is implemented by reflecting the worst point in a sub-complex through the centroid of the other points.

Since the reflected point has a lower criterion value than the worst point, the worst point is discarded and replaced by the new point. Thus an evolution step is completed.

In Figure 16c, the new point is generated by a "contraction" step (the new point lies half-way between the worst point and the centroid of the other points), after rejecting a reflection step for not improving the criterion value.

In Figure 16e, a "mutation" step is taken by random selection of a point in the feasible parameter space to replace the wrong point of the sub-complex. This is realized after a reflection step is attempted, but results in a wrong point, i.e. outside of the feasible parameter space. Another scenario in which a mutation step is taken is when both the reflection step and the contraction step do not improve the criterion value.

Finally, the Figure 16f shows the final complex after NSPL (=5) evolution steps.

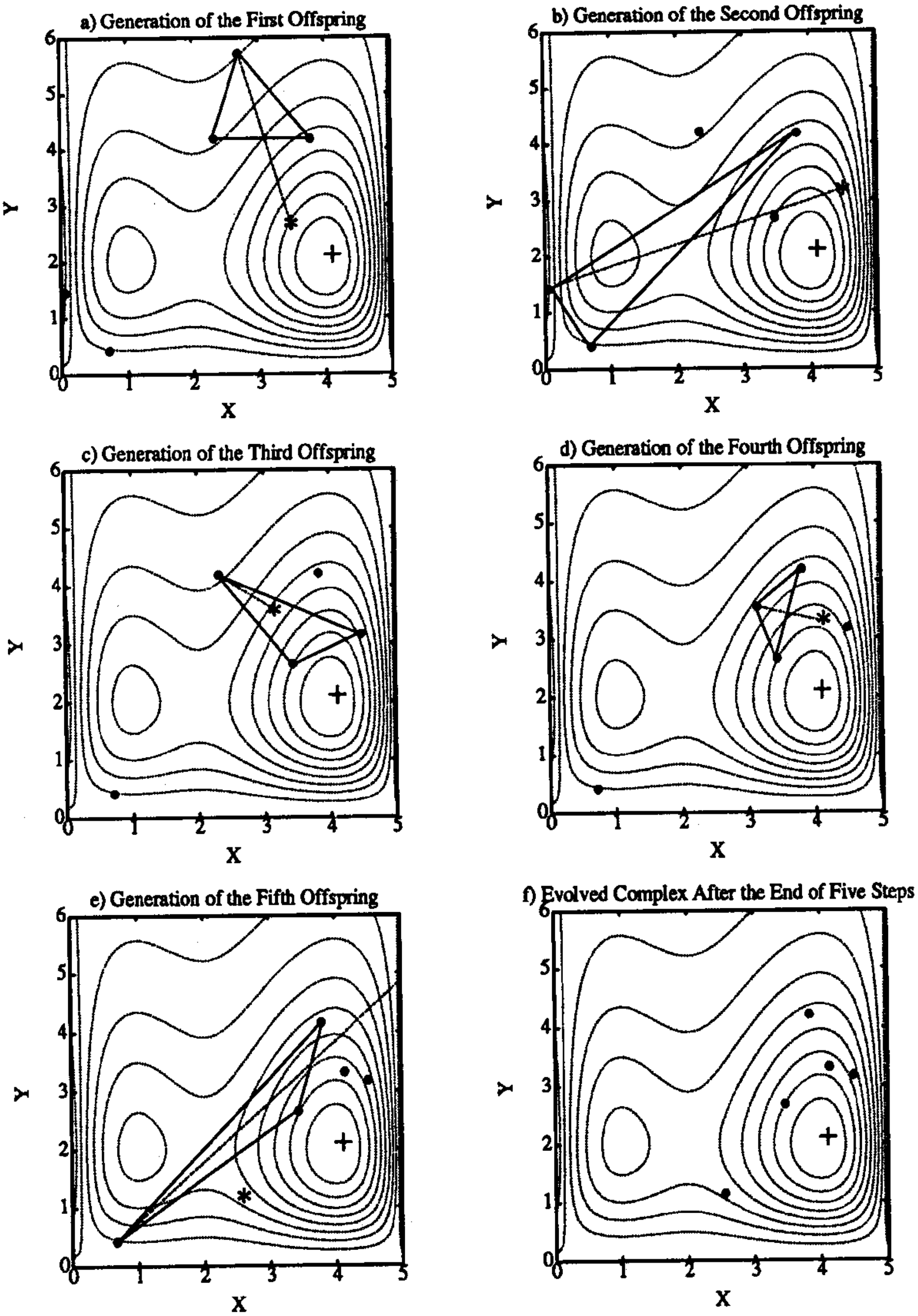


Figure 16 Illustration of the evolution steps taken by each complex (from Duan et al., 1994).

Algorithm parameters

Different parameters of the SCE-UA have to be defined by the user (Table 23), as presented hereafter, and other parameters are directly calculated by the process.

An initial set of **Nopt** parameters is given by the user or is assumed as random depending on the user's needs and the used hydrological models. The other NPT-1 points (or parameters sets) are randomly created by the algorithm, depending on a **SEED** value. The number of points NPG in each complex corresponds to 2·Nopt+1 and the number of points NPS in each sub-complex to Nopt+1 (It has to be noted that each point corresponds to a set of parameters). The number of evolution steps allowed for each complex before complex shuffling, NSPL, is equal to NPG. The number of complexes is defined as **NGS**, which is assumed equal to MINGS according to the Duan investigation (Duan et al., 2004). Then, the total number of points NPT in the entire sample population is NGS·NPG.

Three different convergence criteria are defined by the user:

- The maximum number of function evaluations (or iterations) **MAXN**.
- The number of shuffling loops (**KSTOP**) in which the criterion value must change by a fixed percentage (**PCENTO**) before optimization is finished.
- The **PEPS** parameter which provides a flag indicating whether parameter convergence is reached (It compares the value of PEPS with the normalized geometric mean of parameter ranges).

Table 23 Parameters of the SCE-UA algorithm

Object	Name	Units	Description	Default Value
SCE-UA	MAXM	-	Maximum number of iterations	10000
	NGS	-	Number of complexes	3
	KSTOP	-	Number of shuffling loops	10
	PCENTO	-	Criterion value on shuffling loops	0.1
	PEPS	-	Convergence parameter	0.001
	SEED	-	Seed value	Random

4.4. Uniform Adaptive Monte Carlo

Model architecture

The Uniform Adaptive Monte Carlo (UAMC) algorithm is based on the Monte Carlo experiments that rely on repeated random sampling to obtain simulation results (Gilks et al., 1996; Liu, 2001) ; but has been modified in order to iteratively adjust the solution space.

The algorithm randomly launches a collection of simulations (block) and finds the better results in the solution space. Afterwards, the solution space is adjusted and a new group of simulations starts. The process is repeated until the optimization converges to the best set of parameters (Figure 17).

Algorithm parameters

Different parameters of the UAMC algorithm have to be defined by the user (Table 24), as presented hereafter.

A number of iterations **ITGR** per group is defined for the optimization. Random values of parameters are used for each iteration of the group based on a **SEED** value. Once the first group of iterations is finished, a number **NUMBEST** of best values is applied for calculating the solutions space range for the next group of iterations. This solution space takes into account the minimum and the maximum values of each parameter providing the best values and adds an additional range **COEFRANG**.

Finally, the optimization finishes when the convergence criterion (defined as **ERR**) is achieved, or when the maximum number of iterations **MAXN** is attained.

Table 24 Parameters of the UAMC algorithm

Object	Name	Units	Description	Default Value
UAMC	MAXN	-	Maximum number of iterations	2000
	ITGR	-	Number of iterations per group	100
	NUMBEST	-	Number of best values taken into account for the next group calculation	5
	COEFRANG	-	Additional range coefficient	0.1
	ERR	-	Error difference until convergence	0.001
	SEED	-	Seed value	Random

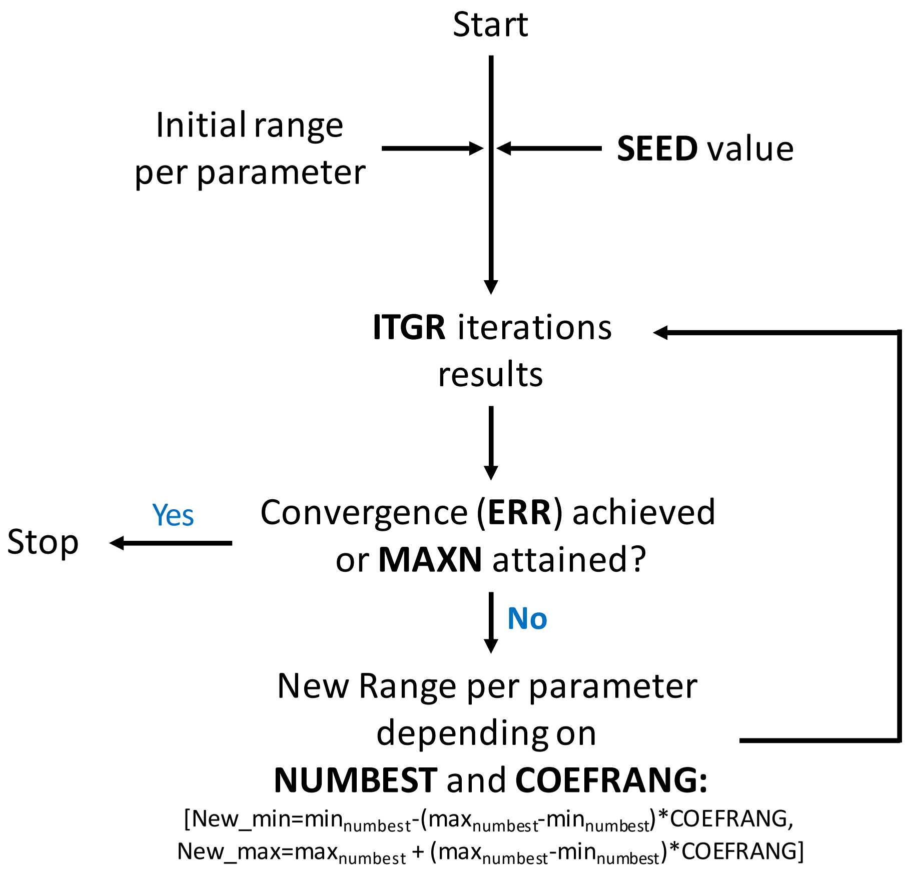

Start
Initial range per parameter
SEED value
ITGR iterations results
Convergence (ERR) achieved or MAXN attained?
Yes
Stop
No
New Range per parameter depending on NUMBEST and COEFRANG:
[New_min=min_numbest-(max_numbest-min_numbest)*COEFRANG,
New_max=max_numbest + (max_numbest-min_numbest)*COEFRANG]

Figure 17 Flow chart of the UAMC algorithm

4.5. Coupled Latin Hypercube and Rosenbrock

Model architecture

The Coupled Latin Hypercube (McKay et al., 1979) and the Rosenbrock algorithm (Rosenbrock, 1960), called hereafter CLHR, generates a powerful tool for optimization of complex problems. This combined algorithm can discretize a wide domain and then narrow your search to smaller sectors (Figure 18).

Scanning of the space of possible solutions is performed by the Latin Hypercube. This algorithm allows pseudo-statistical sampling conditioned by the previous calculated solutions. The Latin hypercube is an evolution of Monte Carlo method, with more homogeneous samples achieved with fewer samples. An important advantage of this method is that the dimension of the problem is defined by the division of the latin hypercube and not by the number of parameters.

The best results from samples become the starting points required for Rosenbrock algorithm. The advantage of this subroutine calculation lies in the speed to obtain near optimal values. This algorithm is based on a gradient search, adjusting axis changes based on the direction of maximum enhancement, thus reducing the number of evaluations of the objective function.

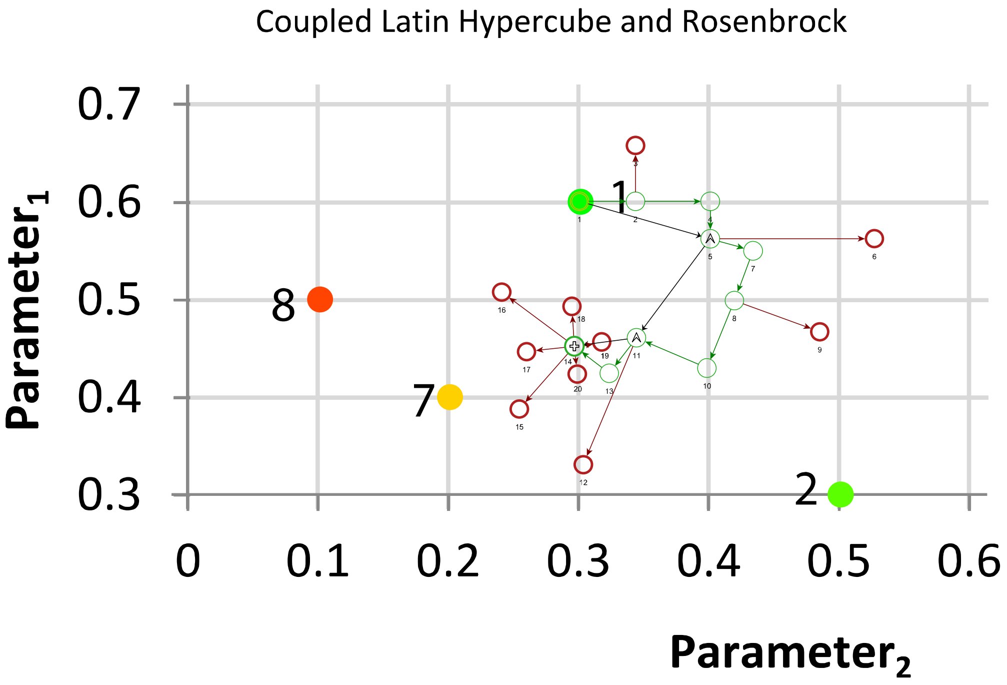


Figure 18 Illustration of the operation of the CLHR algorithm.

A general description of the steps of the CLHR method is given below:

Step 1

Generate sample: Generation and evaluation of a pseudorandom sample by a Latin hypercube within the feasible parameter space. If the size of the hypercube is greater, uncertainty is reduced within the domain of search.

Step 2

Rank points: The results obtained in the first step are ordered (Figure 29). The best results from the Latin hypercube algorithm will serve as starting points to launch the Rosenbrock algorithm.

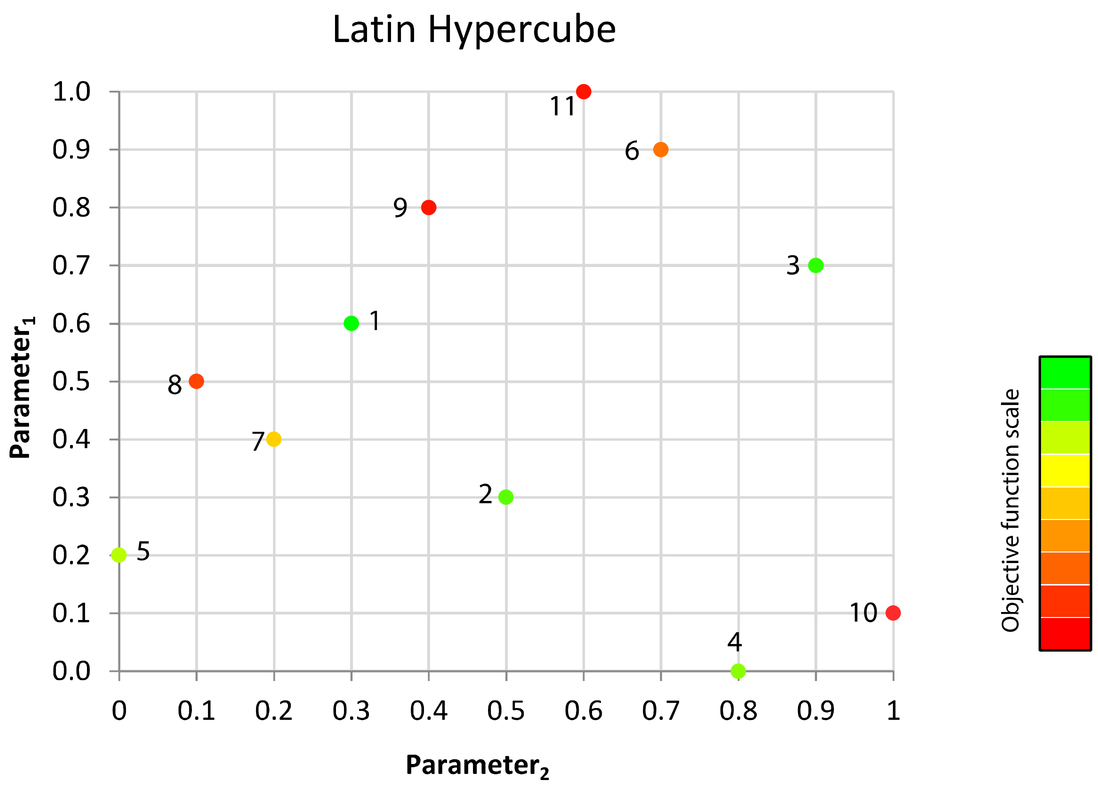


Figure 19 Illustration of the operation of the Latin Hypercube algorithm.

Step 3

Launch of Rosenbrock: Rosenbrock algorithm starts at least once. This subroutine searches around the starting point the values that improve the objective function. Depending on whether the objective function improves or worsens, the parameters values are changed to advance or backward.

Step 4

Axes change: Axes are changed to orient the Cartesian axes to the direction of maximum improvement. For applying this change of axes, it should have obtained worse results in all directions of search and at least an improvement in one of these directions (remember each direction has 2 ways).

Step 5

The best result of all Rosenbrock releases is stored.

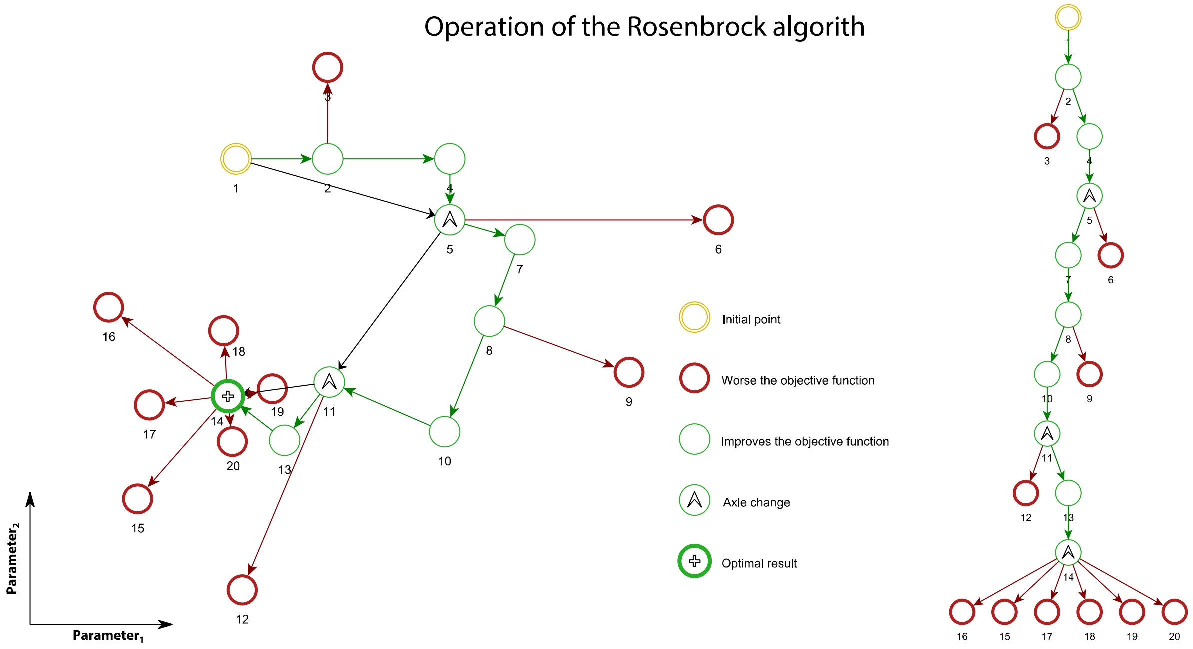

Operation of the Rosenbrock algorith
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Initial point
Worse the objective function
Improves the objective function
Axle change
Optimal result
Parameter₂
Parameter₁

Figure 20 Illustration of the operation of the Rosenbrock algorithm.

Algorithm parameters

Number of tests with the latin hypercube algorithm is equal to the parameter **DivLH** (≥ 2).The **SEED** is responsible for generating the randomness of the sample.

The **RLAUNCHES** (1≥RLAUNCHES≤DivLH) best results from the Latin Hypercube algorithm are used as starting points for the Rosenbrock algorithm.

The **ALPHA** coefficient represents the increment in the direction of search if the objective function improves. The **BETA** coefficient represents the movement if a worse result is obtained.

The **STEPROS** parameter indicates the subdivisions for each parameter's range. It is used to calculate value variations in each of the parameters (Δᵢ) to be studied, as presented in eq. CLHR.1.

$$\frac{MaxParameterValue_i - MinParameterValue_i}{STEPROS} = \Delta_i \qquad \text{CLHR.2}$$

The optimization finishes when the convergence criterion (defined as **ERR**) is achieved, or when the maximum number of iterations **MAXN** is attained.

Table 25 Parameters of the CLHR algorithm

Object	Name	Units	Description	Default Value
CLHR	MAXN	-	Maximum number of iterations	2000
	DivLH	-	Latin hypercube division	50
	SEED	-	Seed value	Random
	RLAUNCHES	-	Rosenbrock algorithm launches	2
	ALPHA	-	Advance coefficient	3
	BETA	-	Backward coefficient	-0.5
	STEPROS	-	Parameter range subdivision	40
	ERR	-	Convergence parameter	0.001

Chapter 5. Expert module - Stochastic equations

5.1. Introduction to the stochastic simulation

The problem of identifiability is basic to all statistical methods and data analysis and it occurs in diverse areas such as reliability theory, survival analysis, econometrics, etc., where stochastic modelling is widely used (Prakasa, 1992).

The module Stochastic of the Expert frame has been implemented for launching multiple simulations depending of the object's selected parameters. This module uses probability distributions to calculate the simulation parameters. The defaults results are two CSV files, one of them is a statistics series of all the simulations and the other is a parameter series with a selected distribution. The user can save all the simulation in a single file when the process is finished.

5.2. Probability distributions

A probability distribution assigns a probability to each measurable subset of the possible outcomes of a random experiment. The predefined distributions are uniform, normal, log normal, exponential and finally it is possible use a customized distribution (Figure 21).

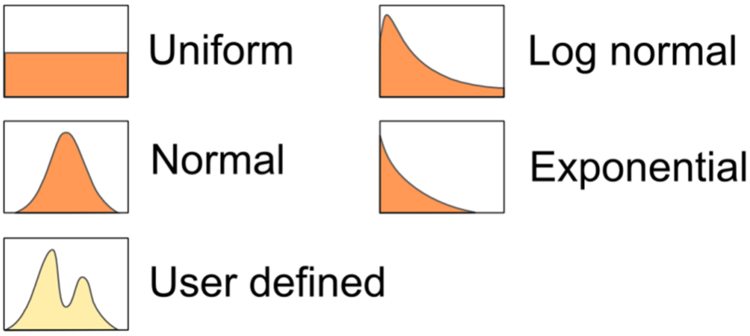


Figure 21 Probability distributions of the stochastic simulations module.

For define a correct probability distribution, the area under the probability density function must be equal to 1. For other hand, cumulative distribution function is bounded between 0 and 1 and also it must be non-decreasing and right-continuous.

Uniform

The continuous uniform distribution or rectangular distribution is a family of symmetric probability distributions such that for each member of the family, all intervals of the same length on the distribution's support are equally probable. The Figure 22 shows an example with a bounded X between [-5, 5].

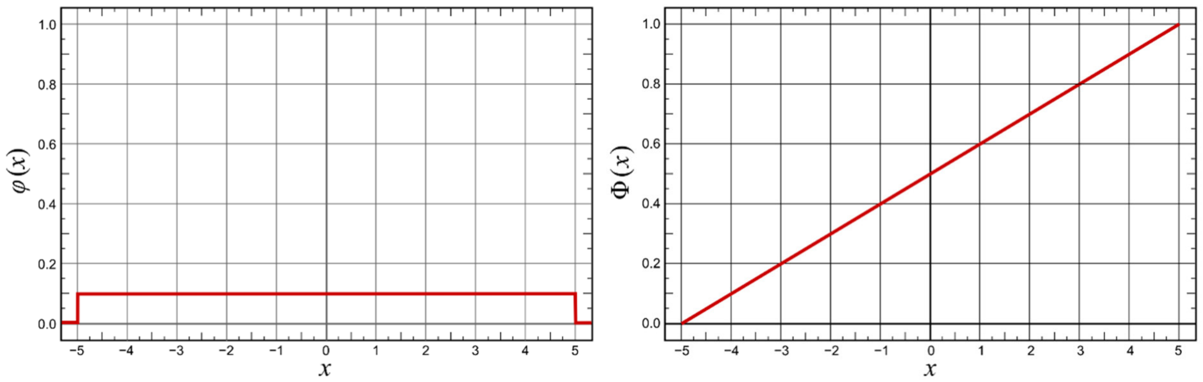


Figure 22 Uniform distribution. Probability density function (left), cumulative distribution function (right).

Normal

The normal distribution (or Gaussian distribution) is a very common continuous probability distribution. Normal distributions are important in statistics and are often used to represent distributions are not known. Several characterizations of the univariate and the multivariate normal distribution are known (Kagan et al., 1967; Prakasa, 1992) The normal distribution is sometimes informally called the bell curve. The following Figure 23 shows an example with a bounded X axis between [-5, 5], where is more probable to find a X value.

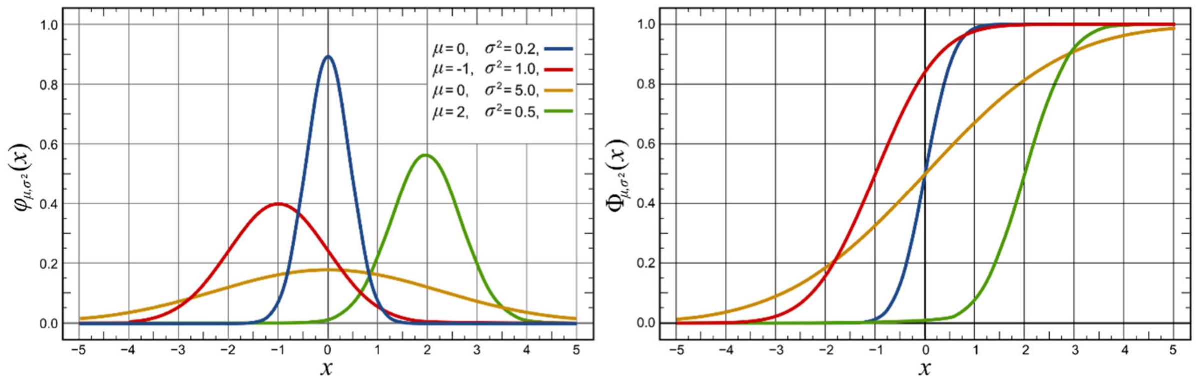


Figure 23 Normal distribution. Probability density function (left), cumulative distribution function (right).

Probability density function:

$$\varphi = f(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad \text{PD.1}$$

with µ: is mean or expectation of the distribution (and also its median and mode); σ^2 : is variance; σ is standard deviation.

Cumulative distribution function:

$$\phi = f(x \mid \mu, \sigma) = \frac{1}{2}\left[1 + \frac{1}{\sqrt{\pi}} \int_{-\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)}^{\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)} e^{-t^2}\, dt\right] \qquad \text{PD.2}$$

The normal distribution is useful because of the central limit theorem. In its most general form, under some conditions, it states that averages of random variables independently drawn from independent distributions converge in distribution to the normal, that is, become normally distributed when the number of random variables is sufficiently large.

Log normal

A log-normal distribution is a continuous probability distribution of a random variable whose logarithm is normally distributed. Consequently, if the random variable X is log-normally distributed, then Y = ln(X) has a normal distribution. Likewise, if Y has a normal distribution, then X = e^y has a log-normal distribution. A random variable which is log-normally distributed takes only positive real values (Johnson et al., 1994). The Figure 24 shows an example with a bounded X axis between [0, 2], where is more probable to find a X value.

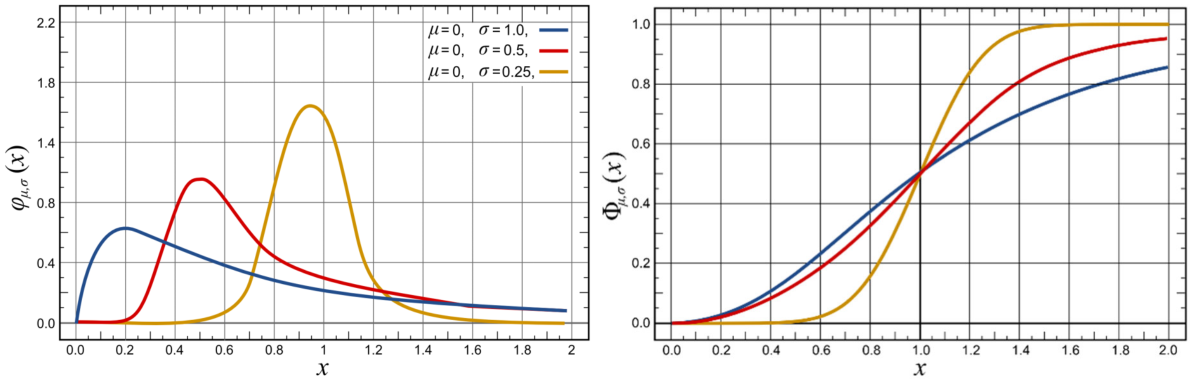


Figure 24 Log normal distribution. Probability density function (left), cumulative distribution function (right).

Probability density function:

$$\varphi = f(x \mid \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln x-\mu)^2}{2\sigma^2}} \tag{PD.3}$$

with µ: is mean or expectation of the distribution (and also its median and mode); σ^2 : is variance; σ is standard deviation.

Cumulative distribution function:

$$\phi = f(x \mid \mu, \sigma) = \frac{1}{2}\left[1 + \frac{1}{\sqrt{\pi}} \int_{-\left(\frac{\ln x-\mu}{\sigma\sqrt{2}}\right)}^{\left(\frac{\ln x-\mu}{\sigma\sqrt{2}}\right)} e^{-t^2}\, dt\right] \tag{PD.4}$$

A log-normal process is the statistical realization of the multiplicative product of many independent random variables, each of which is positive. This is justified by considering the central limit theorem in the log domain. The log-normal distribution is the maximum entropy probability distribution for a random variate X for which the mean and variance of ln(X) are specified (Park and Vera, 2009)

Exponential

The exponential distribution describes the time between events in a Poisson point process. Is a type of random mathematical object that consists of points randomly located on a mathematical space (Stoyan et al., 1995). It is a particular case of the gamma distribution. It is the continuous analogue of the geometric distribution, and it has the key property of being memoryless. The Figure 25 shows an example with a bounded X axis between [0, 10], where is more probable to find a X value.

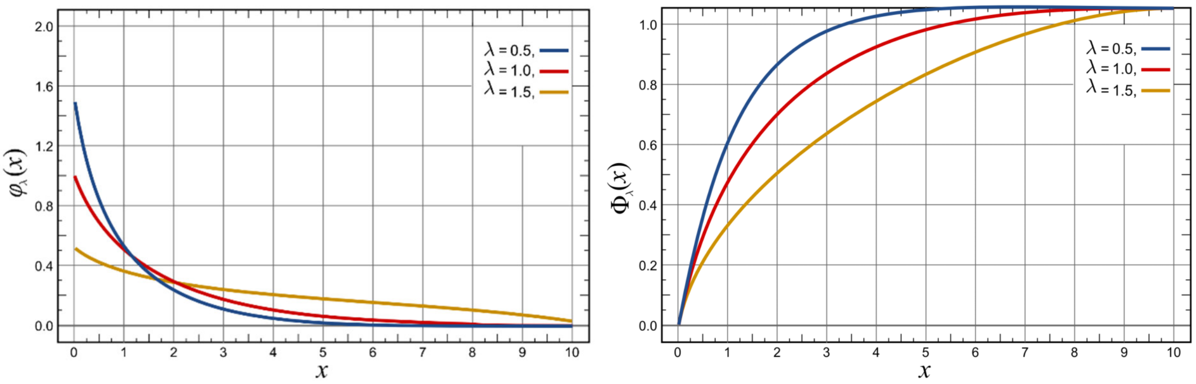


Figure 25 Exponential distribution. Probability density function (left), cumulative distribution function (right).

Probability density function:

$$\varphi = f(x \mid \lambda) = \lambda e^{-\lambda x} \qquad \text{PD.5}$$

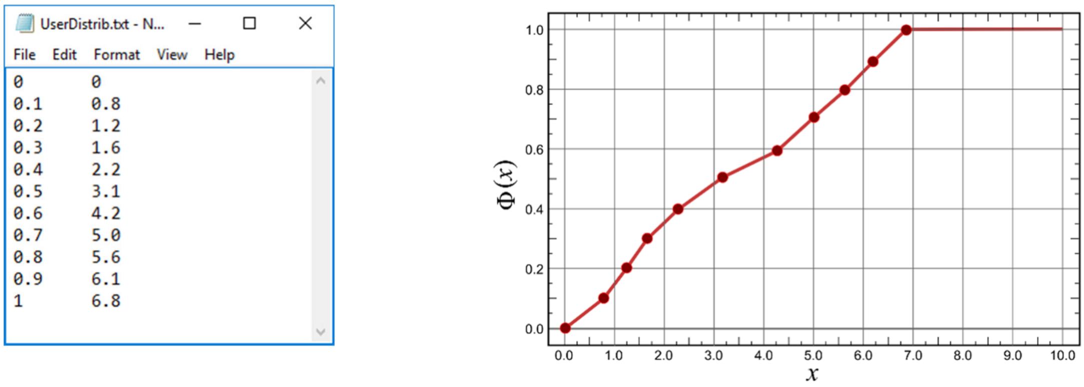
with λ: is the parameter of the distribution, often called the rate parameter [λ>0].

Cumulative distribution function:

$$\phi = f(x \mid \lambda) = 1 - e^{-\lambda x} \qquad \text{PD.6}$$

User defined

For define a custom probability distribution, the user must create a txt file with two columns (Figure 26, left). The first column is the cumulated probability and the second it is for X value associated. The middle values are calculated with a linear interpolation method. Figure 26 right is the cumulative distribution for a user defined distribution.



```
0       0
0.1     0.8
0.2     1.2
0.3     1.6
0.4     2.2
0.5     3.1
0.6     4.2
0.7     5.0
0.8     5.6
0.9     6.1
1       6.8
```

Figure 26 Custom distribution. Format distribution file (left). Cumulative distribution function (right).

The Figure 27 shows probability density function for de user defined distribution. As same to cumulative distribution, program calculate middle point through linear interpolation.

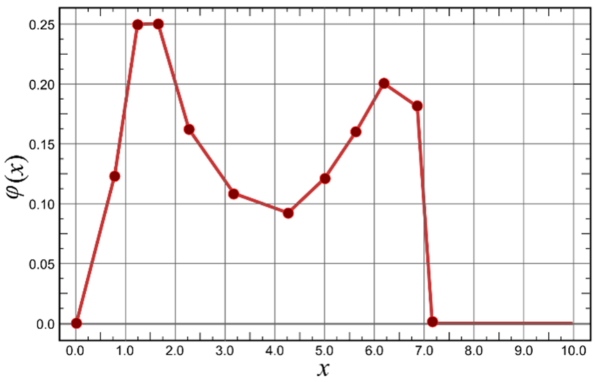


Figure 27 User distribution. Probability density function.

Chapter 6. Visual Basic Scripts

VBScript (Visual Basic Scripting Edition) is an Active Scripting language developed by Microsoft that is modelled on Visual Basic. It is designed as a "lightweight" language with a fast interpreter and can be effectively used for automating day to day office tasks as well as monitoring in the Windows-based environment. It has been installed by default in every desktop release of Microsoft Windows since Windows 98.

This chapter describes the methodology to run tasks using Visual Basic scripts when working with RS MINERVE models (running a model, extracting information from dataset, etc.).

6.1. Introduction to Visual Basic Scripts

A VBScript script must be executed within a host environment, of which there are several provided with Microsoft Windows, including: Windows Script Host (WSH), Internet Explorer (IE), and Internet Information Services (IIS).

VBScript can also be used to create applications that run directly on a workstation running Microsoft Windows. The simplest example is a script that makes use of the Windows Script Host (WSH) environment. Such a script is usually in a stand-alone file with the file extension .vbs.

VBScript can be used for managing RS MINERVE without the graphical interface. The VBScript can be invoked in two ways: from the command line with Cscript, or directly by double-clicking on the script.

In VBScript, *Properties* are attributes or values of an object that can be read and set. As long as the value returned by the property is not an object, setting and retrieving property values requires a simple assignment statement. Many properties return either individual objects or collections (a collection is an object that serves as a container for other data items or objects). These also require assignment statements that use the Set statement. For example, you can retrieve a reference to the root folder of the *C:* drive on a local system with a code fragment like the following:

```
Set oFS = CreateObject("Scripting.FileSystemObject")
```

When running RS MINERVE with VBScripts, the following code line has to be used before executing the specific RS MINERVE tasks:

```
Set rsm = CreateObject("RSMinerve.RS.Task")
```

Also, it is worth mentioning the difference between:

- Procedures (or Sub Procedures): does not return any value.
- Functions (or Function Procedures): return a value.

6.2. Available VBS procedures and functions in RS MINERVE

Table 26 Summary of the VBS procedures and functions available in RS MINERVE

Type	Name of procedure/ function	Page in Manual
Application start	Start	84
Model preparation	LoadDataset	86
	LoadDatasetAndSetDates	86
	MergeLoadedDataset	87
	LoadInitialConditionsFromFile	87
	LoadInitialConditionsFromDataset	88
	LoadInitialConditionsFromFileAndCopyFile	88
	LoadInitialConditionsFromFileAndMoveFile	89
	LoadParametersFromFile	90
	GetStartDateFromDataset	90
	GetEndDateFromDataset	91
	GetStartDateFromModel	91
	GetEndDateFromModel	92
	GetICDateFromICFile	92
	SetDates	92
	SetStartDate	93
	SetEndDate	93
	SetSimulationTimeStep	94
	SetRecordingTimeStep	94
	SetETPMethod	95
	SetLongitudeAndLatitude	95
	SetUniformETPValue	96
	SetSpatialInterpolationMethod	96
	DefineModelProperties	96
	UpdateModelProperties	97
	SetOutputRatioCoeffsInPlanner	98
	SetConditionThresholdInPlanner	98
	SetDaysInPlanner	99
	SetHoursInPlanner	100
Simulation and reports	SavePreSimulationReportAs	101
	SavePreSimulationReport	101
	Simulate	102
	SavePostSimulationReportAs	102
	SavePostSimulationReport	103
Save model and properties	SaveModelAs	104
	SaveModel	104
	SaveInitialConditionsAs	104
	SaveInitialConditions	105
	SaveFinalConditionsAs	105
	SaveFinalConditions	106

	SaveParametersAs	107
	SaveParameters	107
Model results	SaveFullResultsAs	109
	SaveFullResults	109
	SaveSelectionResultsAs	110
	SaveSelectionResults	111
	SaveInitializationDatasetAs	112
	SaveInitializationDataset	112
	ExportResultsTo	113
	MergeFullResults	114
	MergeSelectionResults	114
	MergeDatasetsAs	115
	MergeDatasets	116
	GetPropertyValue	117
	GetIndicatorValue	117
Calibration	LoadCalibrationsInModel	119
	LoadCalibrationsByFile	119
	SaveAllCalibrationsInModel	119
	RunCalibration	120
	RunAllCalibrations	120
	ExportCalibrationResults	120
	ExportAllCalibrationsResults	121
	SetCalibrationDates	121
Application stop	Stop	123
Other procedures/functions	RemoveFile	124
	GetAppVersion	124
	GetReaderVersion	124

6.3. Application Start

Start *Opening of the model*

Description

This procedure opens the indicated model to be used.

Syntax

Start (modelFullPath, debug=false)

Parameters

modelFullPath	String. Full path of the hydrological model. The file extension of the model has to be **.rsm*; otherwise, the script reports an error.
Debug	Boolean (optional). If TRUE, the script writes a text file with the error logs.

Errors

Argument Null Exception	“Model file not defined”
Argument Exception	“Model file not found”
Invalid Operation Exception	“Model file not valid”

6.4. Model preparation

LoadDataset	*Loading of the input dataset*

Description

This procedure loads the indicated input dataset in the model. The model has to have been previously open.

Syntax

LoadDataset (inputDatasetFullPath)

Parameters

inputDatasetFullPath	String. Full path of the input dataset

Errors

Invalid Operation Exception	“Model not loaded”
Argument Null Exception	“Input dataset not defined”
Argument Exception	“Input dataset not found”
Invalid Operation Exception	“Input dataset not valid”
Invalid Operation Exception	“Failed to apply datasources”

LoadDatasetAndSetDates	*Loading of the input dataset and set dates*

Description

The input dataset is loaded, and the datasource and the simulation period are defined based on the dataset. Depending on the parameters and whether each time series have a different period, the simulation period is defined.

Syntax

LoadDatasetAndSetDates (inputDatasetFullPath, canBeginWithMissingData, canEndWithMissingData)

Parameters

inputDatasetFullPath	String. Full path of the input dataset
canBeginWithMissingData	Boolean. If false, the last start date of the time series is used as simulation date start.
canEndWithMissingData	Boolean. If false, the first last date of the time series is used as simulation date end.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Null Exception	“Input dataset not defined”
Argument Exception	“Input dataset not found”

Invalid Operation Exception	“Input dataset not valid”
Invalid Operation Exception	“Failed to apply datasources”
Invalid Operation Exception	“Input dataset is empty”

MergeLoadedDataset *Merging of the input dataset to base dataset*

Description

This procedure merges a loaded dataset to a base dataset. It can be useful when new data is available and the user would like to merge to historical data. The result is an updated base dataset containing the loaded dataset, saved in a text or binary format file.

Syntax

MergeLoadedDataset (baseDatasetFullPath, saveDataInDstFile)

Parameters

baseDatasetFullPath	String. Full path of the base dataset.
saveDataInDstFile	Boolean. If true the dataset values are saved in text format (**.dst*), if false in binary format (**.dsb*).

Errors

Invalid Operation Exception	“Input dataset not loaded, failed to merge with base dataset“
Argument Null Exception	“Input dataset not defined”
Argument Exception	“Fullpath of base dataset not valid”
Invalid Operation Exception	“Failed to create new base dataset”
Invalid Operation Exception	“Base dataset not valid”
Invalid Operation Exception	“Failed to merge input dataset with base dataset”
Invalid Operation Exception	“Failed to save changesin base dataset”

LoadInitialConditionsFromFile *Setting of the initial conditions*

Description

This procedure sets the initial conditions of the model from a text or an Excel file (please refer to the section **7.2** of this manual to learn more about Initial Conditions files).

Syntax

LoadInitialConditionsFromFile (icFileFullPath)

Parameters

icFileFullPath	String. Full path of the IC file. Acceptable file extensions are **.txt* and **.xlsx*.

Errors

Invalid Operation Exception	"Model not loaded"
Argument Null Exception	"IC file not defined"
Argument Exception	"IC file not found"
Invalid Operation Exception	"Failed to import initial conditions from IC file"

LoadInitialConditionsFromDataset *Setting of the initial conditions from a dataset*

Description

The initial conditions are set from a dataset file (please refer to the section **8.2** of this manual to learn more about dataset formats).

Syntax

LoadInitialConditionsFromDataset (icDatasetFullPath)

Parameters

icDatasetFullPath	String. Full path of the dataset from where to extract the initial conditions.

Errors

Invalid Operation Exception	"Model not loaded"
Argument Null Exception	"IC dataset not defined"
Argument Exception	"IC dataset not found"
Invalid Operation Exception	"IC dataset not valid"
Invalid Operation Exception	"Failed to load results from IC dataset"
Invalid Operation Exception	"Failed to initialize the model with IC dataset"

LoadInitialConditionsFromFileAndCopyFile *Setting of the initial conditions from a file and copy the file*

Description

This procedure sets the initial conditions of the model from a text or an Excel file, and copies them in a newly created text file (which name is based on the original IC filename but includes the start date of the simulation). If the indicated folder where to create the new IC file does not exist it will be created.

Syntax

LoadInitialConditionsFromFileAndCopyFile (icFileFullPath, saveFolder)

Parameters

icFileFullPath	String. Full path of the IC file. Acceptable file extensions are *.*txt* and *.*xlsx*.

saveFolder	String. Folder where results have to be stored.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Null Exception	“IC file not defined”
Argument Exception	“IC file not found”
Argument Null Exception	“Save folder not defined”
Argument Exception	“Save folder '*saveFolder*' not valid”
Invalid Operation Exception	“Failed to import initial conditions from IC file”
Invalid Operation Exception	“Failed to copy the IC file”

LoadInitialConditionsFromFileAndMoveFile	*Loading of the initial conditions and moving of the text file*

Description

This procedure sets the initial conditions of the model from a text or an Excel file, and moves the file to the specified folder (the filename is based on the original IC filename but includes the start date of the simulation). The target file must not exist before; otherwise an error occurs.

Syntax

LoadInitialConditionsFromFileAndMoveFile (icFileFullPath, saveFolder)

Parameters

icFileFullPath	String. Full path of the IC file. Acceptable file extensions are *.*txt* and *.*xlsx*.
saveFolder	String. Folder where results are stored.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Null Exception	“IC file not defined”
Argument Exception	“IC file not found”
Argument Null Exception	“Save folder not defined”
Argument Exception	“Save folder '*saveFolder*' not valid”
Invalid Operation Exception	“Failed to import initial conditions from IC file”
Invalid Operation Exception	“Failed to move the IC file”

LoadParametersFromFile *Setting of the parameters*

Description

This procedure sets the parameters of the model from a text or an Excel file (please refer to the section **7.2** of this manual to learn more about Parameters files).

Syntax

LoadParametersFromFile (paramFileFullPath)

Parameters

icFileFullPath	String. Full path of the Parameters file. Acceptable file extensions are **.txt* and **.xlsx*.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Null Exception	“Parameters file not defined”
Argument Exception	“Parameters file not found”
Invalid Operation Exception	“Failed to import parameters from Parameters file”

GetStartDateFromDataset *Getting the start date from a dataset*

Description

This function extracts the start date of a specified dataset file. If the dataset contains different start dates (data series with initial lacking data), the user can choose whether to get the first common date of the complete dataset.

Syntax

GetStartDateFromDataset (DatasetFullPath, canBeginWithMissingData)

Parameters

DatasetFullPath	String. Full path of the dataset file.
canBeginWithMissingData	Boolean. If true the last start date of the time series is used as simulation date start.

Returned value

Start date	It returns the value of the start date of a dataset. The format of the date is: dd.MM.yyyy HH:mm:ss.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Null Exception	“Dataset not defined”
Argument Exception	“Dataset not found”

Invalid Operation Exception	"Dataset not valid"

GetEndDateFromDataset	*Getting the end date from a dataset*

Description

This function extracts the end date of a specified dataset file. If the dataset contains different end dates (data series with final lacking data), the user can choose whether to get the last common date of the complete dataset.

Syntax

GetEndDateFromDataset (DatasetFullPath, canEndWithMissingData)

Parameters

DatasetFullPath	String. Full path of the dataset file.
canEndWithMissingData	Boolean. If true the first last date of the time series is used as simulation date end.

Outputs

End date	It returns the value of the end date of a dataset. The format of the date is: dd.MM.yyyy HH:mm:ss.

Errors

Invalid Operation Exception	"Model not loaded"
Argument Null Exception	"Dataset not defined"
Argument Exception	"Dataset not found"
Invalid Operation Exception	"Dataset not valid"

GetStartDateFromModel	*Getting the start date from model*

Description

This function extracts the start date from the loaded model.

Syntax

GetStartDateFromModel ()

Returned value

Start date	It returns the value of the start date of the model. The format of the date is: dd.MM.yyyy HH:mm:ss.

Errors

Invalid Operation Exception	"Model not loaded"

GetEndDateFromModel *Getting the end date from model*

Description

This function extracts the end date from the loaded model.

Syntax

GetEndDateFromModel ()

Returned value

End date	It returns the value of the end date of the model. The format of the date is: dd.MM.yyyy HH:mm:ss.

Errors

Invalid Operation Exception	“Model not loaded”

GetICDateFromICFile *Getting the date from an Initial Conditions file*

Description

This function extracts the date corresponding to a specified Initial Conditions file.

Syntax

GetICDateFromICFile (icFileFullPath)

Parameters

icFileFullPath	String. Full path of the IC file. Acceptable file extension is **.txt*.

Returned value

Date	It returns the date of the Initial Conditions file. The format of the date is: dd.MM.yyyy HH:mm:ss.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Null Exception	“IC file not defined”
Argument Exception	“IC file not found”
Invalid Operation Exception	“Failed to get IC Date from IC file”

SetDates *Setting of the dates*

Description

This procedure sets the simulation period based on specified start and end dates.

Syntax

SetDates (startDate, endDate)

Parameters

startdate	Date. Start date for the simulation period. The format of the date has to be: dd.MM.yyyy HH:mm:ss.
enddate	Date. End date for the simulation period. The format of the date has to be: dd.MM.yyyy HH:mm:ss.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Exception	“Start Date not valid, unable to convert '*startdate*' to a Date”
Argument Exception	“End Date not valid, unable to convert '*enddate*' to a Date”

SetStartDate	*Setting of the start date*

Description

This procedure sets the simulation period start date.

Syntax

SetStartDate (startDate)

Parameters

startdate	Date. Start date for the simulation period. The format of the date has to be: dd.MM.yyyy HH:mm:ss.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Exception	“Start Date not valid, unable to convert '*startdate*' to a Date”

SetEndDate	*Setting of the end date*

Description

This procedure sets the simulation period end date.

Syntax

SetEndtDate (endDate)

Parameters

enddate	Date. End date for the simulation period. The format of the date has to be: dd.MM.yyyy HH:mm:ss.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Exception	“End Date not valid, unable to convert '*enddate*' to a Date”

SetSimulationTimeStep — *Setting of the simulation timestep*

Description

This procedure sets the simulation timestep and its unit for the simulation.

Syntax

SetSimulationTimeStep (value, unit)

Parameters

value	Double. Simulation time step as double (value has to be >0).
unit	String. Unit: Seconds, Minutes, Hours or Days.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Exception	“Simulation time step not valid, unable to convert '*value*' to a Double”
Argument Exception	“Simulation time step not valid, '*value*' is outside the range of a Double”
Argument Exception	“Simulation time step not valid, unit '*unit*' not recognized”

SetRecordingTimeStep — *Setting of the recording timestep*

Description

This procedure sets the recording timestep and its unit for the simulation.

Syntax

SetRecordingTimeStep (value, unit)

Parameters

value	Double. Recording time step as double (value has to be >0).
unit	String. Unit: Seconds, Minutes, Hours, Days or Months.

Errors

Invalid Operation Exception	“Model not loaded”

Argument Exception	“Recording time step not valid, unable to convert '*value*' to a Double”
Argument Exception	“Recording time step not valid, '*value*' is outside the range of a Double”
Argument Exception	“Recording time step not valid, unit '*unit*' not recognized”

SetETPMethod	*Setting of the ETP method*

Description

This procedure sets the ETP (Potential EvapoTranspiration) method in the model (please refer to the section **7.2** of the User’s Manual to learn more about the model Settings).

Syntax

SetETPMethod (etpMethod)

Parameters

selectedETPMethod	String. Selected ETP Method: Turc, McGuinness, Oudin or Uniform ETP.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Exception	“Selected ETP Method is not implemented in RS MINERVE”

SetLongitudeAndLatitude	*Setting of the Longitude and Latitude*

Description

This procedure sets the Longitude and Latitude used when applying ETP methods.

Syntax

SetLongitudeAndLatitude (longitude, latitude)

Parameters

longitude	Integer. Selected Longitude.
latitude	Integer. Selected Latitude.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Exception	“Longitude not valid”
Argument Exception	“Latitude not valid”

SetUniformETPValue	*Setting of the Uniform ETP*

Description

This procedure sets the Uniform ETP value when Uniform ETP Method has been set.

Syntax

SetUniformETPValue (value)

Parameters

value	String. Uniform ETP value.

Errors

Invalid Operation Exception	"Model not loaded"
Argument Exception	"Uniform ETP value not valid"

SetSpatialInterpolationMethod	*Setting of the Spatial Interpolation method*

Description

This procedure sets the Spatial Interpolation method to be used.

Syntax

SetSpatialInterpolationMethod (spatialInterpolationMethod)

Parameters

selectedETPMethod	String. Selected ETP Method.

Errors

Invalid Operation Exception	"Model not loaded"
Argument Exception	"Selected Spatial Interpolation Method does not exist in RS MINERVE"

DefineModelProperties	*Defining property of an object*

Description

This procedure defines a parameter or an initial condition in all objects included in the specified zone.

Syntax

DefineModelProperties (objectType, propertyName, value, zone = null)

Parameters

objectType	String. Object type (e.g. SOCONT, GSM, Reservoir, etc.). See Table 27.
propertyName	String. Property name (e.g. HGR3Max). See Table 27.

value	Double. Value of the property.
zone = null	String (optional). Zone where the objects are.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Exception	“Object type 'objectType' not valid”
Argument Null Exception	“Property name not defined”
Argument Exception	“Zone 'zone' not found in ‘hydrologicalmodel’”
Argument Exception	“Value is not valid, unable to convert ‘value’ to a Double”
Invalid Operation Exception	“Failed to define model properties”

UpdateModelProperties *Updating of model properties*

Description

This procedure updates a parameter or an initial condition in all objects included in the specified zone.

Syntax

UpdateModelProperties (objectType, propertyName, scale, shift, linkedPropertyName=null, zone = null, objectName=null)

Parameters

objectType	String. Object type (e.g. SOCONT, GSM, Reservoir, etc.). See Table 27.
propertyName	String. Property name (e.g. HGR3Ini). See Table 27.
scale	String. Scale to multiply the current property.
shift	String. Shift to add to the current property.
linkedPropertyName = null	String (optional). Name of the property that restricts the maximum value of the updated property (e.g. the HGR3Max when updating the HGR3Ini).
zone = null	String (optional). Zone where the objects to which the procedure will be applied are.
objectName = null	String (optional). Name of the object to which the procedure will be applied. The object must be within the zone if this one has been specified.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Exception	“Object type 'objectType' not valid”

Argument Null Exception	“Property name not defined”
Argument Exception	“Zone 'zone' not found in ‘hydrologicalmodel’”
Argument Exception	“Scale is not valid, unable to convert ‘scale’ to a Double”
Argument Exception	“Shift is not valid, unable to convert ‘shift’ to a Double”
Invalid Operation Exception	“Failed to update model properties”

SetOutputRatioCoeffsInPlanner	*Setting of the coefficients of the expression defining a rule’s output*

Description

This procedure sets the coefficients A and B in the expression: *Q=A*Output+B* for the definition of a particular Rule output at a specified Planner.

Syntax

SetOutputRatioCoeffsInPlanner (plannerName, ruleName, coefA, coefB)

Parameters

plannerName	String. Name of the Planner object concerned.
ruleName	String. Name of the Rule (defined within the above Planner) for which the output expression’s coefficients are to be defined.
coefA	Double. Value of the coefficient A in the output expression *Q=A*Output+B*
coefB	Double. Value of the coefficient B in the output expression *Q=A*Output+B*

Errors

Argument Null Exception	“Planner name not defined”
Argument Null Exception	“Rule name not defined”
Argument Exception	“Ratio Coeff. A is not valid, unable to convert to a Double”
Argument Exception	“Ratio Coeff. B is not valid, unable to convert to a Double”
Invalid Operation Exception	“No Planner with this name”
Invalid Operation Exception	“No Rule with this name”
Invalid Operation Exception	“Failed to define Ratio Coefficient in Planner”

SetConditionThresholdInPlanner	*Setting of the threshold value of a condition in a specific Planner*

Description

This procedure sets the Threshold value of a condition in a Rule of a specific Planner.

Syntax

SetConditionThresholdInPlanner (plannerName, ruleName, conditionName, value)

Parameters

regulationName	String. Name of the Planner object concerned.
ruleName	String. Name of the Rule, defined within the above Planner.
conditionName	String. Name of the Condition, defined within the above Rule, for which the threshold is to be defined.
value	Double. Value of the threshold of the specified condition.

Errors

Argument Null Exception	"Planner name not defined"
Argument Null Exception	"Rule name not defined"
Argument Null Exception	"Condition name not defined"
Argument Exception	"Value is not valid, unable to convert to a Double"
Invalid Operation Exception	"No Planner with this name"
Invalid Operation Exception	"No Rule with this name"
Invalid Operation Exception	"No condition with this name"
Invalid Operation Exception	"Failed to define Threshold value in Planner"

SetDaysInPlanner	*Setting of the days of a rule in a specific Planner*

Description

This procedure sets the week days of a Rule of a specific Planner.

Syntax

SetDaysInPlanner (plannerName, ruleName, day1, day2, day3, day4, day5, day6, day7)

Parameters

plannerName	String. Name of the Planner object concerned.
ruleName	String. Name of the Rule, defined within the above Planner.
day1,…, day7	Boolean. Whether the rule is active for the nth day of the week (day1=Monday, day2=Tuesday…).

Errors

Argument Null Exception	"Planner name not defined"
Argument Null Exception	"Rule name not defined"

Argument Exception	“No Planner with name plannerName”
Argument Exception	“No Rule with name ruleName”
Invalid Operation Exception	“Failed to define rule days in Planner”

SetHoursInPlanner	*Setting of the hours of a rule in a specific Planner*

Description

This procedure sets the hours of a Rule of a specific Planner.

Syntax

SetHoursInPlanner (plannerName, ruleName, startHour, endHour)

Parameters

plannerName	String. Name of the Planner object concerned.
ruleName	String. Name of the Rule, defined within the above Planner.
startHour	String. Initial hour of the period for which the rule is active.
endHour	String. Final hour of the period for which the rule is active.

Errors

Argument Null Exception	“Planner name not defined”
Argument Null Exception	“Rule name not defined”
Argument Exception	“No Planner with name plannerName”
Argument Exception	“No Rule with name ruleName”
Invalid Operation Exception	“Failed to define hours in Planner”

6.5. Simulation and reports

SavePreSimulationReportAs	*Writing of the pre-simulation report*

Description

This procedure writes the pre-simulation report in a text file, at the specified location with the name: *preSimulationReportFullPath.txt*.

Syntax

SavePreSimulationReportAs (preSimulationReportFullPath)

Parameters

preSimulationReportFullPath	String. Full path of the text file where to save the pre-simulation report. The file extension has to be **.txt*.

Errors

Invalid Operation Exception	"Model not loaded"
Argument Null Exception	"Fullpath not defined"
Argument Exception	"Fullpath contains invalid characters"
Argument Exception	"Filename extension not valid"
Argument Exception	"Fullpath not rooted"
Invalid Operation Exception	"Failed to save pre-simulation report"

SavePreSimulationReport	*Writing of the pre-simulation report*

Description

This procedure writes the pre-simulation report in a text file, at the specified location with the specified reference name.

Syntax

SavePreSimulationReport (saveFolder, referenceName, showInitialDateInFileName = true)

Parameters

saveFolder	String. Folder path where to save the text file.
referenceName	String. Reference name for the file.
showInitialDateInFileName	Boolean (optional). Adds the StartDate of the simulation to the filename if True (e.g. *yyyyMMddHH_referenceName.txt*).

Errors

Invalid Operation Exception	"Model not loaded"
Argument Null Exception	"Reference name not defined"

Argument Null Exception	“Save folder not defined”
Argument Exception	“Save folder 'savefolder' not valid”
Invalid Operation Exception	“Failed to save pre-simulation report”

Simulate	*Running of the model*

Description

This procedure executes the simulation of the model with the loaded parameters and hydrometeorological datasets.

Syntax

Simulate()

Errors

Invalid Operation Exception	“Model not loaded”
Invalid Operation Exception	“Simulation failed, see /log_RSM/errlog.txt”

SavePostSimulationReportAs	*Writing of the post-simulation report*

Description

This procedure writes the post-simulation report in a text file, at the specified location with the name: *postSimulationReportFullPath*.

Syntax

SavePostSimulationReportAs (postSimulationReportFullPath)

Parameters

postSimulationReportFullPath	String. Full path of the text file where to save the post-simulation report. The file extension has to be **.txt*.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Null Exception	“Fullpath not defined”
Argument Exception	“Fullpath contains invalid characters”
Argument Exception	“Filename extension not valid”
Argument Exception	“Fullpath not rooted”
Invalid Operation Exception	“Failed to save post-simulation report”

SavePostSimulationReport *Writing of the post-simulation report*

Description

This procedure writes the post-simulation report in a text file, at the specified location with the specified reference name.

Syntax

SavePostSimulationReport (saveFolder, referenceName, showInitialDateInFileName = true)

Parameters

saveFolder	String. Folder path where to save the text file.
referenceName	String. Reference name for the file.
showInitialDateInFileName	Boolean (optional). Adds the StartDate of the simulation to the filename if True (e.g. *yyyyMMddHH_referenceName.txt*).

Errors

Invalid Operation Exception	“Model not loaded”
Argument Null Exception	“Reference name not defined”
Argument Null Exception	“Save folder not defined”
Argument Exception	“Save folder 'saveFolder' not valid”
Invalid Operation Exception	“Failed to save post-simulation report”

6.6. Save model and properties

SaveModelAs	*Saving of the model*

Description

This procedure saves the loaded model, at a defined location with a specified name: *modelNameFullPath*.

Syntax

SaveModelAs (modelNameFullPath)

Parameters

modelNameFullPath	String. Full path of the model to save. The file extension has to be **.rsm*.

Errors

Invalid Operation Exception	"Model not loaded"
Argument Null Exception	"Fullpath not defined"
Argument Exception	"Fullpath contains invalid characters"
Argument Exception	"Filename extension not valid"
Argument Exception	"Fullpath not rooted"
Invalid Operation Exception	"Failed to save model"

SaveModel	*Saving of the model*

Description

This procedure saves the loaded model at its current location.

Syntax

SaveModel ()

Errors

Invalid Operation Exception	"Model not loaded"
Invalid Operation Exception	"Failed to save model"

SaveInitialConditionsAs	*Saving of the initial conditions*

Description

This procedure saves the initial conditions at starting date of the simulation (in a text or an Excel format), at the specified location with the name: *initialConditionsFullPath*.

Syntax

SaveInitialConditionsAs (initialConditionsFullPath)

Parameters

initialConditionsFullPath	String. Full path of the text or Excel file where to save the Initial Conditions. The file extension has to be **.txt*. or **.xlsx*.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Null Exception	“Fullpath not defined”
Argument Exception	“Fullpath contains invalid characters”
Argument Exception	“Fullpath not rooted”
Invalid Operation Exception	“Failed to create and save initial conditions file”

SaveInitialConditions	*Saving of the initial conditions*

Description

This procedure saves the initial conditions at starting date of the simulation as a text file, at the specified location with the specified reference name.

Syntax

SaveInitialConditions (saveFolder, referenceName, showInitialDateInFileName = true)

Parameters

saveFolder	String. Folder path where to save the file.
referenceName	String. Reference name for the file.
showInitialDateInFileName	Boolean (optional). Add the StartDate of the simulation to the filename if True (e.g. *yyyyMMddHH_referenceName.txt*).

Errors

Invalid Operation Exception	“Model not loaded”
Argument Null Exception	“Reference name not defined”
Argument Null Exception	“Save folder not defined”
Argument Exception	“Save folder 'saveFolder' not valid”
Invalid Operation Exception	“Failed to save initial conditions file”

SaveFinalConditionsAs	*Saving of the final conditions*

Description

This procedure saves the conditions at the end of the simulation (in a text or an Excel format), at the specified location with the name: *finalConditionsFullPath*.

Syntax

SaveFinalConditionsAs (finalConditionsFullPath)

Parameters

finalConditionsFileFullPath	String. Full path of the text or Excel file where to save the Final Conditions. The file extension has to be **.txt*. or **.xlsx*.

Errors

Invalid Operation Exception	"Model not loaded"
Invalid Operation Exception	"Model 'hydrologicalmodel' has no results"
Argument Null Exception	"Fullpath not defined"
Argument Exception	"Fullpath contains invalid characters"
Argument Exception	"Fullpath not rooted"
Invalid Operation Exception	"Failed to generate final conditions"
Invalid Operation Exception	"Failed to create and save final conditions file"

SaveFinalConditions	*Saving of the final conditions*

Description

This procedure saves the initial conditions at the end of the simulation as a text file, at the specified location with the specified reference name.

Syntax

SaveFinalConditions (saveFolder, referenceName, showInitialDateInFileName = true)

Parameters

saveFolder	String. Folder path where to save the file.
referenceName	String. Reference Name for the file.
showInitialDateInFileName	Boolean (optional). Add the EndDate of the simulation to the filename if True (e.g. *yyyyMMddHH_referenceName.txt*).

Errors

Invalid Operation Exception	"Model not loaded"
Invalid Operation Exception	"Model 'hydrologicalmodel' has no results"
Argument Null Exception	"Reference name not defined"
Argument Null Exception	"Save folder not defined"

Argument Exception	“Save folder 'saveFolder' not valid”
Invalid Operation Exception	“Failed to generate final conditions”
Invalid Operation Exception	“Failed to save final conditions file”

SaveParametersAs *Saving of the parameters*

Description

This procedure saves the parameters of the model (in a text or an Excel format), at the specified location with the name: *parametersFullPath*.

Syntax

SaveParametersAs (parametersFullPath)

Parameters

parametersFullPath	Full path of the text or Excel file where to save the parameters. The file extension has to be **.txt*. or **.xlsx*.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Null Exception	“Fullpath not defined”
Argument Exception	“Fullpath contains invalid characters”
Argument Exception	“Fullpath not rooted”
Invalid Operation Exception	“Failed to create and save parameters file”

SaveParameters *Saving of the parameters*

Description

This procedure saves the parameters of the model as a text file, at the specified location with the specified reference name.

Syntax

SaveParameters (saveFolder, referenceName)

Parameters

saveFolder	String. Folder path where to save the file.
referenceName	String. Reference Name for the file.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Null Exception	“Reference name not defined”

Argument Null Exception	“Save folder not defined”
Argument Exception	“Save folder 'saveFolder' not valid”
Invalid Operation Exception	“Failed to save parameters file”

6.7. Model results

SaveFullResultsAs	*Saving of all the results*

Description

This procedure saves the full results dataset of the simulation in a binary or a text format, at the specified location with the name: *resultsDatasetFullPath*.

Syntax

SaveFullResultsAs (resultsDatasetFullPath, saveDataInDstFile)

Parameters

resultsDatasetFullPath	String. Full path of the file where to save the full results dataset. The file extension has to be **.dsx*.
saveDataInDstFile	Boolean. If true the dataset values are saved in text format (**.dst*), if false in binary format (**.dsb*).

Errors

Invalid Operation Exception	"Model not loaded"
Invalid Operation Exception	"Model 'hydrologicalmodel' has no results"
Argument Null Exception	"Fullpath of full results dataset not defined"
Argument Exception	"Fullpath of full results dataset contains invalid characters"
Argument Exception	"Filename extension of full results dataset not valid"
Argument Exception	"Fullpath of full results dataset not rooted"
Invalid Operation Exception	"Failed to create and save full results dataset"

SaveFullResults	*Saving of all the results*

Description

This procedure saves the full results dataset of the simulation in a binary or a text format, at the specified location with the specified reference name and with or without the simulation length (in hours) in the name.

Syntax

SaveFullResults (saveFolder, referenceName, saveDataInDstFile, showDurationInFileName, showInitialDateInFileName = true)

Parameters

saveFolder	String. Folder path where to save the file.
referenceName	String. Reference Name for the file.

saveDataInDstFile	Boolean. If true the dataset values are saved in text format (**.dst*), if false in binary format (**.dsb*).
showDurationInFileName	Boolean. If true it adds the duration in hours of the simulation in the filename (e.g. *referencename_nbHoursH.txt*).
showInitialDateInFileName	Boolean (optional). If true it adds the StartDate of the simulation to the filename (e.g. *yyyyMMddHH_referencename.txt*).

Errors

Invalid Operation Exception	“Model not loaded”
Invalid Operation Exception	“Model 'hydrologicalmodel' has no results”
Argument Null Exception	“Reference name not defined”
Argument Null Exception	“Save folder not defined”
Argument Exception	“Save folder 'saveFolder' not valid”
Invalid Operation Exception	“Failed to save full result”

SaveSelectionResultsAs	*Saving of selection results*

Description

This procedure saves the results dataset of the selection in a binary or a text format, at the specified location with the name: *selectionResultsDatasetFullPath*.

Syntax

SaveSelectionResultsAs (selectionResultsDatasetFullPath, saveDataInDstFile, selectionName)

Parameters

selectionResultsDatasetFullPath	String. Full path of the file where to save the selection results dataset. The file extension has to be **.dsx*.
saveDataInDstFile	Boolean. If true the dataset values are saved in text format (**.dst*), if false in binary format (**.dsb*).
selectionName	String. Name of the selection of results.

Errors

Invalid Operation Exception	“Model not loaded”
Invalid Operation Exception	“Model 'hydrologicalmodel' has no results”
Argument Null Exception	“Fullpath of selection results dataset not defined”
Argument Exception	“Fullpath of selection results dataset contains invalid characters”
Argument Exception	“Filename extension of selection results dataset not valid”

Argument Exception	"Fullpath of selection results dataset not rooted"
Argument Exception	"Selection 'selectionname' not valid"
Invalid Operation Exception	"Failed to create and save selection results dataset

SaveSelectionResults	*Saving of selection results*

Description

This procedure saves the results dataset of the selection in a binary or a text format, at the specified location with the specified reference name and with or without the simulation length (in hours) in the name.

Syntax

SaveSelectionResults (saveFolder, referenceName, saveDataInDstFile, selectionName, showDurationInFileName, showInitialDateInFileName = true)

Parameters

saveFolder	String. Folder path where to save the file.
referenceName	String. Reference Name for the file.
saveDataInDstFile	Boolean. If true the dataset values are saved in text format (**.dst*), if false in binary format (**.dsb*).
selectionName	String. Name of the selection of results. This selection name will be included in the filename.
showDurationInFileName	Boolean. If true it adds the duration in hours of the simulation in the filename (e.g. *referencename_nbHoursH_selectionname.txt*).
showInitialDateInFileName	Boolean (optional). If true it adds the StartDate of the simulation to the filename (e.g. *yyyyMMddHH_referencename_selectionname.txt*).

Errors

Invalid Operation Exception	"Model not loaded"
Invalid Operation Exception	"Model 'hydrologicalmodel' has no results"
Argument Null Exception	"Reference name not defined"
Argument Null Exception	"Save folder not defined"
Argument Exception	"Save folder 'saveFolder' not valid"
Argument Null Exception	"Selection not defined"
Argument Exception	"Selection 'selectionName' not valid"
Invalid Operation Exception	"Failed to save selection results"

SaveInitializationDatasetAs *Saving of series of state variables*

Description

This procedure saves a dataset containing only the necessary state variables to initialize the model (i.e. the initial conditions of all the objects), at the specified location with the specified reference name and with or without the simulation length (in hours) or the initial date in the name.

Syntax

SaveInitializationDatasetAs (saveFolder, referenceName, saveDataInDstFile, showDurationInFileName, showInitialDateInFileName = true)

Parameters

saveFolder	String. Folder path where to save the file.
referenceName	String. Reference Name for the file.
saveDataInDstFile	Boolean. If true the dataset values are saved in text format (**.dst*), if false in binary format (**.dsb*).
showDurationInFileName	Boolean. If true it adds the duration in hours of the simulation in the filename (e.g. *referencename_nbHoursH.dsx*).
showInitialDateInFileName	Boolean (optional). If true it adds the StartDate of the simulation to the filename (e.g. *yyyyMMddHH_referencename.dsx*).

Errors

Invalid Operation Exception	“Model not loaded”
Invalid Operation Exception	“Model has no results”
Argument Null Exception	“Reference name not defined”
Argument Null Exception	“Save folder not defined”
Argument Exception	“Save folder not valid”
Invalid Operation Exception	“Failed to save full results”

SaveInitializationDataset *Saving of series of state variables*

Description

This procedure saves a dataset containing only the necessary state variables to initialize the model (i.e. the initial conditions of all the objects).

Syntax

SaveInitializationDataset (resultsDatasetFullPath, saveDataInDstFile)

Parameters

resultsDatasetFullPath	String. Full path of the file where is the full results dataset. The file extension has to be **.dsx*.
saveDataInDstFile	Boolean. If true the dataset values are saved in text format (**.dst*), if false in binary format (**.dsb*).

Errors

Invalid Operation Exception	“Model not loaded”
Invalid Operation Exception	“Model has no results”
Argument Null Exception	“Fullpath of full results dataset not defined”
Argument Exception	“Fullpath of full results dataset contains invalid characters”
Argument Exception	“Filename extension of full results dataset not valid”
Argument Exception	“Fullpath of full results dataset not rooted”
Invalid Operation Exception	“Failed to create and save full results dataset”

ExportResultsTo	*Exporting of the selection results*

Description

This procedure exports the selection results to an Excel or CSV file at the specified location with the name: *selectionFileFullPath*.

Syntax

ExportResultsTo (selectionFileFullPath, selectionName)

Parameters

selectionResultsFileFullPath	String. Full path of the selection results file. Acceptable file extensions are **.csv* and **.xlsx*.
selectionName	String. Name of the selection of results.

Errors

Invalid Operation Exception	“Model not loaded”
Invalid Operation Exception	“Model 'hydrologicalmodel' has no results”
Argument Null Exception	“Fullpath not defined”
Argument Exception	“Fullpath of selection results contains invalid characters”
Argument Exception	“Fullpath of selection results dataset not rooted”
Argument Exception	“Selection 'selectionName' not valid”
Argument Exception	“Fullpath extension not valid”
Invalid Operation Exception	“Failed to create and save selection results”

MergeFullResults	*Merging of full results dataset with the base full dataset*

Description

This procedure merges full results from the current simulation with the specified full results dataset (containing the same structure).

Syntax

MergeFullResults (resultsDatasetFullPath, saveDataInDstFile)

Parameters

resultsDatasetFullPath	String. Full path of full results dataset. The file extension has to be **.dsx*.
saveDataInDstFile	Boolean. If true the dataset values are saved in text format (**.dst*), if false in binary format (**.dsb*).

Errors

Invalid Operation Exception	"Model not loaded"
Invalid Operation Exception	"Model 'hydrologicalmodel' has no results"
Argument Null Exception	"Fullpath of full results dataset not defined"
Argument Exception	"Fullpath of full results dataset contains invalid characters"
Argument Exception	"Filename extension of full results dataset not valid"
Argument Exception	"Fullpath of full results dataset not rooted"
Invalid Operation Exception	"Failed to create new full results dataset"
Invalid Operation Exception	"Full results dataset not valid"
Invalid Operation Exception	"Failed to retrieve results to update full results dataset"
Invalid Operation Exception	"Failed to merge results with full results dataset"
Invalid Operation Exception	"Failed to save changes in full results dataset"

MergeSelectionResults	*Merging of selection results dataset with the base selection dataset*

Description

This procedure merges the selection results from the current simulation with the specified selection results dataset (containing the same structure).

Syntax

MergeSelectionResults (selectionResultsDatasetFullPath, saveDataInDstFile, selectionName)

Parameters

selectionResultsDatasetFull	String. Full path of the selection results file. The file

Path	extension has to be *.dsx*.
saveDataInDstFile	Boolean. If true the dataset values are saved in text format (*.dst*), if false in binary format (*.dsb*).
selectionName	String. Name of the selection of results

Errors

Invalid Operation Exception	“Model not loaded”
Invalid Operation Exception	“Model 'hydrologicalmodel' has no results”
Argument Null Exception	“Fullpath of selection results dataset not defined”
Argument Null Exception	“Selection not defined”
Argument Exception	“Selection 'selectionName' not valid”
Argument Exception	“Fullpath of selection results dataset contains invalid characters”
Argument Exception	“Filename extension of selection results dataset not valid”
Argument Exception	“Fullpath of selection results dataset not rooted”
Invalid Operation Exception	“Failed to create new selection results dataset”
Invalid Operation Exception	“Selection results dataset not valid”
Invalid Operation Exception	“Failed to retrieve results to update selection results dataset”
Invalid Operation Exception	“Failed to merge results with selection results dataset”
Invalid Operation Exception	“Failed to save changes in selection results dataset”

MergeDatasetsAs	*Merging of a source dataset with another dataset*

Description

This procedure merges a source dataset with a specified dataset (containing the same structure) at a specific location. If two stations have the same name, it can merge the data of both stations; the second dataset is prioritized for the merge of common dates in common stations.

Syntax

MergeDatasetsAs (nDataset1, nDataset2, MergedDatasetFullPath, merge=true, saveDataInDstFile=true)

Parameters

nDataset1	String. Full path of the first dataset to be merged. The file extension has to be *.dsx*.
nDataset2	String. Full path of the second dataset to be merged. The file extension has to be *.dsx*.
MergedDatasetFullPath	String. Full path of the dataset where the 2 datasets will

	be merged. The file extension has to be **.dsx*.
merge	Boolean. If true, the resulting dataset merge the data of stations with the same name. If false, it duplicates one of the stations with a "_(0)" suffix.
saveDataInDstFile	Boolean. If true, the dataset values are saved in text format (**.dst*), if false in binary format (**.dsb*).

Errors

Argument Null Exception	"First dataset not defined"
Argument Null Exception	"Second dataset not defined"
Argument Null Exception	"Fullpath of merged dataset not defined"
Argument Exception	"Fullpath of full merged dataset contains invalid characters"
Argument Exception	"Filename extension of merged dataset not valid"
Argument Exception	"Fullpath of full merged dataset not rooted"
Invalid Operation Exception	"First dataset not valid"
Invalid Operation Exception	"Second dataset not valid"
Invalid Operation Exception	"Failed to create merged dataset"

MergeDatasets	*Merging of a source dataset with another dataset*

Description

This procedure merges a source dataset with a specified dataset (containing the same structure) and saves the resulting dataset in the source dataset. If two stations have the same name, it can merge the data of both stations; the second dataset (update dataset) is prioritized for the merge of common dates in common stations.

Syntax

MergeDatasets (sourceDatasetFullPath, updateDatasetFullPath, merge=true)

Parameters

sourceDatasetFullPath	String. Full path of the source dataset where the new dataset is added. The file extension has to be **.dsx*.
updateDatasetFullPath	String. Full path of the update dataset to be added to the source dataset. The file extension has to be **.dsx*.
merge	Boolean. If true, the resulting dataset merge the data of stations with the same name. If false, it duplicates one of the stations with a "_(0)" suffix.

Errors

Argument Null Exception	"First dataset not defined"

Argument Null Exception	“Second dataset not defined”
Invalid Operation Exception	“First dataset not valid”
Invalid Operation Exception	“Second dataset not valid”
Invalid Operation Exception	“Failed to merge datasets”

GetPropertyValue	*Getting property value from an object*

Description

This function gets the value of a parameter or an initial condition of an object.

Syntax

GetPropertyValue (objectType, propertyName, zone = null)

Parameters

objectType	String. Object type (e.g. SOCONT, GSM, Reservoir, etc.). See Table 27.
propertyName	String. Property name (e.g. HGR3Max). See Table 27.
zone	String (optional). Zone where the objects are.

Returned value

property value	It returns the property value (parameter or initial condition) of the object.

Errors

Invalid Operation Exception	“Model not loaded"
Argument Exception	“Object type 'objectType' not valid”
Argument Null Exception	“Property name not defined”
Argument Exception	“Zone 'zone' not found in 'hydrologicalmodel”
Invalid Operation Exception	“Failed to provide model property”
Invalid Operation Exception	“Model property not found”

GetIndicatorValue	*Getting the value of an indicator from a comparator*

Description

This function gets the specified indicator value from a Comparator object.

Syntax

GetIndicatorValue (indicatorName, objectName)

Parameters

indicatorName	String. Indicator name. Indicators: Nash, Nash-ln, Pearson Correlation Coeff, Kling-Gupta Efficiency, Bias Score, RRMSE, Relative Volume Bias, Normalized Peak Error
objectName	String. User-defined object name.

Returned value

indicator value	It returns the value of the indicator.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Exception	“Comparator 'objectName' not found”
Argument Exception	“Indicator name 'indicatorName' not valid”

6.8. Calibration

LoadCalibrationsInModel	*Loading of all the calibrations defined in a model*

Description

This procedure loads all the calibration configurations defined in the opened model.

Syntax

LoadCalibrationsInModel ()

Errors

Invalid Operation Exception	"Model not loaded"

LoadCalibrationsByFile	*Loading of all the calibration configurations defined in a configuration file*

Description

This procedure loads all the calibration configurations defined in a specified configuration file.

Syntax

LoadCalibrationsByFile (filePath)

Parameters

filePath	String. Full path of the calibration configuration file defined in the Calibrator module of RS MINERVE. The file extension has to be **.cbt*.

Errors

Argument Null Exception	"File path is not defined"
Argument Exception	"Selected file is not a XML file"
Invalid Operation Exception	"No Calibration configuration is loaded"

SaveAllCalibrationsInModel	*Saving all calibration configurations in the RSM model*

Description

This procedure saves all the calibration configurations in the loaded RSM model. Configurations will then be saved in the *.rsm* file).

Syntax

SaveAllCalibrationsInModel ()

Errors

Invalid Operation Exception	"Model not loaded"

RunCalibration *Running a specified calibration*

Description

This procedure runs a specified calibration available among the loaded configuration(s).

Syntax

RunCalibration (configName, logFilePath, deleteLogAfter=false)

Parameters

configName	String. Name given in the Calibrator module of RS MINERVE to the calibration configuration to be executed.
logFilePath	String. Full path of the log file where the logs are written.
deleteLogAfter	Boolean. If true, the log file is deleted after the calibration process is ended.

Errors

Argument Null Exception	"The name of the calibration configuration is not defined"
Argument Exception	"No calibration configuration called configName"
Invalid Operation Exception	"The model has fatal errors and cannot continue"

RunAllCalibrations *Running all the calibrations*

Description

This procedure runs all the calibration configurations previously loaded.

Syntax

RunAllCalibrations (logFilePath, deleteLogAfter=false)

Parameters

logFilePath	String. Full path of the log file where the logs are written.
deleteLogAfter	Boolean. If true, the log file is deleted after the calibration process is ended.

Errors

Invalid Operation Exception	"The model has fatal errors and cannot continue"

ExportCalibrationResults *Exporting of the calibration's results*

Description

This procedure exports the results of a specified calibration, to a *.txt* file.

Syntax

ExportCalibrationResults (configName, resultFilePath)

Parameters

configName	String. Name given in the Calibrator module of RS MINERVE to the calibration configuration to be executed.
resultFilePath	String. Full path of the file where to save the calibration results. The file extension has to be **.txt*.

Errors

Argument Null Exception	“The name of the calibration configuration is not defined”
Argument Exception	“The file path for the results is not defined”
Argument Exception	“No calibration configuration called configName”
Invalid Operation Exception	“There are no results to export in calibration configName”

ExportAllCalibrationsResults	*Exporting of all the calibrations’ results*

Description

This procedure exports the results of all the calibration configurations loaded, to a *.txt* file.

Syntax

ExportAllCalibrationsResults (resultFolderPath)

Parameters

resultFolderPath	String. Folder path to export the results of all the calibration configurations. The exported results file is named as *ConfigName.txt*, where ConfigName is the name of each calibration configuration executed.

Errors

Argument Null Exception	“The folder path for the results is not defined”
Argument Exception	“Selected folder does not exist”
Invalid Operation Exception	“There are no results to export for some calibrations”

SetCalibrationDates	*Setting of the start and end dates of the calibration period*

Description

This procedure defines the start and end dates of the period used by the Calibrator module to execute the calibration.

Syntax

SetCalibrationDates (configName, startDate, endDate)

Parameters

configName	String. Name given in the Calibrator module of RS MINERVE to the calibration configuration for which the start and end dates are to be set.
startDate	Date. Start date for the calibration period. The format of the date has to be: dd.MM.yyyy HH:mm:ss.
endDate	Date. End date for the calibration period. The format of the date has to be: dd.MM.yyyy HH:mm:ss.

Errors

Invalid Operation Exception	“Model not loaded”
Argument Null Exception	“The name of the calibration configuration is not defined”
Argument Exception	“No calibration configuration called configName”

6.9. Application Stop

Stop	*Stop of the execution*

Description

This procedure stops the execution of the model.

Syntax

Stop ()

6.10. Other procedures/functions

RemoveFile	*Removing any type of files*

Description

This procedure remove for the computer any file given by its full path.

Syntax

RemoveFile (fileFullPath)

Parameters

fileFullPath	String. Full path of the file to be removed. The file extension of the file must be specified.

Errors

Argument Null Exception	“File not defined”
Argument Exception	“File not found”
Invalid Operation Exception	“Failed to remove the file”

GetAppVersion	*Getting the installed RS MINERVE software version*

Description

This function gets the version number of the installed RS MINERVE software.

Syntax

GetAppVersion ()

Returned value

Software version	It returns the version number of the installed RS MINERVE software (e.g., ***2.6.2.0***).

GetReaderVersion	*Getting the installed model reader (xml) version*

Description

This function gets the number of the installed model reader (xml) version.

Syntax

GetReaderVersion ()

Returned value

Reader version	It returns the number of the installed model reader (xml) version (e.g., ***4.001***).

6.11. Objects nomenclature in VBScript

When using VBScripts to perform tasks within RS MINERVE, it is necessary to indicate the type of the object or the name of a property as arguments for some functions (**DefineModelProperties**, **UpdateModelProperties** and **GetPropertyValue**). Table 27 gathers the proper nomenclatures of these arguments to be used in the corresponding VBScripts functions.

Table 27 Nomenclature of objects, parameters and initial condition to be used in the **GetPropertyValue** function

Object class	Names to be used in VBScripts		
	objectType	propertyName	
		Parameters	Initial conditions
Hydrology	Station	X, Y, Z, Search Radius, No. min. of stations, Gradient P, Gradient T, Gradient ETP, Coeff P, Coeff T, Coeff ETP	-
	SnowSD	S, SInt, SMin, SPh, ThetaCri, bp, Tcp1, Tcp2, Tcf	SWEIni, ThetaIni
	SWMM	A, L, J0, K	HIni
	GSM	A, S, SInt, SMin, SPh, ThetaCri, bp, Tcp1, Tcp2, Tcf, G, GInt, GMin, Tcg, Kgl, Ksn	SWEIni, ThetaIni, QsnowIni, QglacierIni, HglacierIni
	SOCONT	A, S, SInt, SMin, SPh, ThetaCri, bp, Tcp1, Tcp2, Tcf, HGR3Max, KGR3, L, J0, Kr	SWEIni, ThetaIni, HGR3Ini, HrIni
	HBV	A, CFMax, CFR, CWH, TT, TTInt, TTSM, Beta, FC, PWP, SUMax, Kr, Ku, Kl, Kperc	SWEIni, WHIni, HumIni, SUIni, SLIni
	GR4J	A, X1, X2, X3, X4	Sini, RIni
	SAC	A, Adimp, Pctim, Riva, UztwMax, UzfwMax, Uzk, Zperc, Rexp, Pfree, LztwMax, LzfpMax, LzfsMax, Rserv, Lzpk, Lzsk, Side	AdimIni, UztwIni, UzfwIni, LztwIni, LzfpIni, LzfsIni
River	LagTime	L, B0, m, J0, K, lag	Qini
	Kinematic	L, B0, m, J0, K, N	Qini
	MuskingumCunge		
	StVenant		
Infrastructure	Reservoir	-	Hini, HvarIni
	Hydropower	Zplant, L, D, K, v, Default Price	-
	Consumer	Default QDemand, LossRate, ConsumptionRate	-
	StructureEfficiency	Efficiency	-

Chapter 7. File formats

RS MINERVE generates model files saved in **.rsm** format as well as different other formats for results of configuration data.

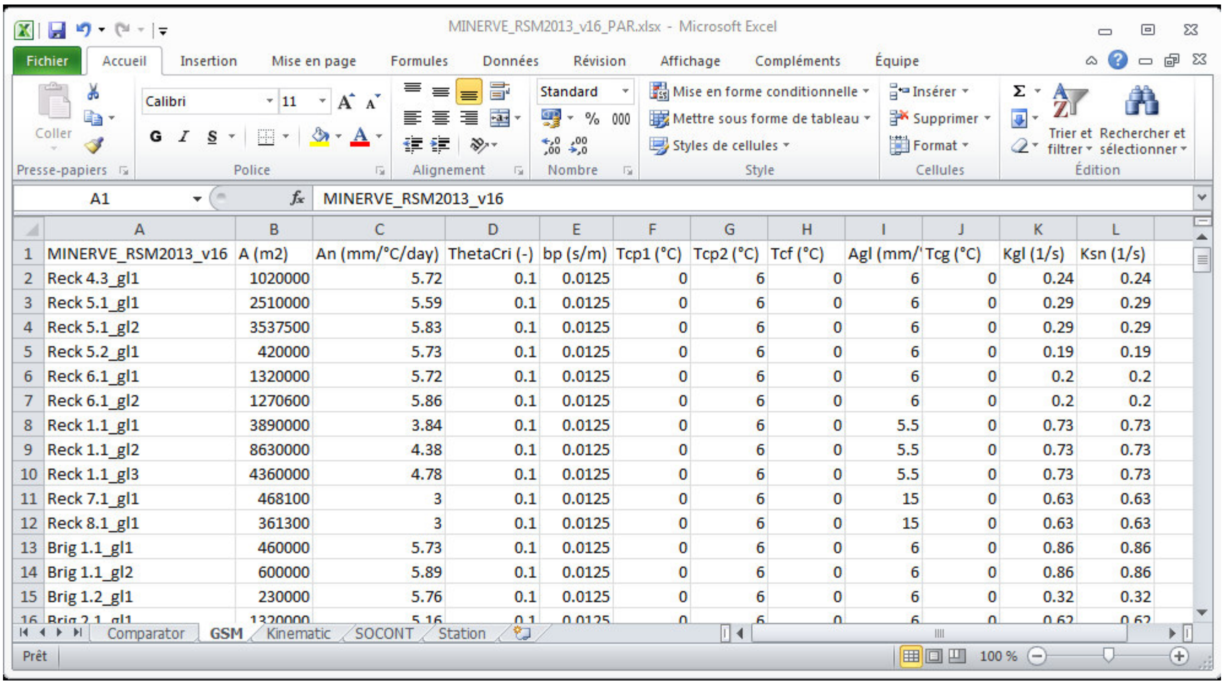
7.1. Model files

Both *Save* and *Export* generate files of the same format (***.rsm***). With *Save*, the entire model is saved, whereas with *Export*, only the objects (submodels included) contained in the active hierarchical level are exported (refer to the **Exportation of a submodel** section in the User's Manual).

7.2. Parameters and initial conditions files

Parameters (or initial conditions) of the model can be saved as:

- Excel files (***.xlsx***). Each worksheet presents the parameters (or initial conditions) of one hydrological object, as presented in the example of the Figure 28.
- Text files (***.txt***). Each hydrological object is presented per line with its parameters or initial conditions



	A	B	C	D	E	F	G	H	I	J	K	L
1	MINERVE_RSM2013_v16	A (m2)	An (mm/°C/day)	ThetaCri (-)	bp (s/m)	Tcp1 (°C)	Tcp2 (°C)	Tcf (°C)	Agl (mm/'	Tcg (°C)	Kgl (1/s)	Ksn (1/s)
2	Reck 4.3_gl1	1020000	5.72	0.1	0.0125	0	6	0	6	0	0.24	0.24
3	Reck 5.1_gl1	2510000	5.59	0.1	0.0125	0	6	0	6	0	0.29	0.29
4	Reck 5.1_gl2	3537500	5.83	0.1	0.0125	0	6	0	6	0	0.29	0.29
5	Reck 5.2_gl1	420000	5.73	0.1	0.0125	0	6	0	6	0	0.19	0.19
6	Reck 6.1_gl1	1320000	5.72	0.1	0.0125	0	6	0	6	0	0.2	0.2
7	Reck 6.1_gl2	1270600	5.86	0.1	0.0125	0	6	0	6	0	0.2	0.2
8	Reck 1.1_gl1	3890000	3.84	0.1	0.0125	0	6	0	5.5	0	0.73	0.73
9	Reck 1.1_gl2	8630000	4.38	0.1	0.0125	0	6	0	5.5	0	0.73	0.73
10	Reck 1.1_gl3	4360000	4.78	0.1	0.0125	0	6	0	5.5	0	0.73	0.73
11	Reck 7.1_gl1	468100	3	0.1	0.0125	0	6	0	15	0	0.63	0.63
12	Reck 8.1_gl1	361300	3	0.1	0.0125	0	6	0	15	0	0.63	0.63
13	Brig 1.1_gl1	460000	5.73	0.1	0.0125	0	6	0	6	0	0.86	0.86
14	Brig 1.1_gl2	600000	5.89	0.1	0.0125	0	6	0	6	0	0.86	0.86
15	Brig 1.2_gl1	230000	5.76	0.1	0.0125	0	6	0	6	0	0.32	0.32
16	Brig 2.1_gl1	1320000	5.16	0.1	0.0125	0	6	0	6	0	0.62	0.62

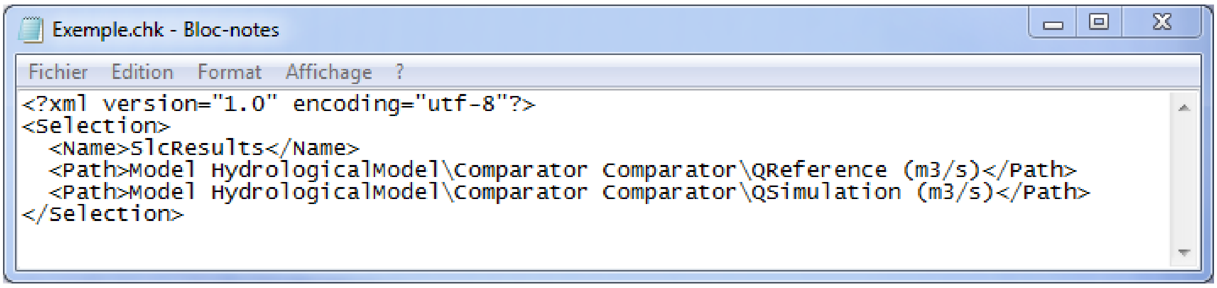
Figure 28 Example of the parameters of the GSM hydrological models in an Excel file

7.3. Object selection files

A selection of the objects data can be saved with the help of the "Selection and plots". The exported selection is saved as a ***.chk*** file. The format of this file is presented in Figure 29. It contains:

- the name of the selection (given by the user in the "Selection and plots") in first line,
- the number of selected series (second line),

- all selections, with the name of the **.rsm** model, the name submodel (if it exists), the name of the selected object and its chosen series.



```
<?xml version="1.0" encoding="utf-8"?>
<Selection>
  <Name>SlcResults</Name>
  <Path>Model HydrologicalModel\Comparator Comparator\QReference (m3/s)</Path>
  <Path>Model HydrologicalModel\Comparator Comparator\QSimulation (m3/s)</Path>
</Selection>
```

Figure 29 Example of selected series saved in a .chk file

7.4. Exported results files

The results in the model selection can be also achieved from the “*Selection and plots*”. Several file formats are possible:

- Excel file (***.xlsx***).
- CSV file (***.csv***).
- Aquatool input file (**.apo**), for use’s working with Aquatool DMA from UPV.

7.5. Calibration configuration files

The full configuration of the calibration, with information such as the selected objects, parameters to calibrate, or the objective function to take into account, is saved in a **.cbt** file.

7.6. Database files

The format of the database is explained in detail in chapter 7. Databases contains two different files

- Database files in text format (***.dbt*** and ***.dbx***) or in binary format (***.dbb*** and ***.dbx***).
- Dataset files in text format (***.dst*** and ***.dsx***) or in binary format (***.dsb*** and ***.dsx***).
- CSV files in text format (***.csv***).

Chapter 8. Database formats

The different input data as well as exported results are managed within a database. The *Database Viewer,* accessible from the *RS MINERVE* window, is used to create or edit the database linked to the active model.

8.1. The Database Viewer

The *Database Viewer* window appears when a database is created (✚ *Add)* or opened for edition (🛢 *Open*).

The database structure, presented in Table 28, is organized in five hierarchical levels:

- **Database**
  - Description of the database
  - Contains one or more groups
- **Group**
  - Separation based on category of data (Measures, Forecasts, Simulations,) 💡[3]
  - Contains one or more datasets
- **Dataset**
  - Set of data of common type (Meteo data, Flow data,...) 💡1
  - Contains one or more stations
- **Station**
  - Information about the station (name and coordinates)
  - Contains one or more sensors
- **Sensor**
  - Description of the sensor (name, units and data)
  - Contains one data series

The user can create a complete database or different datasets to add to an existing database.

[3] Definition and use of *Groups* and *Datasets* can also be done in a different way by the user.

Table 28 Structure of the database interface, with editable information in italic blue

Node	Name	Description	Example
Database			
	Description	*Name of the database*	*Database_tutorial*
	Date creation	Date of creation of database	14.12.2012 13:32
	Date modification	Date of last modification	04.01.2013 14:30
	Time zone	*Tine scale of the data*	*UTC+01:00*
	Filename	File name and path	(Full path of file)
	XML file	XML file size	21 Ko
	Data file	Data file size	161 Ko
Group			
	Description	*Name of the group*	*Measure*
	Category	*Type of data (Inputs, Outputs or None)*	*Inputs*
	Date capture	Date of last dataset capture date in the group	18/12/2012 10:54
Dataset			
	Description	*Name of the dataset*	*Meteo_data*
	Dataset creation	Dataset creation date	14.12.2012 00:00
	Dataset reception	Date of receipt of data	14.12.2012 03:00
	Data modification	Date of last modification	14.12.2012 03:00
	Data capture	*First date of the data (editable)*	*14.12.2012 00:00*
	Time zone	Tine scale of the data	UTC+01:00
	Filename	File name and path	(Full path of file)
	XML file	XML file size	21 Ko
	Data file	Data file size	161 Ko
Station			
	Name	*Name of the station*	*Station_meteo_1*
	X	*Coordinate X in meter*	*650000*
	Y	*Coordinate Y in meter*	*130000*
	Z	*Altitude in meter above sea level*	*1957*
Sensor			
	Description	*Name of the variable*	*Precipitation*
	Category	*Variable type*	*Precipitation*
	Unit	*Variable unit*	*MillimetersPerHour*
	Interpolation [4]	*Data assimilation method*	*Linear*
	Initial date	The first date of the time series	01/01/2011 00:00:00
	Final date	The last date of the time series	31/12/2013 00:00:00
	Min value	Minimum value of the series	0 m^3/s
	Average value	Average value of the series	1.1884 m^3/s
	Max value	Maximum value of the series	3.2485 m^3/s

[4] Data can be interpolated in three different ways: *linear*, *constant before* and *constant after*. With *constant before*, the value for a given time is used as a constant for the preceding time interval, whereas the *constant after* method applies the value for the following time interval.

8.2. Database, dataset and CSV formats

Databases and datasets also correspond to different saving formats:

- Database files **([.dbt and .dbx] in text format or [.dbb and .dbx] in binary format)** can be created, opened, imported, exported or saved from the complete database (database level).
- Dataset files **([.dst and .dsx] in text format or [.dsb and .dsx] in binary format)** can also be created, opened, imported, exported or saved from the Database (dataset level).
- CSV files **([.csv] in text format)** can also be created, opened, imported or exported from the Database (dataset level).

Database files

The database file includes two different files:

- A XML file (*.dbx) describing the structure of the database (groups, datasets, stations and sensors)
- A file containing the data series. This file can be in text format (*.dbt) or binary format (*.dbb). Binary format allows to reduce the file size.

Note that in the text file (*.dbt), if the database is not empty and contains one or more sensors (then at least a dataset with at least a station)[5], each data series should be composed of:

- A header containing four fields separated by a backslash character: Group\Dataset\Station\Sensor. Each field name must be identical to the name found in the corresponding <Name> node in the XML file (*.dbx) (Figure 30);
- Two columns (date and data) separated by a tabulation character:

[5] If the database is not containing sensor, the data file (*.dbt or *.dbb) will be empty.

***.dbx file:**

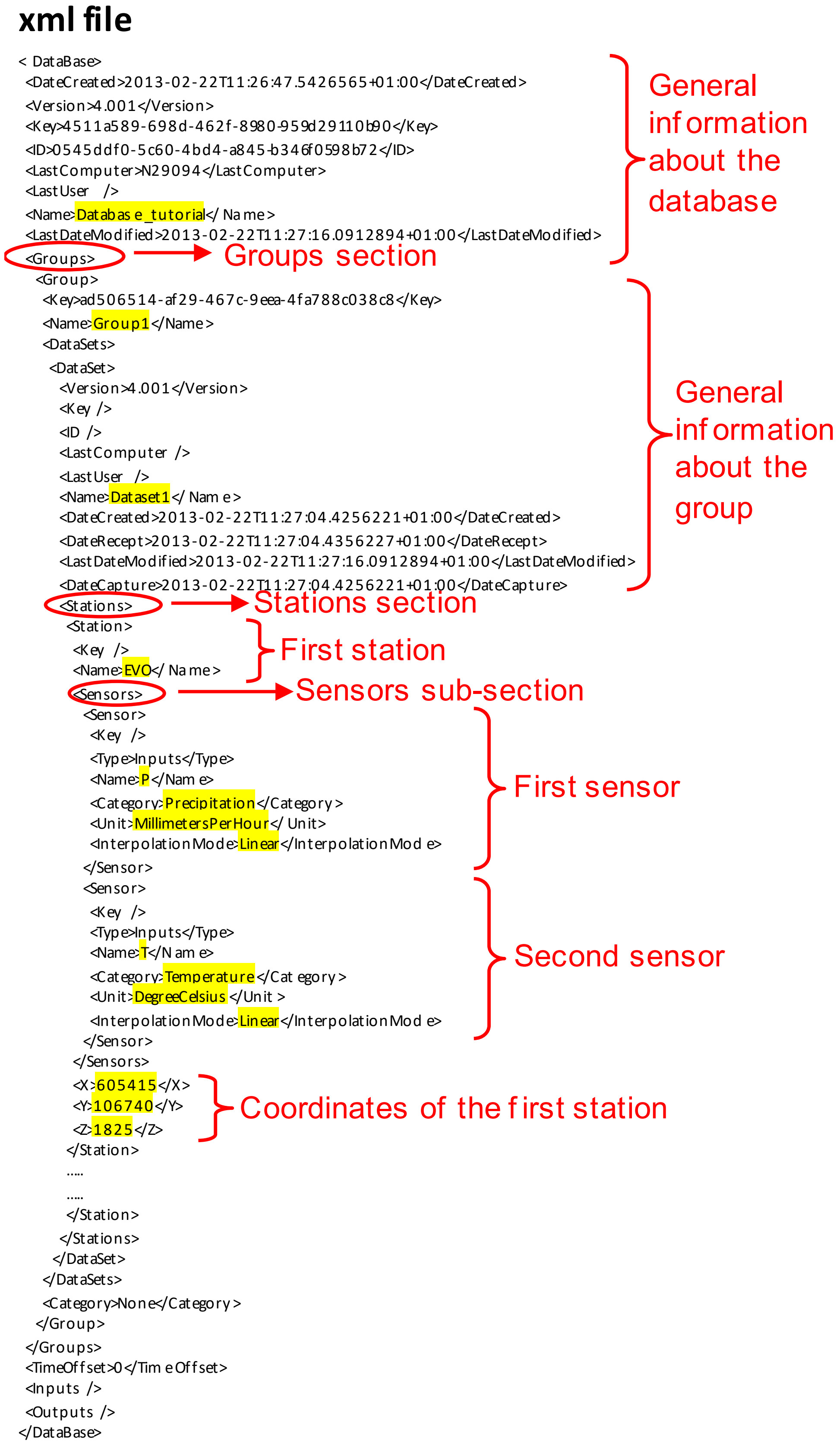
```
  …
  <Group>
      …
      <Name>Group1</Name>
      …
        <Dataset>
              …
              <Name>Dataset1</Name>
                  …
                  <Station>
                        …
                        <Name>EVO</Name>
                            …
                            <Sensor>
                                …
                                <Name>P</Name>
                            …
                            </Sensor>
                  …
                  </Station>
        …
        </Dataset>
  …
  </Group>
…
```

***.dbt file:**

```
Group1\Dataset1\EVO\P
31.03.2014 15:00    57.3
…
```

Figure 30 Format of the two database files (in text format for the data file)

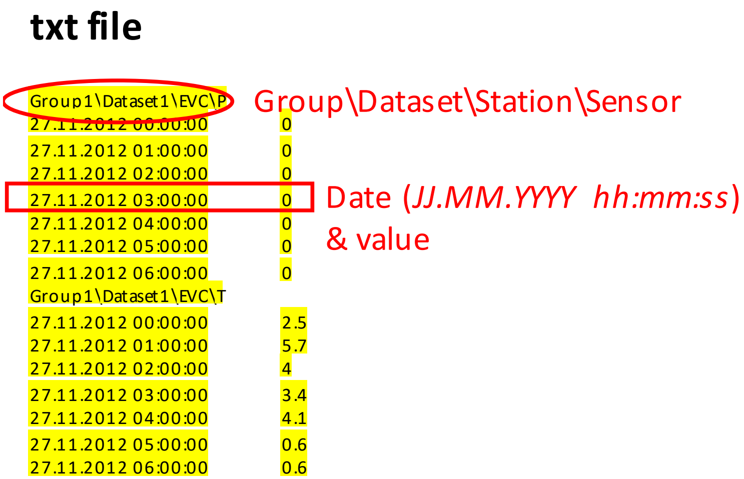
An example of the XML file and of the text file is showed respectively in Figure 31 and Figure 32.



```
< DataBase>
 <DateCreated>2013-02-22T11:26:47.5426565+01:00</DateCreated>
 <Version>4.001</Version>
 <Key>4511a589-698d-462f-8980-959d29110b90</Key>
 <ID>0545ddf0-5c60-4bd4-a845-b346f0598b72</ID>
 <LastComputer>N29094</LastComputer>
 <LastUser />
 <Name>Database_tutorial</ Name>
 <LastDateModified>2013-02-22T11:27:16.0912894+01:00</LastDateModified>
 <Groups>
  <Group>
   <Key>ad506514-af29-467c-9eea-4fa788c038c8</Key>
   <Name>Group1</Name>
   <DataSets>
    <DataSet>
     <Version>4.001</Version>
     <Key />
     <ID />
     <LastComputer />
     <LastUser />
     <Name>Dataset1</ Name>
     <DateCreated>2013-02-22T11:27:04.4256221+01:00</DateCreated>
     <DateRecept>2013-02-22T11:27:04.4356227+01:00</DateRecept>
     <LastDateModified>2013-02-22T11:27:16.0912894+01:00</LastDateModified>
     <DateCapture>2013-02-22T11:27:04.4256221+01:00</DateCapture>
     <Stations>
      <Station>
       <Key />
       <Name>EVO</ Name>
      <Sensors>
       <Sensor>
        <Key />
        <Type>Inputs</Type>
        <Name>P</Name>
        <Category>Precipitation</Category>
        <Unit>MillimetersPerHour</ Unit>
        <InterpolationMode>Linear</InterpolationMode>
       </Sensor>
       <Sensor>
        <Key />
        <Type>Inputs</Type>
        <Name>T</Name>
        <Category>Temperature</Category>
        <Unit>DegreeCelsius </Unit >
        <InterpolationMode>Linear</InterpolationMode>
       </Sensor>
      </Sensors>
      <X>605415</X>
      <Y>106740</Y>
      <Z>1825</Z>
      </Station>
      .....
      .....
      </Station>
     </Stations>
    </DataSet>
   </DataSets>
   <Category>None</Category>
  </Group>
 </Groups>
 <TimeOffset>0</TimeOffset>
 <Inputs />
 <Outputs />
</DataBase>
```

xml file
General information about the database
Groups section
General information about the group
Stations section
First station
Sensors sub-section
First sensor
Second sensor
Coordinates of the first station

Figure 31 Example of the database XML file



txt file

```
Group1\Dataset1\EVC\P
27.11.2012 00:00:00	0
27.11.2012 01:00:00	0
27.11.2012 02:00:00	0
27.11.2012 03:00:00	0
27.11.2012 04:00:00	0
27.11.2012 05:00:00	0
27.11.2012 06:00:00	0
Group1\Dataset1\EVC\T
27.11.2012 00:00:00	2.5
27.11.2012 01:00:00	5.7
27.11.2012 02:00:00	4
27.11.2012 03:00:00	3.4
27.11.2012 04:00:00	4.1
27.11.2012 05:00:00	0.6
27.11.2012 06:00:00	0.6
```

Group\Dataset\Station\Sensor

Date (*JJ.MM.YYYY hh:mm:ss*)
& value

Figure 32 Example of the database data file in text format

Dataset files

The dataset file includes also two different files (Figure 34 and Figure 35):

- A XML file (*.dsx) describing the structure of the dataset (stations and sensors)
- A file containing the data series. This file can be in text (*.dst) or in binary (*.dsb) format. This last format allows to reduce the file size.

Note that in the text file (*.dst), if the dataset is not empty and contains one or more sensors (then at least a dataset with at least a station)[6], each data series should be composed of:

- A header containing two fields separated by a backslash character: Station\Sensor. Each field name must be identical to the name found in the corresponding <Name> node in the XML file (*.dsx) (Figure 31);
- Then two columns (date and data) separated by a tabulation character.

[6] If the dataset is not containing sensor, the data file (*.dst or *.dsb) will be empty.

```
*.dbx file
…
<Station>
  …
  <Name>EVO</Name>
  …
  <Sensor>
    …
    <Name>P</Name>
    …
  </Sensor>
  …
</Station>
…
```

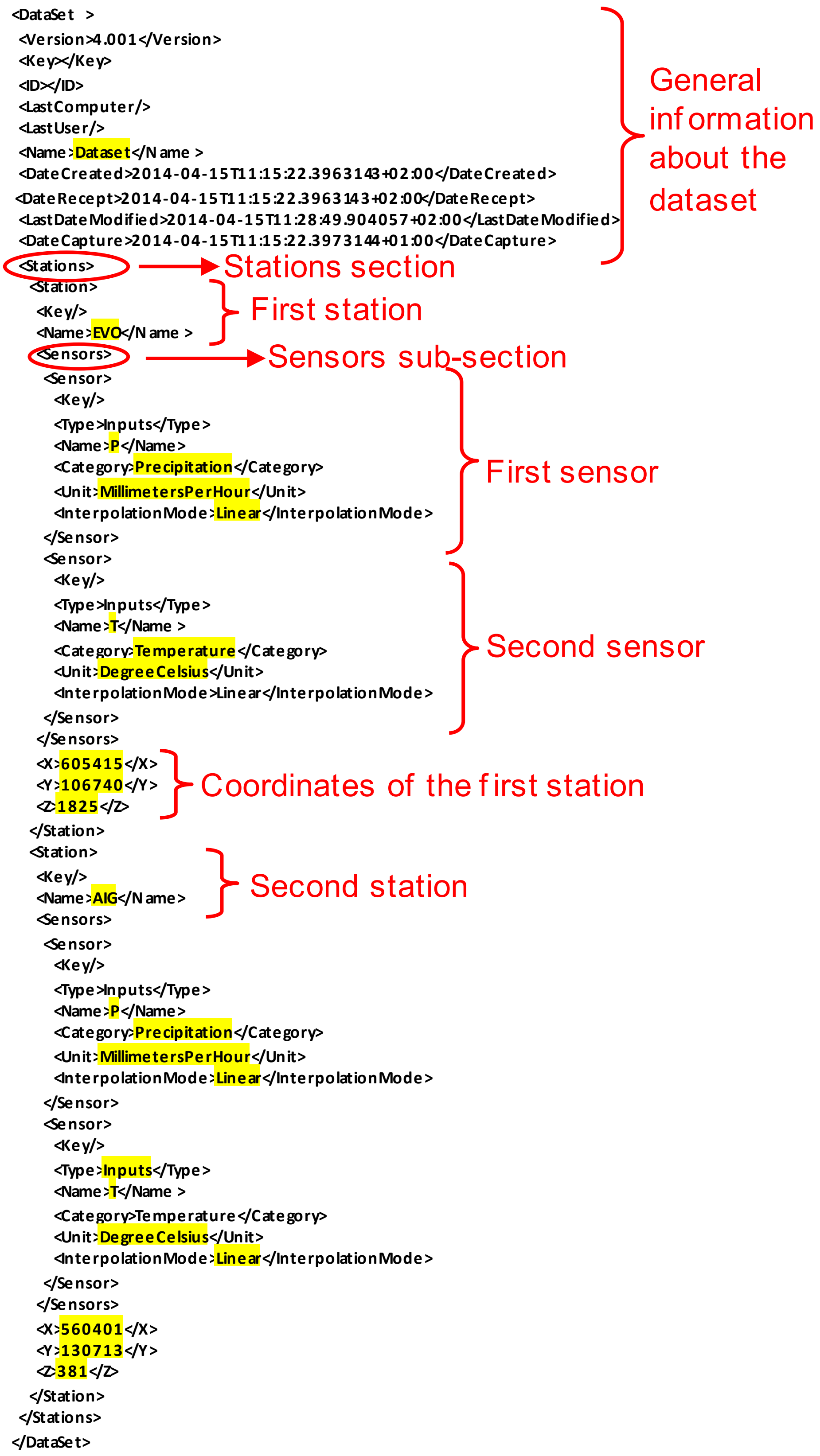
```
*.dst file

EVO\P
31.03.2014 15:00     57.3
```

Figure 33 Format of the two dataset files (in text format for the data file)

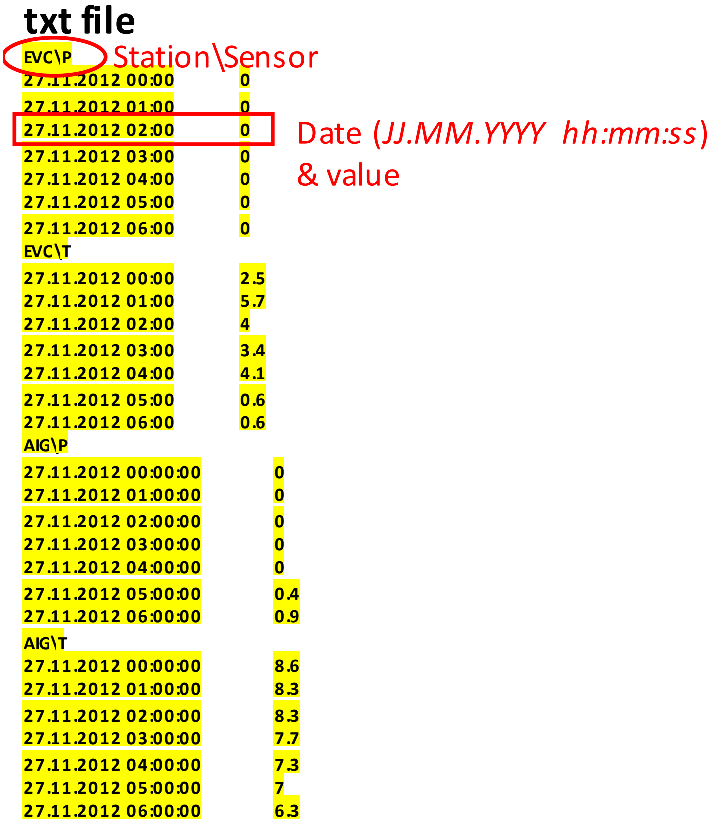
An example of the XML and text files is showed respectively in Figure 34 and Figure 35.

xml file



```
<DataSet >
 <Version>4.001</Version>
 <Key></Key>
 <ID></ID>
 <LastComputer/>
 <LastUser/>
 <Name>Dataset</Name >
 <DateCreated>2014-04-15T11:15:22.3963143+02:00</DateCreated>
<DateRecept>2014-04-15T11:15:22.3963143+02:00</DateRecept>
 <LastDateModified>2014-04-15T11:28:49.904057+02:00</LastDateModified>
 <DateCapture>2014-04-15T11:15:22.3973144+01:00</DateCapture>
 <Stations>
  <Station>
   <Key/>
   <Name>EVO</Name >
   <Sensors>
    <Sensor>
     <Key/>
     <Type>Inputs</Type>
     <Name>P</Name>
     <Category>Precipitation</Category>
     <Unit>MillimetersPerHour</Unit>
     <InterpolationMode>Linear</InterpolationMode>
    </Sensor>
    <Sensor>
     <Key/>
     <Type>Inputs</Type>
     <Name>T</Name >
     <Category>Temperature</Category>
     <Unit>DegreeCelsius</Unit>
     <InterpolationMode>Linear</InterpolationMode>
    </Sensor>
   </Sensors>
   <X>605415</X>
   <Y>106740</Y>
   <Z>1825</Z>
  </Station>
  <Station>
   <Key/>
   <Name>AIG</Name>
   <Sensors>
    <Sensor>
     <Key/>
     <Type>Inputs</Type>
     <Name>P</Name>
     <Category>Precipitation</Category>
     <Unit>MillimetersPerHour</Unit>
     <InterpolationMode>Linear</InterpolationMode>
    </Sensor>
    <Sensor>
     <Key/>
     <Type>Inputs</Type>
     <Name>T</Name >
     <Category>Temperature</Category>
     <Unit>DegreeCelsius</Unit>
     <InterpolationMode>Linear</InterpolationMode>
    </Sensor>
   </Sensors>
   <X>560401</X>
   <Y>130713</Y>
   <Z>381</Z>
  </Station>
 </Stations>
</DataSet>
```

Figure 34 Example of the dataset XML file



txt file

Station\Sensor

Date (*JJ.MM.YYYY hh:mm:ss*) & value

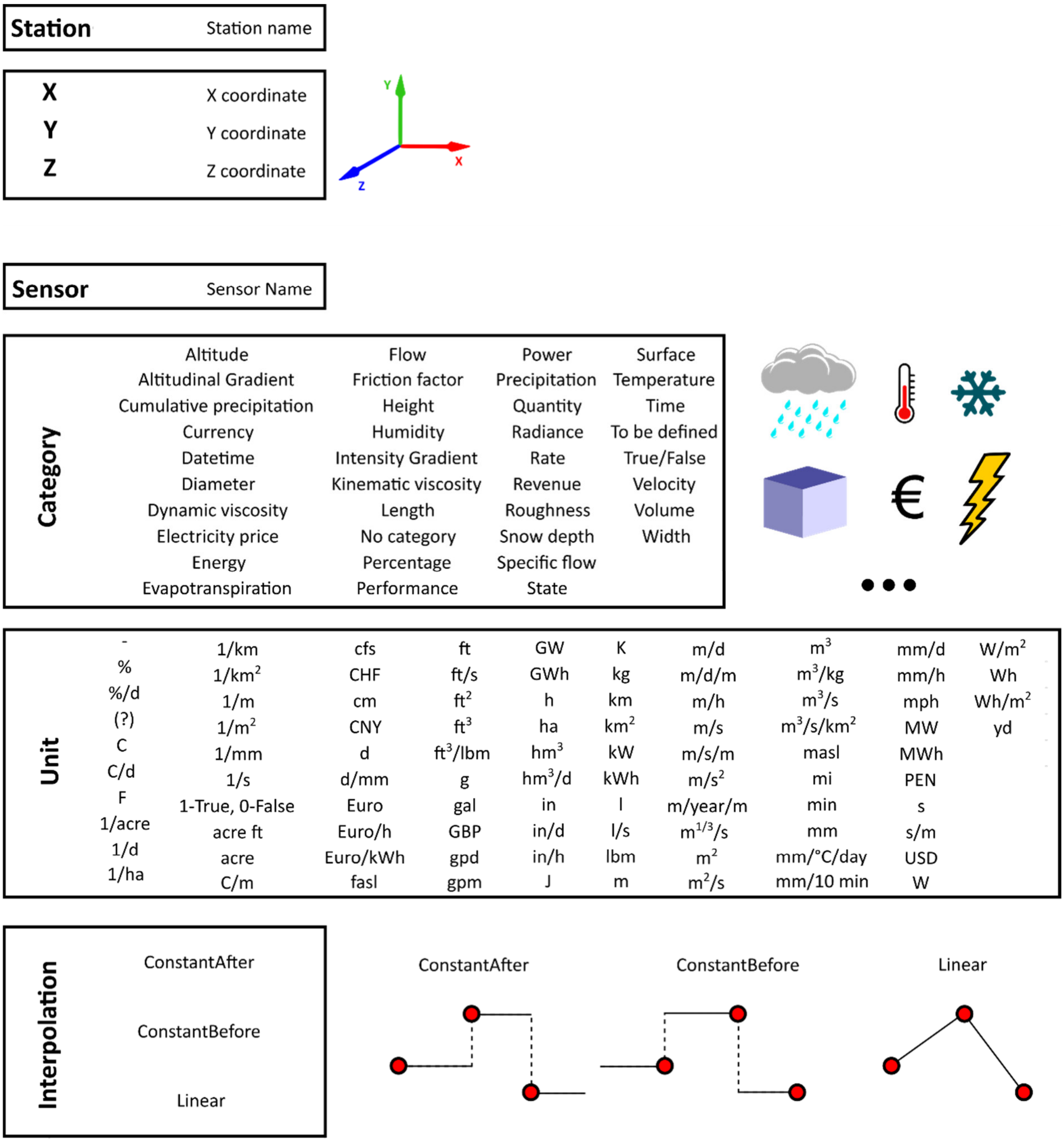
```
EVC\P
27.11.2012 00:00    0
27.11.2012 01:00    0
27.11.2012 02:00    0
27.11.2012 03:00    0
27.11.2012 04:00    0
27.11.2012 05:00    0
27.11.2012 06:00    0
EVC\T
27.11.2012 00:00    2.5
27.11.2012 01:00    5.7
27.11.2012 02:00    4
27.11.2012 03:00    3.4
27.11.2012 04:00    4.1
27.11.2012 05:00    0.6
27.11.2012 06:00    0.6
AIG\P
27.11.2012 00:00:00      0
27.11.2012 01:00:00      0
27.11.2012 02:00:00      0
27.11.2012 03:00:00      0
27.11.2012 04:00:00      0
27.11.2012 05:00:00      0.4
27.11.2012 06:00:00      0.9
AIG\T
27.11.2012 00:00:00      8.6
27.11.2012 01:00:00      8.3
27.11.2012 02:00:00      8.3
27.11.2012 03:00:00      7.7
27.11.2012 04:00:00      7.3
27.11.2012 05:00:00      7
27.11.2012 06:00:00      6.3
```

Figure 35 Example of the dataset data file in text format

CSV files

CSV files can be used to import and export Dataset files. It is possible to open a CSV file and RS MINERVE will create a Database required to contain the Dataset. The structure of the CSV files is showed in Figure 36.

IMPORTANT: RS MINERVE will automatically remove ‘NA’, ‘NaN’, ‘N/A’ or ‘NULL’ values contained in the time series before creating the dataset.



Station Station name

X X coordinate

Y Y coordinate

Z Z coordinate

Sensor Sensor Name

Category

Altitude
Altitudinal Gradient
Cumulative precipitation
Currency
Datetime
Diameter
Dynamic viscosity
Electricity price
Energy
Evapotranspiration
Flow
Friction factor
Height
Humidity
Intensity Gradient
Kinematic viscosity
Length
No category
Percentage
Performance
Power
Precipitation
Quantity
Radiance
Rate
Revenue
Roughness
Snow depth
Specific flow
State
Surface
Temperature
Time
To be defined
True/False
Velocity
Volume
Width

Unit

-
%
%/d
(?)
C
C/d
F
1/acre
1/d
1/ha
1/km
$1/km^2$
1/m
$1/m^2$
1/mm
1/s
1-True, 0-False
acre ft
acre
C/m
cfs
CHF
cm
CNY
d
d/mm
Euro
Euro/h
Euro/kWh
fasl
ft
ft/s
ft^2
ft^3
ft^3/lbm
g
gal
GBP
gpd
gpm
GW
GWh
h
ha
hm^3
hm^3/d
in
in/d
in/h
J
K
kg
km
km^2
kW
kWh
l
l/s
lbm
m
m/d
m/d/m
m/h
m/s
m/s/m
m/s^2
m/year/m
$m^{1/3}/s$
m^2
m^2/s
m^3
m^3/kg
m^3/s
$m^3/s/km^2$
masl
mi
min
mm
mm/°C/day
mm/10 min
mm/d
mm/h
mph
MW
MWh
PEN
s
s/m
USD
W
W/m^2
Wh
Wh/m^2
yd

Interpolation

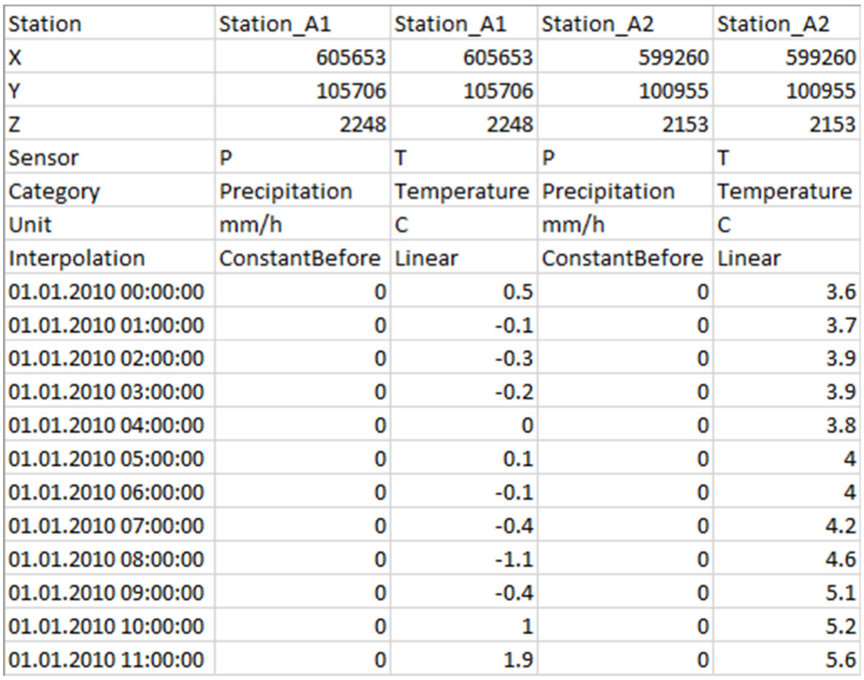
ConstantAfter

ConstantBefore

Linear

Figure 36 Structure of CSV file in RSMINERVE.

And Figure 37 shows an example of CSV file:



Station	Station_A1	Station_A1	Station_A2	Station_A2
X	605653	605653	599260	599260
Y	105706	105706	100955	100955
Z	2248	2248	2153	2153
Sensor	P	T	P	T
Category	Precipitation	Temperature	Precipitation	Temperature
Unit	mm/h	C	mm/h	C
Interpolation	ConstantBefore	Linear	ConstantBefore	Linear
01.01.2010 00:00:00	0	0.5	0	3.6
01.01.2010 01:00:00	0	-0.1	0	3.7
01.01.2010 02:00:00	0	-0.3	0	3.9
01.01.2010 03:00:00	0	-0.2	0	3.9
01.01.2010 04:00:00	0	0	0	3.8
01.01.2010 05:00:00	0	0.1	0	4
01.01.2010 06:00:00	0	-0.1	0	4
01.01.2010 07:00:00	0	-0.4	0	4.2
01.01.2010 08:00:00	0	-1.1	0	4.6
01.01.2010 09:00:00	0	-0.4	0	5.1
01.01.2010 10:00:00	0	1	0	5.2
01.01.2010 11:00:00	0	1.9	0	5.6

Figure 37 Example of CSV file in text format.

8.3. Automatic creation of databases or datasets

An automatic creation of a database or a dataset is possible. Concerning the text file (respectively *.dbt, *.dst or *.csv), it can be easily created based on information provided in Chapter 7.2. Regarding the .xml file (*.dbx or *.dsx) or plain file (*.csv), additional information for the automatic creation is presented hereafter.

Table 29 and Table 30 describe all the possible nodes respectively represented in a database (*.dbx) and a dataset XML file (*.dsx). The presented tables takes into account an example with a dataset “Measure” with a station “Aigle” and a sensor of observed precipitation “P”. The information visible in Database is also presented in last three columns.

To create an XML file, all the possible nodes are very useful but not all required, since some of them are also necessary for operational hydro-meteorological systems.

“Key” node generally allows distinguishing different databases (or datasets) with different name and structure. “ID” node is only useful for operational system, since it can be used when datasets with same “Key” (as well as same filename and structure) are overwritten at each new forecast. Then, “ID” node allows to Database knowing if the dataset has been updated or not.

Table 29 Description of all the possible nodes in a database XML file

Node	Required	Suggested	Example	Purpose	Visible		Non visible
					Editable	Non editable	
DataBase	X		**<Database>**	**Starts the database node**			X
DateCreated		X	<DateCreated>2014-02-05T17:10:10.1625449+01:00</DateCreated>	Database description		X	
Version		X	<Version>4.001</Version>	To verify RS Database version			X
Key		X	<Key>2234d822-7fb7-4d77-afb5-564f029f4c66</Key>	To distinguish different databases			X
ID		X	<ID>d11fa701-0e05-4833-847e-4385a8b808f1<\ID>	To distinguish databases with identical Key			X
LastComputer		X	<LastComputer>D29493</LastComputer>	Computer origin			X
LastUser		X	<LastUser>D29493</LastUser>	User			X
Name	X		<Name>Database</Name>	Database description	X		
LastDateModified		X	<LastDateModified>2014-04-10T15:31:46.2862121+02:00</LastDateModified>	Database description		X	
TimeOffset		X	<TimeOffset>36000000000</TimeOffset>	Database description	X		
Groups	X		**<Groups>**	**Starts the groups node**			X
Group	X		**<Group>**	**Starts the description of a group**			X
Key		X	<Key>7f63d446-5eb7-4d75-8d2a-e84c79a6bf89</Key>	To distinguish groups in the database			X
Name	X		<Name>Measure</Name>	Group description	X		
DataSets	X		**<DataSets>**	**Starts the datasets node**			X
DataSet	X		**<DataSet>**	**Starts the description of a dataset**			X
Version		X	<Version>4.001</Version>	To verify RS Database version			X
Key		X	<Key>a3e0e759-6f10-4a93-8aa1-cb213ac767f1</Key>	To distinguish different datasets			X
ID		X	<ID>a5a3c179-5eb0-47ab-8d9c-f820e741977c</ID>	To distinguish datasets with identical Key			X
LastComputer		X	<LastComputer>vmcrealp <\LastComputer>	Computer origin			X
LastUser		X	<LastUser>crealp</LastUser>	User			X
Name	X		<Name>201404090700_Measure</Name>	Dataset description	X		
DateCreated		X	<DateCreated>2014-04-10T10:00:12+02:00</DateCreated>	Dataset description		X	
DateRecept		X	<DateRecept>2014-04-10T15:29:36.0541902+02:00</DateRecept>	Dataset description		X	
LastDateModified		X	<LastDateModified>2014-04-10T15:29:36.0531901+02:00</LastDateModified>	Dataset description		X	
DateCapture	X		<DateCapture>2014-04-09T07:00:00.0000000+02:00</DateCapture>	Dataset description	X		
Stations	X		**<Stations>**	**Starts the stations node**			X
Station	X		**<Station>**	**Starts the description of a station**			X
Key		X	<Key>AIG</Key>	To distinguish stations in the database			X
Name	X		<Name>Aigle</Name>	Station description	X		
Sensors	X		**<Sensors>**	**Starts the sensors node**			X
Sensor	X		**<Sensor>**	**Starts the description of a sensor**			X
Key		X	<Key>AIG\P</Key>	To distinguish sensors in the database			X
Type		X	<Type>Inputs</Type>	Type of data			X
Name	X		<Name>P</Name>	Sensor description	X		
Category	X		<Category>Precipitation</Category>	Sensor description	X		
Unit	X		<Unit>MillimetersPerHour</Unit>	Sensor description	X		
InterpolationMode	X		<InterpolationMode>ConstantBefore</InterpolationMode>	Sensor description	X		
Sensor	X		**</Sensor>**	Finish the description of a sensor			X
Sensors	X		**</Sensors>**	Finish the sensors node			X
X	X		<X>560401</X>	Station description	X		
Y	X		<Y>130713</Y>	Station description	X		
Z	X		<Z>381</Z>	Station description	X		
Station	X		**</Station>**	Finish the description of a station			X
Stations	X		**</Stations>**	Finish the stations node			X
DataSet	X		**</DataSet>**	Finish the description of a dataset			X
DataSets	X		**</DataSets>**	Finish the datasets node			X
Category		X	<Category>Outputs</Category>	Group description	X		
Group	X		**</Group>**	Finish the description of a group			X
Groups	X		**</Groups>**	Finish the groups node			X
TimeOffset		X	<TimeOffset>36000000000</TimeOffset>	Database description	X		
DataBase	X		**</Database>**	Finish the database node			X

Table 30 Description of all the possible nodes in a dataset XML file

Node	Required	Suggested	Example	Purpose	Visible		Non visible
					Editable	Non editable	
DataSets	X		**<DataSets>**	**Starts the datasets node**			X
DataSet	X		**<DataSet>**	**Starts the description of a dataset**			X
Version		X	<Version>4.001</Version>	To verify RS Database version			X
Key		X	<Key>a3e0e759-6f10-4a93-8aa1-cb213ac767f1</Key>	To distinguish different datasets			X
ID		X	<ID>a5a3c179-5eb0-47ab-8d9c-f820e741977c</ID>	To distinguish datasets with identical Key			X
LastComputer		X	<LastComputer>vmcrealp <\LastComputer>	Computer origin			X
LastUser		X	<LastUser>crealp</LastUser>	User			X
Name	X		<Name>201404090700_Measure</Name>	Dataset description	X		
DateCreated		X	<DateCreated>2014-04-10T10:00:12+02:00</DateCreated>	Dataset description		X	
DateRecept		X	<DateRecept>2014-04-10T15:29:36.0541902+02:00</DateRecept>	Dataset description		X	
LastDateModified		X	<LastDateModified>2014-04-10T15:29:36.0531901+02:00</LastDateModified>	Dataset description		X	
DateCapture	X		<DateCapture>2014-04-09T07:00:00.0000000+02:00</DateCapture>	Dataset description	X		
Stations	X		**<Stations>**	**Starts the stations node**			X
Station	X		**<Station>**	**Starts the description of a station**			X
Key		X	<Key>AIG</Key>	To distinguish stations in the			X
Name	X		<Name>Aigle</Name>	Station description	X		
Sensors	X		**<Sensors>**	**Starts the sensors node**			X
Sensor	X		**<Sensor>**	**Starts the description of a sensor**			X
Key		X	<Key>AIG\P</Key>	To distinguish sensors in the database			X
Type		X	<Type>Inputs</Type>	Type of data			X
Name	X		<Name>P</Name>	Sensor description	X		
Category	X		<Category>Precipitation</Category>	Sensor description	X		
Unit	X		<Unit>MillimetersPerHour</Unit>	Sensor description	X		
InterpolationMode	X		<InterpolationMode>ConstantBefore</InterpolationMode>	Sensor description	X		
Sensor	X		**</Sensor>**	Finish the description of a sensor			X
Sensors	X		**</Sensors>**	Finish the sensors node			X
X	X		<X>560401</X>	Station description	X		
Y	X		<Y>130713</Y>	Station description	X		
Z	X		<Z>381</Z>	Station description	X		
Station	X		**</Station>**	Finish the description of a station			X
Stations	X		**</Stations>**	Finish the stations node			X
DataSet	X		**</DataSet>**	Finish the description of a dataset			X
DataSets	X		**</DataSets>**	Finish the datasets node			X

Chapter 9. GIS formats

RS MINERVE includes a GIS module accepting different formats.

9.1. Model files

Layers with formats presented in Table 31 can be imported in GIS.

Table 31 Formats and extensions supported by GIS

Formats	Extensions
Vector	*.shp
Raster	*.bgd
Image	*.bmp, *.emf, *.exf, *.gif, *.ico, *.jpg, *.png, *.tif, *.wmf
DotSpatial.Shapefile	*.shp
DotSpatial – Binary file	*.bgd
DotNet Image - Image	*.bmp, *.emf, *.exf, *.gif, *.ico, *.jpg, *.png, *.tif, *.wmf
DotNet Image - Bitmap	*.bmp

Bibliography

Abbott, M. B. and Refsgaard, J. C. (1996). *Distributed Hydrological Modelling*. Springer Science & Business Media, 321 pp.

AghaKouchak, A.; Habib, E. (2010). *Application of a Conceptual Hydrologic Model in Teaching Hydrologic Processes*. International Journal of Engineering Education 26(4), 963-973.

Ajami, N. K., Gupta, H., Wagener, T. and Sorooshian, S. (2004). *Calibration of a semi-distributed hydrologic model for streamflow estimation along a river system*. Journal of Hydrology 298, 112-135.

Bergström, S. (1976). Development and application of a conceptual runoff model for Scandinavian catchments. Ph.D. Thesis. SMHI Reports RHO No. 7, Norrköping.

Bergström, S. (1992). *The HBV model - its structure and applications*. SMHI Reports RH, No. 4, Norrköping.

Blasone, R.-S., Madsen, H. and Rosbjerg, D. (2007). Parameter estimation in distributed hydrological modelling: comparison of global and local optimisation techniques. Nordic Hydrology, Vol 38, No. 4-5, 451-476.

Boillat J.-L., (1980) *Polycopié « Hydraulique urbaine »*. Laboratoire de Constructions Hydrauliques, Ecole Polytechnique Fédérale de Lausanne.

Burnash, R. J. C., Ferral, R. L., and McGuire, R. A. (1973). *A generalized streamflow simulation system – Conceptual modelling for digital computers*. US Department of Commerce, National Weather Service and State of California, Department of Water Resources, p 204, 1973.

Burnash, R. J. C. (1995). *The NWS River Forecast System - catchment modeling*. In: Singh, V. P. (Ed.). Computer Models of Watershed Hydrology, 311-366.

Colebrook, C. F. and White, C. M. (1937). *Experiments with Fluid Friction in Roughened Pipes*. Proceedings of the Royal Society of London, Series A, Mathematical, Physical and Engineering Sciences 161, 367-381.

Colebrook, C.F. (1939). Turbulent flow in pipes, with particular reference to the transition region between smooth and rough pipe laws. Journal of the Institution of Civil Engineers 11, 133-156.

Consuegra D., Niggli M. and Musy A. (1998). *Concepts méthodologiques pour le calcul des crues. Application au bassin versant supérieur du Rhône*. Wasser, Energie, Luft – eau, énergie, air, Heft 9/10, 223-231.

Cunge, J.A. (1969). Au sujet d'une méthode de calcul de propagation des crues (Méthode Muskingum). Journal of Hydraulic Research 7, 205-230.

Cunge, J.A. (1991). *Polycopié Simulation des écoulements non-permanents dans les rivières et canaux*. Institut National Polytechnique de Grenoble, Ecole Nationale Supérieure d'Hydraulique et de Mécanique de Grenoble.

N.A.C. Cressie, (1993), *Statistics for Spatial Data*, Wiley

Darcy, H. (1857). Recherches Expérimentales Relatives au Mouvement de l'Eau dans les Tuyaux [Experimental Research on the Movement of Water in Pipes]. Mallet-Bachelier, Paris, 268 p. & atlas.

Duan, Q., Sorooshian, S. and Gupta, V. (1992). *Effective and Efficient Global Optimization for Conceptual Rainfall-Runoff Models*. Water Resources Management, Vol. 28, No. 4, 1015-1031.

Duan, Q., Gupta, V. K. and Sorooshian, S. (1993). *A shuffled complex evolution approach for effective and efficient global minimization*. Journal of Optimization Theory and Applications, Vol. 76, No. 3, 501-521.

Duan, Q., Sorooshian, S. and Gupta, V. K. (1994). *Optimal use of SCE-UA global optimization method for calibrating watershed models*. Journal of Hydrology, Vol. 158, 265-284.

Dubois, J. and Boillat, J.-L. (2000). *Routing System - Modélisation du routage des crues dans des systèmes hydrauliques à surface libre*. Communication 9 du Laboratoire de Constructions Hydrauliques, Ed. A. Schleiss, Lausanne.

El-Nasr, A. A., Arnold, J. G., Feyen, J. and Berlamont, J. (2005). *Modelling the hydrology of a catchment using a distributed and a semi-distributed model*. Hydrological Processes, Vol. 19 (3), 573-587.

Feyen, L., Vázquez, R., Christiaens, K., Sels, O. and Feyen, J. (2000). *Application of a distributed physically-based hydrological model to a medium size catchment*. Hydrology and Earth System Sciences 4 (1), 47-63.

Foehn, A., García Hernández, J., Roquier, B., Fluixá-Sanmartín, J., Paredes Arquiola, J. and De Cesare, G. (2020). RS MINERVE – User manual, v2.15. Ed. CREALP, Switzerland. ISSN 2673-2653.

Gabellani, S., Boni, G., Ferraris, L., von Hardenberg, J. and Provenzale, A. (2007). *Propagation of uncertainty from rainfall to runoff: A case study with a stochastic rainfall generator*. Advances in Water Resources 30, 2061-2071.

García Hernández, J., Jordan, F., Dubois, J. and Boillat, J.-L. (2007). *Routing System II. Flow modelling in hydraulic systems*. Communication 32 du Laboratoire de Constructions Hydrauliques, Ed. A. Schleiss, Lausanne.

García Hernández, J. (2011). Flood management in a complex river basin with a real-time decision support system based on hydrological forecasts. PhD Thesis N°5093, Ecole Polytechnique Fédérale de Lausanne, EPFL, Switzerland.

Gilks, W. R., Richardson, S. and Spiegelhalter, D. J. (1996) *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London, 512p.

Griessinger, N., Seibert, J., Magnusson, J. and Jonas, T. (2016). *Assessing the benefit of snow data assimilation for runoff modeling in Alpine catchments*. Hydrology and Earth System Sciences, 20(9). 3895–3905. ISSN 1607-7938. doi: 10.5194/hess-20-3895-2016.

Gupta, H. V., Kling, H., Yilmaz, K. K. and Martinez, G. F. (2009). *Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling*. Journal of Hydrology Vol. 377, 80-91.

Hamdi, Y., Hingray, B. and Musy, A. (2005*). Un modèle de prévision hydro-météorologique pour les crues du Rhône supérieur en suisse*. Eau, energie, air, 97(11):325–332.

Hapuarachchi H.A.P., LI, Z, Wang, S. (2001). *Application of SCE-UA Method for Calibrating the Xinanjiang Watershed Model*. Journal of Lake Sciences, Vol 12, No. 4, 304-314.

Heppner, C. S., Ran, Q., Vanderkwaak, J. E. and Loague, K (2006). *Adding sediment transport to the integrated hydrology model (InHM): Development and testing*. Advances in Water Resources 29 (6), 930-943.

Jeremiah, E., Sisson, S.A., Sharma, A. & Marshall, L., (2012). *Efficient hydrological model parameter optimization with Sequential Monte Carlo sampling*. Environmental Modelling and Software 38, pp. 283 – 295

Johnson, Norman L.; Kotz, Samuel; Balakrishnan, N. (1994), "14: Lognormal Distributions", Continuous univariate distributions. Vol. 1, Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics (2nd ed.), New York: John Wiley & Sons, ISBN 978-0-471-58495-7

Jordan, F. (2007). Modèle de prévision et de gestion des crues - optimisation des opérations des aménagements hydroélectriques à accumulation pour la réduction des débits de crue. Thesis Report N°3711, Ecole Polytechnique Fédérale de Lausanne, EPFL, Lausanne.

Kagan, A.M. and Shalaevski,O. (1967). Characterization of normal law by a property of the non-central chi-square distribution, Lithuanian Journal of Math., 7.

Kamali, M., Ponnambalam, K. and Soulis, E D. (2013). *Comparison of several heuristic approaches to calibration of WATCLASS hydrologic model*. Canadian Water Resources Journal, Vol. 38, No. 1, 40–46,

Kang, F., Li, J. and Ma, Z. (2011). Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions. Information Sciences, 181(16), 3508-3531.

Kling, H., Fuchs, M. and Paulin, M. (2012). *Runoff conditions in the upper Danube basin under an ensemble of climate change scenarios*. Journal of Hydrology, Vol 424-425, 264-277.

Krause, P., Boyle, D. P. and Bäse, F. (2005). *Comparison of different efficiency criteria for hydrological model assessment*. Advances in Geosciences 5, 89-97.

Lin, J.-Y., Cheng, C.-T. and Lin, T. (2008). *A Pareto Strength SCE-UA Algorithm for Reservoir Optimization Operation*. Proceedings of the Fourth International Conference on Natural Computation, 406-412, ISBN: 978-0-7695-3304-9.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientic Computing*. Springer Verlag, New-York, 343p.

Magnusson, J., Gustafsson, D., Hüsler, F. and Jonas, T. (2014*). Assimilation of point SWE data into a distributed snow cover model comparing two contrasting methods*. Water Resources Research. October 2014. Vol. 50, no. 10, p. 7816–7835. DOI 10.1002/2014WR015302.

Manzato, A., 2007. A Note On the Maximum Peirce Skill Score. Weather and Forecasting 22, 1148–1154. https://doi.org/10.1175/WAF1041.1

Mariani, V. C., Justi Luvizotto, L. G., Guerra, F. A. and Coelho., L. d. S. (2011). *A hybrid shuffled complex evolution approach based on differential evolution for unconstrained optimization*. Applied Mathematics and Computation, In Press, Corrected Proof, Available online 21 December 2010, ISSN 0096-3003, DOI: 10.1016/j.amc.2010.12.064.

Masmoudi, M. and Habaieb, H. (1993). The performance of some real-time statistical flood forecasting models seen through multicriteria analysis. Water Resources Management 7 (1), 57-67.

McGuinness, J.L. and Bordne, E.F. (1972). *A comparison of lysimeter-derived potential evapotranspiration with computed values*. Technical Bulletin 1452, Agricultural Research Service, U.S. Department of Agriculture, Washington D.C., 71 pp.

McKay, M.D., Beckman, R.J. and Conover, W.J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21 (2), 239–245.

Metcalf and Eddy, Inc., University of Florida, and Water Resources Engineers, Inc. (1971). *Storm Water Management Model, Vol. I*. Final Report, 11024DOC07/71 (NTIS PB-203289), U.S. EPA, Washington, DC, 20460.

Michel, C. and Edijatno (1988). Réflexion sur la mise au point d'un modèle pluie-débit simplifié sur plusieurs bassins versants représentatifs et expérimentaux. CEMAGREF Antony.

Moriasi, D.N., Arnold, J.G., Van Liew, M. W., Bingner, R.L., Harmel, R.D., Veith, T.L. (2007). *Model evalutation guidelines for systematic quantification of accuracy in watershed simulations*. American Society of Agricultural and Biological Engineers 50(3), 885-900.

Muttil, N. and Liong, S.-Y. (2004). *Superior exploration-exploitation balance in shuffled complex evolution*. Journal of Hydraulic Engineering, Vol. 130, No. 12, 1202-1205.

Nash, J. E. and Sutcliffe, J. V. (1970). River flow forecasting through conceptual models: part 1 - A discussion of principles. Journal of Hydrology 10, 282-290.

Nelder, J. A. and Mead, R. (1965). *A Simplex Method for function minimization*. Computer Journal, Vol. 7, No. 4, 308-313.

Nóbrega, M. T., Collischonn, W., Tucci, C.E.M., Paz, A.R. (2011). *Uncertainty in climate change impacts on water resources in the Rio Grande Basin, Brazil*. Hydrology and Earth System Sciences 15, 585–595.

Oudin, L. (2004). Recherche d'un modèle d'évapotranspiration potentielle pertinent comme entrée d'un modèle pluie-débit global. Thèse, Ecole Nationale du Génie Rural, des Eaux et des Forêts, Paris.

Parajka, J., Blöschl, G., 2008. Spatio-temporal combination of MODIS images - potential for snow cover mapping: SPATIO-TEMPORAL COMBINATION OF MODIS IMAGES. Water Resources Research 44.

Park, Sung Y. ;Bera, Anil K. (2009) "Maximum entropy autoregressive conditional heteroskedasticity model". Journal of Econometrics (Elsevier) 150 (2): 219–230.

Peirce, C.S. (1884). *The numerical measure of the success of predictions*. Science ns-4, 453–454.

Perrin, C., Michel, C. and Andréassian, V. (2003). *Improvement of a parsimonious model for streamflow simulation*. Journal of Hydrology 279, 275-289.

Prakasa Rao, B.L.S. (1992). Identiability in Stochastic Models: Characterization of Probability Distributions, Academic Press, Boston.

Rosenbrock, H.H. (1960). An automatic method for finding the greatest or least value of a function. The Computer Journal 3, 175–184.

Schaefli, B., Hingray, B., Niggli, M. and Musy, A. (2005). *A conceptual glacio-hydrological model for high mountainous catchments*. Hydrology and Earth System Sciences Discussions 2, 73-117.

Seibert, J. (1997). *Estimation of parameter uncertainty in the HBV model*. Nordic Hydrology 28(4/5), 247-262.

Shepard, D. (1968). *A two-dimensional interpolation function for irregularly-spaced data*. Proceedings 23rd National Conference ACM, 517–524.

Simmons, C. T. (2008). Henry Darcy (1803–1858): Immortalised by his scientific legacy. Hydrogeology Journal 16, 1023-1038.

Slater, A. G. and Clark, M.P. (2006). *Snow Data Assimilation via an Ensemble Kalman Filter*. Journal of Hydrometeorology. June 2006. Vol. 7, no. 3, p. 478–493. DOI 10.1175/JHM505.1.

Stoyan, D., Kendall, W. S. and Mecke, J. (1995). Stochastic geometry and its applications, volume 2. Wiley, 1995.

Sun, X., Mein, R.G., Keenan, T.D. and Elliott, J.F. (2000). *Flood estimation using radar and raingauge data*. Journal of Hydrology 239, 4-18.

Turc, L. (1955). Le bilan de l'eau des sols. Relations entre les precipitations, l'evaporation et l'ecoulement. Ann. Agro. 6, 5-152, INRA.

Turc, L. (1961). Evaluation des besoins en eau d’irrigation, formule climatique simplifiée et mise à jour. Ann. Agro. 12: 13-49, INRA.

van Griensven, A., Meixner, T., Grunwald, S., Bishop, T., Diluzio, A. and Srinivasan, R. (2006). *A global sensitivity analysis tool for the parameters of multi-variable catchment models*. Journal of Hydrology, 324(1-4), 10-23.

Viviroli, D., Mittelbach, H., Gurtz, J., Weingartner, R. (2009). Continuous simulation for flood estimation in ungauged mesoscale catchments of Switzerland – Part II: Parameter regionalisation and flood estimation results. Journal of Hydrology 377, 208-225.

Vrugt, J.A., Gupta, H.V., Bastidas, L.A., Bouten, W. and Sorooshian, S. (2003). *Effective and efficient algorithm for multiobjective optimization of hydrologic models*. Water resources research 29, 1214.

Wang, L., Nyunt, C. T., Koike, T., Saavedra, O., Nguyen, L. C. and Sap, T. V. (2010). *Development of an integrated modeling system for improved multi-objective reservoir operation*. Frontiers of Architecture and Civil Engineering in China, Vol. 4, No. 1, 47-55.

Wang, Q.J., Pagano, T.C., Zhou, S.L., Hapuarachchi, H.A.P., Zhang, L. and Robertson, D.E. (2011). *Monthly versus daily water balance models in simulating monthly runoff.* Journal of Hydrology 404, 166-175.

Zhu, X., Wu, J. and Wu, J. (2006). *Application of SCE-UA to Optimize the Management Model of Groundwater Resources in Deep Aquifers of the Yangtze Delta*. Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences, Hangzhou, Zhejiang, China, 303-308, ISBN: 0-7695-2581-4.

Acknowledgments

RS MINERVE is developed by the research center **CREALP** and the engineering office **HydroCosmos SA**, with the collaboration of two universities (**Ecole Polytechnique Fédérale de Lausanne** and **Universitat Politècnica de València**) and the **Hydro10** Association.

We would like to thank sincerely these organizations for their support.

In addition, we would like to express our gratitude to the people who have personally contributed to the improvement of the program and its documentation, in particular: Stéphane Micheloud[1], Néstor Lerma[2], Alex Dionisio-Calado[1], Nicolas Rey[3], Edgar Belda[2], Samuel Alesina[1], Aurélien Claude[1] and Javier Fluixá Sanmartín[1].

[1] Centre de recherche sur l'environnement alpin - CREALP

[2] Université Polytechnique de Valence - UPV

[3] HydroCosmos SA

Appendix: List of notifications from pre-simulation and post-simulation reports

The list of notifications (notes, warnings and fatal) of the pre- and post-simulation reports are presented hereafter.

A.1. Isolated objects

Pre-simulation report

Warning	If an object is isolated and is not taken into account for calculation.	**The object 'Object 1' has no links and is isolated.**

A.2. Hydrology

Virtual Station

Pre-Simulation Report

Fatal	If a Virtual Station exists and is not isolated, but no database is connected to the model.	**Missing database to get station data.**
Fatal	If this object exists and a database is connected to the hydrological model, but any dataset is selected for meteorological data utilisation.	**No dataset is selected for station 'Station 1' (X m, Y m, Z masl).**
Fatal	If this object exists and a database is connected to the hydrological model, but no datagroup is selected in the Data source.	**No datagroup is selected for station 'Station 1' (X m, Y m, Z masl).**
Fatal	If this object exists and a database is connected to the hydrological model, but a precipitation input is not available in the database.	**No precipitation values for station 'Station 1' (X m, Y m, Z masl).**
Warning	If this object exists and a database is connected to the hydrological model, but a temperature / ETP input is not available.	**No temperature/ETP values for station 'Station 1' (X m, Y m, Z masl). A null series will be considered.**

Post-Simulation Report

Warning	If this object does not find data for Precipitation, Temperature or ETP in any station of the database within in the radius of research, research radius is extended until finding one.	**No precipitation found within search radius of virtual station 'Station 1'. Radius extended to satisfy No. min of stations.**

Snow-SD model description

Pre-simulation report

Fatal	If this object exists and is connected to other objects, but not all Snow-SD model inputs are available.	**The object 'Snow-SD 1' has no Precipitation 'i' / Temperature 'T' as input.**

Runoff (SWMM) model description

Pre-simulation report

Fatal	If the object exists and is not isolated, but has no intensity as input.	**The object 'SWMM 1' has no Precipitation 'i' as input.**

GSM model description

Pre-simulation report

Fatal	If this object exists and is not isolated, but not all GSM model inputs are available.	**The object 'GSM 1' has no Precipitation 'P' / Temperature 'T' as input.**

SOCONT model description

Pre-simulation report

Fatal	If this object exists and is not isolated, but not all SOCONT model inputs are available.	**The object 'SOCONT 1' has no Precipitation 'P' / Temperature 'T'/ETP 'ETP' as input.**

HBV model description

Pre-simulation report

Fatal	If this object exists and is not isolated, but not all HBV model inputs are available.	**The object 'HBV 1' has no Precipitation 'P'/Temperature 'T'/ETP 'ETP' as input.**

GR4J model description

Pre-simulation report

Fatal	If this object exists and is not isolated, but not all HBV model inputs are available.	**The object 'GR4J' has no Precipitation 'P'/ETP 'ETP' as input.**

SAC model description

Pre-simulation report

Fatal	If this object exists and is not isolated, but not all SAC model inputs are available.	**The object 'SAC' has no Precipitation 'P'/ETP 'ETP' as input.**

A.3. Rivers

Channel routing description

No fatal, warnings or notes messages for this object.

A.4. Standard objects

Junction

No fatal, warnings or notes messages for this object.

Time Series

Pre-simulation report

Fatal	If this object if used and is not isolated, but the H-Q paired data is not provided.	**Times Series 'Time series 1'is missing paired data values.**

Source

Pre-simulation report

Warning	If a source object exists and no database is connected to the model.	**Missing database to get source data.**
Warning	If no group is selected in the Data source for sources.	**Missing database group to source data.**
Warning	If this object is used and is not isolated, but the source has no a station from database to provide data series.	**No station or sensor for source 'Source 1'.**

Comparator

No fatal errors, warnings or note messages for this object

Sub-model

No fatal errors, warnings or note messages for this object

Group Interface

No fatal errors, warnings or note messages for this object

A.5. Structures objects

Reservoir

Pre-simulation report

Fatal	If this object is used and is not isolated, but the H-V paired data is not provided.	**Reservoir 'Reservoir 1' is missing H-V paired data.**
Fatal	If this object is used and is not isolated, but does not have an inflow.	**The object 'Reservoir 1' has no Flow 'Qe' as input.**
Warning	If the initial level of the reservoir is out of the range proposed in the H-V paired data. The simulation is achieved, assuming the level equals to the maximum level of the reservoir.	**Hini of reservoir 'Reservoir 1'is out of range.**
Note	If a Reservoir object exists but no database is connected to the model.	**Missing database to get initial height of reservoirs.**
Note	If no group is selected in the Data source for reservoirs.	**Missing database group to get HPP data.**
Note	If the reservoir uses the initial condition proposed by the user (and not data from database as initial condition).	**Hini of reservoir 'Reservoir 1'used as initial height.**
Note	If the reservoir uses the initial condition from the database.	**Initial height of reservoir 'Reservoir 1' taken from database.**

Post-Simulation Report

Warning	If the volume becomes negative in a moment of the simulation. Nevertheless, the simulation is computed (with a correct volume variation, V; keeping the reservoir level, H, in zero for negative volumes).	**Volume of reservoir 'Reservoir 1' contains negative values.**
Warning	If the volume exceeds the maximum volume proposed in the relation H-V. Nevertheless, the simulation is computed (with a correct volume variation, V, keeping the reservoir level, H, to maximum value of paired data H-V when volume exceeds the maximum volume).	**Volume of reservoir 'Reservoir 1' exceeds its capacity.**

HQ

Pre-simulation report

Fatal	If this object is used and is not isolated, but the H-Q paired data is not provided.	**HQ 'HQ 1'is missing H-Q paired data.**

Post-Simulation Report

Warning	The computed simulation runs even if the level is equal or over the maximum level proposed in the relation H-Q. In that case, the outflow is kept as the maximum discharge existing in the relation H-Q.	**The maximum level of HQ 'HQ 1' has been attained or exceeded.**

Turbine

Pre-simulation report

Fatal	If this object is used and is not isolated, but do not receive the Q_{wanted} data as input.	**The object 'Turbine 1' has no Flow 'Qwanted' as input.**

TurbineDB

Pre-simulation report

Warning	If a TurbineDB object exists and no database is connected to the model.	**Missing database to get TurbineDB data.**
Warning	If this object if used and is not isolated, but no group is selected in the Data source for TurbineDB.	**Missing database group to get TurbineDB data.**
Warning	If this object if used and is not isolated, but the TurbineDB has not a station from database to provide data series.	**No station or sensor for TurbineDB 'TurbineDB 1'.**

Hydropower

Pre-simulation report

Fatal	If this object if used and is not isolated, but the Q-η paired data is not provided.	**Hydropower 'Hydropower 1'is missing paired data values.**
Warning	If a source object exists and no database is connected to the model.	**Missing database to get Hydropower data. Default Price value(s) will be used.**
Warning	If no group is selected in the Data source for sources.	**Missing database group to get Hydropower data. Default Price value(s) will be used.**
Warning	If this object is used and is not isolated, but the consumer has not a station from database to provide data series.	**No station or sensor for Hydropower 'Hydropower 1'. Price defined from Default Price.**

Diversion

Pre-simulation report

Fatal	If this object if used and is not isolated, but the Q_{Up}-$Q_{Diverted}$ paired data is not provided.	**Diversion 'Diversion 1'is missing QUp-QDiverted paired data.**

Consumer

Pre-simulation report

Warning	If a source object exists and no database is connected to the model.	**Missing database to get Consumer data. Default QDemand values will be used.**
Warning	If no group is selected in the Data source for sources.	**Missing database group to consumer data. Default QDemand values will be used.**
Warning	If this object is used and is not isolated, but the consumer has not a station from database to provide data series.	**No station or sensor for Consumer 'Consumer 1'. Consumption defined from Default QDemand.**

A.6. Planner object

Planner

Pre-simulation report

Fatal Error	A rule in required.	**The planner 'Planner 1' has no rules defined.**

Rules

Pre-simulation report

Fatal Error	A condition is required.	**The rule 'Rule 1' in planer 'Planner 1' has no condition defined.**

Conditions

Pre-simulation report

Fatal Error	An input is required.	**The condition 'Condition 1' in 'Rule 1' of ' Planner 10 has no input defined.**
Fatal Error	An object is required.	**The condition 'Condition 1' in 'Rule 1' of ' Planner 10 has no object defined.**
Fatal Error	A result is required.	**The condition 'Condition 1' in 'Rule 1' of ' Planner 10 has no result defined.**

RS MINERVE

CREALP - Centre de recherche sur l'environnement alpin

Rue de l'Industrie 45
CH-1951 Sion

Tél. +41 (0) 27 607 11 80
Fax +41 (0) 27 607 11 94
crealp@crealp.vs.ch