Authors:

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

Please refer this technical manual as:

4.1. Introduction to the calibration module ................................................................. 64
4.2. Objective function ............................................................................................... 64
4.3. Shuffled Complex Evolution – University of Arizona ........................................... 65
4.4. Uniform Adaptive Monte Carlo .......................................................................... 71
4.5. Coupled Latin Hypercube and Rosenbrock......................................................... 73
Chapter 5. Expert module - Stochastic equations ........................................................ 77
5.1. Introduction to the stochastic simulation ............................................................. 77
5.2. Probability distributions ...................................................................................... 77
Chapter 6. Visual Basic Scripts ................................................................................ 82
6.1. Introduction to Visual Basic Scripts ..................................................................... 82
6.2. Available VBS procedures and functions in RS MINERVE ................................. 83
6.3. Application Start ................................................................................................. 84
6.4. Model preparation ............................................................................................... 86
6.5. Simulation and reports ....................................................................................... 101
6.6. Save model and properties ................................................................................ 104
6.7. Model results ..................................................................................................... 109
6.8. Calibration ........................................................................................................ 119
6.9. Application Stop ................................................................................................ 123
6.10. Other procedures/functions ............................................................................ 124
6.11. Objects nomenclature in VBScript ................................................................. 125
Chapter 7. File formats ............................................................................................. 126
7.1. Model files ........................................................................................................ 126
7.2. Parameters and initial conditions files ............................................................... 126
7.3. Object selection files ....................................................................................... 126
7.4. Exported results files ...................................................................................... 127
7.5. Calibration configuration files .......................................................................... 127
7.6. Database files .................................................................................................. 127
Chapter 8. Database formats .................................................................................. 128
8.1. The Database Viewer ....................................................................................... 128
8.2. Database, dataset and CSV formats ................................................................. 130
8.3. Automatic creation of databases or datasets ..................................................... 138
Chapter 9. GIS formats ............................................................................................ 141
9.1. Model files ...................................................................................................... 141
Bibliography .............................................................................................................. 142
Acknowledgments ....................................................................................................... 147
Appendix: List of notifications from pre-simulation and post-simulation reports ........ 1
A.1. Isolated objects .................................................................................................. 1
A.2. Hydrology ......................................................................................................... 1
A.3. Rivers ............................................................................................................... 3
A.4. Standard objects ............................................................................................. 4
A.5. Structures objects ........................................................................................... 5
A.6. Planner object
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, GRAJ 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 École 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 École 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

<table>
<thead>
<tr>
<th>Object</th>
<th>Name</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X, Y, Z</td>
<td>Coordinates of the virtual station</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search Radius</td>
<td>m</td>
<td></td>
<td>Search radius of the virtual stations</td>
</tr>
<tr>
<td>No. min. of stations</td>
<td></td>
<td></td>
<td>Minimal number of stations used for interpolation (higher priority than “Search Radius”)</td>
</tr>
<tr>
<td>Gradient P</td>
<td>1/m</td>
<td></td>
<td>Precipitation gradient</td>
</tr>
<tr>
<td>Gradient T</td>
<td>°C/m</td>
<td></td>
<td>Temperature gradient</td>
</tr>
<tr>
<td>Gradient ETP</td>
<td>1/m</td>
<td></td>
<td>Evapotranspiration gradient</td>
</tr>
<tr>
<td>Coeff P</td>
<td></td>
<td></td>
<td>Multiplying correction coefficient</td>
</tr>
<tr>
<td>Coeff T</td>
<td>°C</td>
<td></td>
<td>Adding correction coefficient</td>
</tr>
<tr>
<td>Coeff ETP</td>
<td></td>
<td></td>
<td>Multiplying correction coefficient</td>
</tr>
</tbody>
</table>

| Station               |            |       |                                                                          |
| Gradient P            | 1/m        |       |                                                                          |
| Gradient T            | °C/m       |       |                                                                          |
| Gradient ETP          | 1/m        |       |                                                                          |
| Coeff P               |            |       |                                                                          |
| Coeff T               | °C         |       |                                                                          |
| Coeff ETP             |            |       |                                                                          |

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 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.

\[
P_s = \text{Coeff } P_s \cdot \left(1 + \text{Grad } P_s \cdot (z_s - z_k)\right) \cdot P_k
\]

\[
T_s = \text{Coeff } T_s + \text{Grad } T_s \cdot (z_s - z_k) + T_k
\]

\[
ETP_s = \text{Coeff } ETP_s \cdot \left(1 + \text{Grad } ETP_s \cdot (z_s - z_k)\right) \cdot ETP_k
\]

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 k [I.U.]; \( ETP_k \): value of the potential evapotranspiration in the meteorological station k [I.U.]; \( \text{Coeff } P_s \): precipitation coefficient [-]; \( \text{Coeff } T_s \): temperature coefficient [°C]; \( \text{Coeff } ETP_s \):
potential evapotranspiration coefficient [-]; GradP: precipitation gradient \( (dP/dz) \) [1/m]; GradT: temperature gradient \( (dT/dz) \) [°C/m]; GradETP: potential evapotranspiration gradient \( (dETP/dz) \) [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 \( r_s \) (search radius) and \( \text{No. min. of stations} \) (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 \( s \).

\[
d_{i,s} = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2}
\]

(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 with the corresponding parameter.

\[
d_{i,s} \leq r_s
\]

(A.5)

*with* \( r_s \): 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.

\[
P_s = \text{Coeff}_{P_s} \cdot \left( \frac{\sum_{i=1}^{n} \left( (1 + \text{GradP}_s \cdot (z_s - z_i)) \cdot P_i \cdot \frac{1}{d_{i,s}^2} \right)}{\sum_{i=1}^{n} \frac{1}{d_{i,s}^2}} \right)
\]

(A.6)

\[
T_s = \text{Coeff}_{T_s} + \left( \frac{\sum_{i=1}^{n} \left( (\text{GradT}_s \cdot (z_s - z_i) + T_i) \cdot \frac{1}{d_{i,s}^2} \right)}{\sum_{i=1}^{n} \frac{1}{d_{i,s}^2}} \right)
\]

(A.7)

\[
ETP_s = \text{Coeff}_{ETP_s} \cdot \left( \frac{\sum_{i=1}^{n} \left( (1 + \text{GradETP}_s \cdot (z_s - z_i)) \cdot ETP_i \cdot \frac{1}{d_{i,s}^2} \right)}{\sum_{i=1}^{n} \frac{1}{d_{i,s}^2}} \right)
\]

(A.8)

*with* \( P_i \): 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.]; \( P_s \): 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.].
station i [I.U.]; CoeffP: precipitation coefficient [-]; CoeffT: temperature coefficient [°C]; CoeffETP: potential evapotranspiration coefficient [-]; GradP: precipitation gradient (\(dP/dz\)) [1/m]; GradT: temperature gradient (\(dT/dz\)) [°C/m]; GradETP: potential evapotranspiration gradient (\(dETP/dz\)) [1/m]; zs: altitude of the virtual station s [m a.s.l.]; zi: 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).

![Figure 1: Selection of the ETP calculation](image)

The available methods are presented hereafter in detail.

**a) Turc**

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

\[
ETP = \text{CoeffETP} \cdot \frac{T}{T+15} \cdot (R_g+50) \quad \text{if } T > 0 \\
ETP = 0 \quad \text{if } T \leq 0
\]  

(A.9)

with \(ETP\): potential evapotranspiration [mm/month]; \(T\): air temperature [°C]; \(R_g\): global radiation [cal/cm²/day]; \(K\): constant [-].

The constant \(K\) value is:

\[
K = 0.4 \quad \text{if } Month \neq \text{February} \\
K = 0.37 \quad \text{if } Month = \text{February}
\]  

(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/m²/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](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:

$$
ETP = \text{Coeff}_{ETP} \cdot \frac{R_g}{\lambda + \rho} \cdot \frac{T_a + 5}{68} \quad \text{if } T > -5
$$

$$
ETP = 0 \quad \text{if } T \leq -5
$$

(A.11)

with $ETP$: potential evapotranspiration [m/d]; $R_g$: global radiation [MJ/m$^2$/day]; $T_a$: air temperature [°C]; $\rho$: water density, constant value of 1'000 [kg/m$^3$]; $\lambda$: 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](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:

\[
ETP = \text{CoeffETP} \cdot \frac{R_e \cdot T + 5}{\lambda \cdot \rho} \cdot \frac{g}{100} \quad \text{if } T > -5
\]

\[
ETP = 0 \quad \text{if } T \leq -5
\]

(A.12)

with ETP: potential evapotranspiration [m/d]; \(R_e\): extra-terrestrial radiation [MJ m\(^{-2}\) d\(^{-1}\)]; \(T\): Air temperature [°C]; \(\rho\): water density, constant value of 1’000 [kg/m\(^3\)]; \(\lambda\): 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:

\[
R_e = 37.6 \cdot dr \cdot (\omega \cdot \sin(\psi) \cdot \sin(\delta) + \sin(\omega) \cdot \cos(\psi) \cdot \cos(\delta))
\]

(A.13)

\[
dr = 1 + 0.033 \cdot \cos\left(\frac{2 \cdot \pi \cdot J_d}{365}\right)
\]

(A.14)

\[
\omega = \arccos(-\tan(\psi) \cdot \tan(\delta))
\]

(A.15)

\[
\delta = 0.409 \cdot \sin\left(\frac{2 \cdot \pi \cdot J_d}{365} - 1.39\right)
\]

(A.16)

\[
J_d = 275 \cdot \frac{\text{month}}{9} - 30 + \text{Dm} \quad \text{if } \text{month} < 3
\]

\[
J_d = 275 \cdot \frac{\text{month}}{9} - 31 + \text{Dm} \quad \text{if } \text{month} \geq 3 \text{ and leap year} = \text{true}
\]

(A.17)

\[
J_d = 275 \cdot \frac{\text{month}}{9} - 32 + \text{Dm} \quad \text{if } \text{month} \geq 3 \text{ and leap year} = \text{false}
\]

with \(dr\): relative distance Sun-Earth [-]; \(\delta\): solar declination [rad]; \(J_d\): Julian day [-]; \(\psi\): latitude, negative in the south hemisphere [rad]; \(\omega\): hour angle of the sun [rad]; \(\text{month}\): month of the year, 1 to 12 [-]; \(\text{Dm}\):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.

\[
ETP = \text{CoeffETP} \cdot X
\]

(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](image)

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

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

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):

\[
P_w = \alpha \cdot P
\] (B.1)
\[ P_{sn} = (1 - \alpha) \cdot P \]  
\[ \alpha = 0 \quad \text{if} \quad T \leq T_{cp1} \]  
\[ \alpha = (T - T_{cp1})/(T_{cp2} - T_{cp1}) \quad \text{if} \quad T_{cp1} < T < T_{cp2} \]  
\[ \alpha = 1 \quad \text{if} \quad T \geq T_{cp2} \]  

with \( P_w \): liquid precipitation [L/T]; \( \alpha \): separation factor; \( P \): precipitation [L/T]; \( P_{sn} \): solid precipitation [L/T]; \( T \): temperature [°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):

\[ S' = \max \left( S_{Min} ; S + \frac{S_{Int}}{2} \cdot \sin \left( 2\pi \frac{n - S_{Ph}}{365} \right) \right) \]  
\[ M_{sn} = S' \cdot \left( 1 + b_p \cdot P_w \right) \cdot (T - T_{cf}) \quad \text{if} \quad T > T_{cf} \]  
\[ M_{sn} = S' \cdot CFR \cdot (T - T_{cf}) \quad \text{if} \quad T \leq T_{cf} \]  
\[ \frac{dH_{snow}}{dt} = P_{sn} - M_{sn} \]  
\[ M_{sn} \leq P_{sn} + \frac{H_{snow}}{dt} \]  
\[ M_{sn} \geq -W_{snow}/dt \]

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]; \( d_t \): 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):

\[
\begin{align*}
\theta &= \frac{W_{snow}}{H_{snow}} \\
\frac{P_{eq}}{dt} &= P_w + \frac{W_{snow}}{dt} \quad \text{if } H_{snow} = 0 \\
\frac{P_{eq}}{dt} &= 0 \quad \text{if } H_{snow} > 0 \text{ et } \theta \leq \theta_{cr} \\
\frac{P_{eq}}{dt} &= (\theta - \theta_{cr}) \cdot \frac{H_{snow}}{dt} \quad \text{if } H_{snow} > 0 \text{ et } \theta > \theta_{cr} \\
\frac{dW_{snow}}{dt} &= P_w + M_{sn} - P_{eq}
\end{align*}
\] (B.7)

with \( \theta \) (Theta in RS MINERVE): relative water content in the snow pack [-]; \( \theta_{cr} \) (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):

\[
SWE = H_{snow} + W_{snow}
\] (B.10)

with SWE: snow water equivalent [L].

The variables for the initial situation associated to this model are \( \theta \) (Theta in RS MINERVE) and SWE. The parameters to adjust are \( S, S_{int} \) and CFR. The parameters \( S_{phi}, b_p, \theta_{cr}, T_{cp1}, T_{cp2} \) and \( T_{cf} \) can be assumed as constant (\( b_p = 0.0125, \theta_{cr} = 0.1, T_{cp1} = 0 \text{ °C}, T_{cp2} = 4 \text{ °C}, T_{cf} = 0 \text{ °C}, S_{phi} = 80 \) (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).

![SWMM Runoff model](image)

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

<table>
<thead>
<tr>
<th>Object</th>
<th>Name</th>
<th>Units</th>
<th>Description</th>
<th>Regular Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>m²</td>
<td>Surface of runoff</td>
<td>&gt;0</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>m</td>
<td>Length of the plane</td>
<td>&gt;0</td>
</tr>
<tr>
<td>J₀</td>
<td>J₀</td>
<td>-</td>
<td>Runoff slope</td>
<td>&gt;0</td>
</tr>
<tr>
<td>K</td>
<td>K</td>
<td>m^{1/3}/s</td>
<td>Strickler coefficient</td>
<td>0.1 to 90</td>
</tr>
<tr>
<td>H_{ini}</td>
<td>H_{ini}</td>
<td>m</td>
<td>Initial water level downstream of the surface</td>
<td>-</td>
</tr>
</tbody>
</table>

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:

\[
dH/dt = 2 \cdot (i_{net} - i_r)
\]

\[
H_r \geq 0
\]  

\[
i_r = K \cdot \sqrt{J_0} \cdot H^{5/3} \cdot \frac{1}{L}
\] \hspace{1cm} (E.2)

\[
Q = i_r \cdot A
\] \hspace{1cm} (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]; \(J_0\): average slope of the plane [-]; L: length of the plane [L]; A: run-off surface [L²].

The variable for the initial condition associated to the model is \(H_r\). The parameter to adjust is \(K\). The other parameters (\(J_0\), 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](image-url)
Table 4 List of parameters and initial conditions for the GSM model

<table>
<thead>
<tr>
<th>Object</th>
<th>Name</th>
<th>Units</th>
<th>Description</th>
<th>Regular Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>m²</td>
<td></td>
<td>Surface of infiltration</td>
<td>&gt;0</td>
</tr>
<tr>
<td>S</td>
<td>mm/°C/d</td>
<td></td>
<td>Reference degree-day snowmelt coefficient</td>
<td>0.5 to 20</td>
</tr>
<tr>
<td>SInt</td>
<td>mm/°C/d</td>
<td></td>
<td>Degree-day snowmelt interval</td>
<td>0 to 4</td>
</tr>
<tr>
<td>SMin</td>
<td>mm/°C/d</td>
<td></td>
<td>Minimal degree-day snowmelt coefficient</td>
<td>≥0</td>
</tr>
<tr>
<td>Sph</td>
<td>d</td>
<td></td>
<td>Phase shift of the sinusoidal function</td>
<td>1 to 365</td>
</tr>
<tr>
<td>ThetaCri</td>
<td>-</td>
<td></td>
<td>Critical relative water content of the snow pack</td>
<td>0.1</td>
</tr>
<tr>
<td>bp</td>
<td>d/mm</td>
<td></td>
<td>Melt coefficient due to liquid precipitation</td>
<td>0.0125</td>
</tr>
<tr>
<td>Tcp1</td>
<td>°C</td>
<td></td>
<td>Minimum critical temperature for liquid precipitation</td>
<td>0</td>
</tr>
<tr>
<td>Tcp2</td>
<td>°C</td>
<td></td>
<td>Maximum critical temperature for solid precipitation</td>
<td>4</td>
</tr>
<tr>
<td>Tcf</td>
<td>°C</td>
<td></td>
<td>Critical snowmelt temperature</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>mm/°C/d</td>
<td></td>
<td>Reference degree-day glacier melt coefficient</td>
<td>0.5 to 20</td>
</tr>
<tr>
<td>GInt</td>
<td>mm/°C/d</td>
<td></td>
<td>Degree-day glacier melt interval</td>
<td>0 to 4</td>
</tr>
<tr>
<td>GMin</td>
<td>mm/°C/d</td>
<td></td>
<td>Minimal degree-day glacier melt coefficient</td>
<td>≥0</td>
</tr>
<tr>
<td>Tcg</td>
<td>°C</td>
<td></td>
<td>Critical glacier melt temperature</td>
<td>0</td>
</tr>
<tr>
<td>Kgl</td>
<td>1/d</td>
<td></td>
<td>Release coefficient of glacier melt reservoir</td>
<td>0.1 to 5</td>
</tr>
<tr>
<td>Ksn</td>
<td>1/d</td>
<td></td>
<td>Release coefficient of snowmelt reservoir</td>
<td>0.1 to 5</td>
</tr>
<tr>
<td>CFR</td>
<td>-</td>
<td></td>
<td>Refreezing coefficient</td>
<td>0 to 1</td>
</tr>
<tr>
<td>SWEIni</td>
<td>m</td>
<td></td>
<td>Initial snow water equivalent height</td>
<td>-</td>
</tr>
<tr>
<td>ThetalIni</td>
<td>-</td>
<td></td>
<td>Initial relative water content in the snow pack</td>
<td>-</td>
</tr>
<tr>
<td>QsnowIni</td>
<td>m³/s</td>
<td></td>
<td>Initial outflow of linear snow reservoir</td>
<td>-</td>
</tr>
<tr>
<td>QglacierIni</td>
<td>m³/s</td>
<td></td>
<td>Initial outflow of linear glacier reservoir</td>
<td>-</td>
</tr>
</tbody>
</table>

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):

\[
R_w = \alpha \cdot P \quad \text{(F.1)}
\]

\[
P_{sn} = (1 - \alpha) \cdot P \quad \text{(F.2)}
\]

\[
\alpha = 0 \quad \text{if} \quad T \leq T_{cp1}
\]

\[
\alpha = \frac{(T - T_{cp1})/(T_{cp2} - T_{cp1})}{ \text{if} \quad T_{cp1} < T < T_{cp2}}
\]

\[
\alpha = 1 \quad \text{if} \quad T \geq T_{cp2}
\]

with \( P_w \): liquid precipitation \([L/T]\); \( \alpha \): separation factor; \( P \): precipitation \([L/T]\); \( P_{sn} \): solid precipitation \([L/T]\); \( T \): temperature \([°C]\); \( T_{cp1} \): 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)

\[
S' = \max \left( S_{Min} ; S + \frac{S_{Int}}{2} \sin \left( \frac{2\pi n - S_{ph}}{365} \right) \right) \quad \text{(F.4)}
\]
Chapter 2. Hydrological and hydraulics models description

\[ M_{sn} = S' \cdot (1 + b_p \cdot P_w) \cdot (T - T_{cf}) \quad \text{if} \quad T > T_{cf} \]
\[ M_{sn} = S' \cdot \mathrm{CFR} \cdot (T - T_{cf}) \quad \text{if} \quad T \leq T_{cf} \quad (F.5) \]
\[ \frac{dH_{\text{snow}}}{dt} = P_{sn} - M_{sn} \]
\[ M_{sn} \leq P_{sn} + H_{\text{snow}}/dt \quad \text{(F.6)} \]
\[ M_{sn} \geq -W_{\text{snow}}/dt \]

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]; \( S_{\text{Min}} \): Minimal degree-day snowmelt coefficient [L/T/°C]; \( M_{\text{sn}} \): snowmelt or freezing [L/T]; \( b_p \): melt coefficient due to liquid precipitation [T/L]; \( T_{cf} \): critical snowmelt temperature [°C]; \( \mathrm{CFR} \): refreezing coefficient [-]; \( H_{\text{snow}} \): water content of the solid fraction of snow [L]; \( dt \): time step [T]; \( W_{\text{snow}} \): water content of the liquid fraction of snow [L].

The \( S_{\text{ph}} \) 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):

\[ \theta = \frac{W_{\text{snow}}}{H_{\text{snow}}} \quad (F.7) \]
\[ P_{eq} = P_w + \frac{W_{\text{snow}}}{dt} \quad \text{if} \quad H_{\text{snow}} = 0 \]
\[ P_{eq} = 0 \quad \text{if} \quad H_{\text{snow}} > 0 \quad \text{et} \quad \theta \leq \theta_{cr} \quad (F.8) \]
\[ P_{eq} = (\theta - \theta_{cr}) \cdot \frac{H_{\text{snow}}}{dt} \quad \text{if} \quad H_{\text{snow}} > 0 \quad \text{et} \quad \theta > \theta_{cr} \]
\[ \frac{dW_{\text{snow}}}{dt} = P_w + M_{\text{sn}} - P_{eq} \quad (F.9) \]

with \( \theta \) (Theta in RS MINERVE): relative water content in the snow pack [-]; \( \theta_{cr} \) (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 (\( R_{\text{sn}} \)), generating an outflow according to equations F.10 and F.11.

\[ \frac{dH_{R_{\text{sn}}}}{dt} = P_{eq} - K_{\text{sn}} \cdot H_{R_{\text{sn}}} \quad (F.10) \]
\[ Q_{\text{snow}} = K_{\text{sn}} \cdot H_{R_{\text{sn}}} \cdot A \quad (F.11) \]

with \( P_{eq} \): equivalent precipitation [L/T]; \( H_{R_{\text{sn}}} \): level in linear snow reservoir [L]; \( K_{\text{sn}} \): release coefficient of linear snow reservoir [1/T]; \( Q_{\text{snow}} \): outflow of linear snow reservoir [L/T]; \( A \): glacier surface [L²].

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

\[ \text{SWE} = H_{\text{snow}} + W_{\text{snow}} \quad (F.12) \]

with SWE: snow water equivalent [L].

The glacier melt sub-model only provides a discharge when the snow level is zero (\( H_{\text{snow}}=0 \)).

The water intensity produced by the glacier melt (\( P_{\text{eqGL}} \)) is transferred to the linear glacier reservoir (\( R_{\text{gl}} \)) and the resulting discharge (\( Q_{\text{glacier}} \)) at the outlet of the sub-catchment.

The glacier melt \( Q_{\text{glacier}} \) is defined according to equations F.13 to F.16:

\[ G' = \max \left( G_{\text{Min}} \cdot G + \frac{G_{\text{int}}}{2} \cdot \sin \left( 2\pi \frac{n - S_{\text{ph}}}{365} \right) \right) \quad (F.13) \]
\[ P_{eqGL} = 0 \quad \text{if} \quad T \leq T_{cg} \quad \text{or} \quad H_{\text{snow}} > 0 \]
\[ P_{eqGL} = G \cdot (T - T_{cr}) \quad \text{if} \quad T > T_{cg} \quad \text{and} \quad H_{\text{snow}} = 0 \quad (F.14) \]
\[
\frac{dH_{Rgl}}{dt} = P_{eqGL} - K_{gl} \cdot H_{Rgl} \\
Q_{glacier} = K_{gl} \cdot H_{Rgl} \cdot A
\]

with \( P_{eqGL} \): glacier melt \([L/T]\); \( T_{cg} \): critical glacier melt temperature \([°C]\); \( G \): degree-day glacier melt coefficient \([L/T/°C]\); \( G_{Min} \): 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 \( Q_{snow} \):

\[
Q = Q_{snow} + Q_{glacier}
\]

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

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

The parameters to adjust are \( S \), \( S_{Int} \), CFR, \( \theta_{cr} \), \( G \), \( G_{Int} \), \( K_{gl} \) and \( K_{sn} \). The parameters \( S_{ph} \), \( b_p \), \( \theta_{cr} \), \( T_{cp1} \), \( T_{cp2} \), \( T_{cf} \) and \( T_{cg} \) can be assumed as constant (\( b_p = 0.0125 \), \( \theta_{cr} = 0.1 \), \( T_{cp1} = 0 \, ^{°}C \), \( T_{cp2} = 4 \, ^{°}C \), \( T_{cf} = 0 \, ^{°}C \), \( T_{cg} = 0 \, ^{°}C \), \( S_{ph} = 80 \) (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.

![SOCONT Model](image)

*Figure 6 SOCONT Model*
### Table 5 List of parameters and initial conditions for the SOCONT model

<table>
<thead>
<tr>
<th>Object</th>
<th>Name</th>
<th>Units</th>
<th>Description</th>
<th>Regular Range</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SOCONT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>m²</td>
<td>Surface</td>
<td>&gt;0</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>mm/°C/d</td>
<td>Reference degree-day snowmelt coefficient</td>
<td>0.5 to 20</td>
</tr>
<tr>
<td></td>
<td>SInt</td>
<td>mm/°C/d</td>
<td>Degree-day snowmelt interval</td>
<td>0 to 4</td>
</tr>
<tr>
<td></td>
<td>SMIn</td>
<td>mm/°C/d</td>
<td>Minimal degree-day snowmelt coefficient</td>
<td>≥0</td>
</tr>
<tr>
<td></td>
<td>SPh</td>
<td>d</td>
<td>Phase shift of the sinusoidal function</td>
<td>1 to 365</td>
</tr>
<tr>
<td></td>
<td>ThetaCri</td>
<td></td>
<td>Critical relative water content of the snow pack</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>bp</td>
<td>d/mm</td>
<td>Melt coefficient due to liquid precipitation</td>
<td>0.0125</td>
</tr>
<tr>
<td></td>
<td>Tcp1</td>
<td>°C</td>
<td>Minimum critical temperature for liquid precipitation</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Tcp2</td>
<td>°C</td>
<td>Maximum critical temperature for solid precipitation</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Tcf</td>
<td>°C</td>
<td>Critical snowmelt temperature</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>HGR3Max</td>
<td>m</td>
<td>Maximum height of infiltration reservoir</td>
<td>0 to 2</td>
</tr>
<tr>
<td></td>
<td>KGR3</td>
<td>1/s</td>
<td>Release coefficient of infiltration reservoir</td>
<td>0.00025 to 0.1</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>m</td>
<td>Length of the plane</td>
<td>&gt;0</td>
</tr>
<tr>
<td></td>
<td>J0</td>
<td></td>
<td>Runoff slope</td>
<td>&gt;0</td>
</tr>
<tr>
<td></td>
<td>Kr</td>
<td>m¹³/s</td>
<td>Strickler coefficient</td>
<td>0.1 to 90</td>
</tr>
<tr>
<td></td>
<td>CFR</td>
<td></td>
<td>Refreezing coefficient</td>
<td>0 to 1</td>
</tr>
<tr>
<td></td>
<td>SWEIni</td>
<td>m</td>
<td>Initial snow water equivalent height</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HGR3Ini</td>
<td>m</td>
<td>Initial level in infiltration reservoir</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HrlIni</td>
<td>m</td>
<td>Initial runoff water level downstream of the surface</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ThetaIni</td>
<td></td>
<td>Initial relative water content in the snow pack</td>
<td></td>
</tr>
</tbody>
</table>

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):

\[
P_w = \alpha \cdot P \quad \text{(G.1)}
\]

\[
P_{sn} = (1 - \alpha) \cdot P \quad \text{(G.2)}
\]

\[
\alpha = 0 \quad \text{if } T \leq T_{cp1}
\]

\[
\alpha = \frac{(T - T_{cp1})(T_{cp2} - T_{cp1})}{T_{cp2} - T_{cp1}} \quad \text{if } T_{cp1} < T < T_{cp2}
\]

\[
\alpha = 1 \quad \text{if } T \geq T_{cp2} \quad \text{(G.3)}
\]

with $P_w$: liquid precipitation [L/T]; $\alpha$: separation factor; $P$: precipitation [L/T]; $P_{sn}$: solid precipitation [L/T]; $T$: temperature [°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):

\[
S' = \max \left( S_{Min} ; S + \frac{S_{Int}}{2} \sin(2\pi \frac{n - S_{Ph}}{365}) \right) \quad \text{(G.4)}
\]
Chapter 2. Hydrological and hydraulics models description

\[ M_{sn} = S'(1 + b_p \cdot P_w) \cdot (T - T_{cf}) \quad \text{if} \quad T > T_{cf} \]
\[ M_{sn} = S' \cdot CFR \cdot (T - T_{cf}) \quad \text{if} \quad T \leq T_{cf} \]  
\[ dH_{snow}/dt = P_{sn} - M_{sn} \]  
\[ M_{sn} \leq P_{sn} + H_{snow}/dt \]  
\[ M_{sn} \geq -W_{snow}/dt \]  

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

The \( S_{\text{Ph}} \) 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 \( (equations \ G.7 \text{ to } G.9) \):

\[ \theta = W_{\text{snow}}/H_{\text{snow}} \]  
\[ P_{eq} = P_w + W_{\text{snow}}/dt \quad \text{if} \quad H_{\text{snow}} = 0 \]  
\[ P_{eq} = 0 \quad \text{if} \quad H_{\text{snow}} > 0 \quad \text{et} \quad \theta \leq \theta_{cr} \]  
\[ P_{eq} = (\theta - \theta_{cr}) \cdot H_{\text{snow}}/dt \quad \text{if} \quad H_{\text{snow}} > 0 \quad \text{et} \quad \theta > \theta_{cr} \]  
\[ dW_{\text{snow}}/dt = P_w + M_{sn} - P_{eq} \]  

\textit{with} \, \theta \, (\text{Theta in RS MINERVE}): \text{relative water content in the snow pack} \, [-] \; ; \, \theta_{cr} \, (\text{ThetaCri in RS MINERVE}): \text{critical relative water content in the snow pack} \, [-] \; ; \, P_{eq}: \text{equivalent precipitation} \, [\text{L/T}] .

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

\[ SWE = H_{\text{snow}} + W_{\text{snow}} \]  

\textit{with} SWE: \text{snow water equivalent} \, [\text{L}] .

Next, the infiltration reservoir is computed as follows:

\[ i_{inf} = P_{eq} \cdot (1 - (H_{GR3}/H_{GR3Max})^2) \quad \text{if} \quad H_{GR3} \leq H_{GR3Max} \]  
\[ i_{inf} = 0 \quad \text{if} \quad H_{GR3} > H_{GR3Max} \]  
\[ ETR = ETP \cdot \sqrt{H_{GR3}/H_{GR3Max}} \quad \text{if} \quad H_{GR3} \leq H_{GR3Max} \]  
\[ ETR = ETP \quad \text{if} \quad H_{GR3} > H_{GR3Max} \]  
\[ i_{Net} = P_{eq} - i_{inf} \]  
\[ Q = K_{GR3} \cdot H_{GR3} \cdot A \quad \text{if} \quad H_{GR3} \leq H_{GR3Max} \]  
\[ Q = K_{GR3} \cdot H_{GR3Max} \cdot A \quad \text{if} \quad H_{GR3} > H_{GR3Max} \]  
\[ dH_{GR3}/dt = i_{inf} - ETR - Q/A \]  

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

\[
\frac{dH_r}{dt} = 2 \cdot (i_{Net} - i_r) \\
H_r \geq 0
\] (G.16)

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

\[
Q = i_r \cdot A
\] (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 \(\theta\) (Theta in RS MINERVE), SWE, \(H_{GR3}\) and \(H_r\). The parameters to adjust are \(S\), \(S_{Int}\), \(b_p\), CFR, \(\theta_{cr}\), \(K_{GR3}\), \(H_{GR3Max}\) and \(K_r\). The parameters \(S_{Phi}\), \(b_p\), \(\theta_{cr}\), \(T_{cp1}\), \(T_{cp2}\) 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 SOCINT 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.

![HBV model diagram](image)

**Figure 7** HBV model
Table 6 List of parameters and initial conditions for the HBV model

<table>
<thead>
<tr>
<th>Object</th>
<th>Name</th>
<th>Units</th>
<th>Description</th>
<th>Regular Range</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HBV</td>
<td></td>
<td>Surface of the basin</td>
<td>&gt;0</td>
</tr>
<tr>
<td>A</td>
<td>m²</td>
<td></td>
<td>Melting factor</td>
<td>0.5 to 20</td>
</tr>
<tr>
<td>CFR</td>
<td>-</td>
<td>Refreezing factor</td>
<td></td>
<td>0.05</td>
</tr>
<tr>
<td>CWH</td>
<td>-</td>
<td>Critical relative water content of the snow pack</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>TT</td>
<td>°C</td>
<td>Temperature of rain/snow</td>
<td>0 to 3</td>
<td></td>
</tr>
<tr>
<td>TTInt</td>
<td>°C</td>
<td>Temperature interval for rain/snow mixing</td>
<td>0 to 3</td>
<td></td>
</tr>
<tr>
<td>TTSM</td>
<td>°C</td>
<td>Threshold temperature for snow melt</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Beta</td>
<td>-</td>
<td>Model parameter (shape coefficient)</td>
<td>1 to 5</td>
<td></td>
</tr>
<tr>
<td>FC</td>
<td>m</td>
<td>Maximum soil storage capacity</td>
<td>0.05 to 0.65</td>
<td></td>
</tr>
<tr>
<td>PWP</td>
<td>-</td>
<td>Soil permanent wilting point threshold</td>
<td>0.03 to 1</td>
<td></td>
</tr>
<tr>
<td>SUMax</td>
<td>m</td>
<td>Upper reservoir water level threshold</td>
<td>0 to 0.10</td>
<td></td>
</tr>
<tr>
<td>Kr</td>
<td>1/d</td>
<td>Near surface flow storage coefficient</td>
<td>0.05 to 0.5</td>
<td></td>
</tr>
<tr>
<td>Ku</td>
<td>1/d</td>
<td>Interflow storage coefficient</td>
<td>0.01 to 0.4</td>
<td></td>
</tr>
<tr>
<td>Kl</td>
<td>1/d</td>
<td>Baseflow storage coefficient</td>
<td>0 to 0.15</td>
<td></td>
</tr>
<tr>
<td>Kperc</td>
<td>1/d</td>
<td>Percolation storage coefficient</td>
<td>0.0 to 0.8</td>
<td></td>
</tr>
<tr>
<td>SWEIni</td>
<td>m</td>
<td>Initial snow water equivalent height</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>WHIni</td>
<td>-</td>
<td>Initial relative water content in the snow pack</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>HumIni</td>
<td>m</td>
<td>Initial humidity</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>SUIni</td>
<td>m</td>
<td>Initial upper reservoir water level</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>SLIni</td>
<td>m</td>
<td>Initial lower reservoir water level</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

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.

\[
RF = \alpha \cdot P \quad (H.1)
\]

\[
SF = (1 - \alpha) \cdot P \quad (H.2)
\]

\[
\alpha = 0 \quad \text{if} \quad T < TT - TTInt/2
\]

\[
\alpha = \frac{T - (TT - TTInt/2)}{TTInt} \quad \text{if} \quad TT - TTInt/2 < T < TT + TTInt/2 \quad (H.3)
\]

\[
\alpha = 1 \quad \text{if} \quad T > TT + TTInt/2
\]

with $RF$: rainfall [L/T]; $\alpha$: 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:

\[
M_{sn} = CFMax \cdot (T - TTSM) \quad \text{if} \quad T > TTSM \quad (H.4)
\]

\[
M_{sn} = CFR \cdot CFMax \cdot (T - TTSM) \quad \text{if} \quad T \leq TTSM
\]

\[
dH_{snow}/dt = SF - M_{sn} \quad (H.5)
\]

\[
M_{sn} \leq SF + H_{snow}/dt
\]

\[
M_{sn} \geq -W_{snow}/dt
\]
with $M_{sn}$: snowmelt or freezing [L/T]; $CF_{Max}$: 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):

$$WH = W_{snow}/H_{snow}$$  \hspace{1cm} (H.6)

$$P_{eq} = RF + W_{snow}/dt \quad \text{if} \quad H_{snow} = 0$$

$$P_{eq} = 0 \quad \text{if} \quad H_{snow} > 0 \text{ and } WH \leq CWH$$

$$P_{eq} = (WH - CWH) \cdot H_{snow}/dt \quad \text{if} \quad H_{snow} > 0 \text{ and } WH > CWH$$  \hspace{1cm} (H.7)

$$dW_{snow}/dt = RF + M_{sn} - P_{eq}$$  \hspace{1cm} (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 $H_{snow}$ and $W_{snow}$ (equation H.9):

$$SWE = H_{snow} + W_{snow}$$  \hspace{1cm} (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, $P_{eq}$) 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 $FC$.

$$i_{Recharge} = \left(\frac{Hum}{FC}\right)^{Beta} \cdot P_{eq}$$  \hspace{1cm} (H.10)

$$ETR = ETP \cdot \frac{Hum}{(PWP \cdot FC)} \quad \text{if} \quad Hum < (PWP \cdot FC)$$

$$ETR = ETP \quad \text{if} \quad Hum \geq (PWP \cdot FC)$$  \hspace{1cm} (H.11)

$$dHum/dt = (P_{eq} - i_{Recharge}) - ETR \quad Hum \geq 0$$  \hspace{1cm} (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 $K_{r}$.

$$Q_{r} = K_{r} \cdot (SU - SU_{Max}) \cdot A \quad \text{if} \quad SU > SU_{Max}$$

$$Q_{r} = 0 \quad \text{if} \quad SU \leq SU_{Max}$$  \hspace{1cm} (H.13)

with $Q_{r}$: Near surface flow (or run-off flow) [L/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 [L²].

The Upper reservoir (or interflow reservoir), corresponding to the upper soil storage and producing the interflow, is computed as follows:
\[
d\frac{SU}{dt} = i_{\text{Recharge}} - (K_{\text{perc}} + K_u) \cdot SU - Q_r / A \\
SU \geq 0
\]

\[
i_{\text{perc}} = K_{\text{perc}} \cdot SU
\]

\[
Q_u = K_u \cdot SU \cdot A
\]

with \( K_{\text{perc}} \): Percolation storage coefficient [1/T]; \( K_u \): Interflow storage coefficient [1/T]; \( i_{\text{perc}} \): 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

\[
d\frac{SL}{dt} = i_{\text{perc}} - K_l \cdot SL \\
SL \geq 0
\]

\[
Q_l = K_l \cdot SL \cdot A
\]

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:

\[
Q_{\text{tot}} = Q_r + Q_u + Q_l
\]

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

The initial conditions associated to this model are \( \text{SWE}_{\text{ini}}, \text{WH}_{\text{ini}}, \text{Hum}_{\text{ini}}, \text{SU}_{\text{ini}} \) and \( SL_{\text{ini}} \). The parameters to adjust are CFMax, CFR, CWH, TT, TTInt, TTSM, Beta, FC, PWP, \( SU_{\text{max}} \), \( K_r \), \( K_u \), \( K_l \) and \( K_{\text{perc}} \). 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 (\( Q_{\text{tot}} \)).
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](image)
Chapter 2. Hydrological and hydraulics models description

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

<table>
<thead>
<tr>
<th>Object</th>
<th>Name</th>
<th>Units</th>
<th>Description</th>
<th>Regular Range</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>m²</td>
<td>Surface of the basin</td>
<td>&gt;0</td>
</tr>
<tr>
<td></td>
<td>X1</td>
<td>m</td>
<td>Capacity of production store</td>
<td>0.01 to 1.2</td>
</tr>
<tr>
<td>GR4J</td>
<td>X2</td>
<td>m</td>
<td>Water exchange coefficient</td>
<td>-0.005 to 0.003</td>
</tr>
<tr>
<td></td>
<td>X3</td>
<td>m</td>
<td>Capacity of routing store</td>
<td>0.01 to 0.5</td>
</tr>
<tr>
<td></td>
<td>X4</td>
<td>d</td>
<td>UH time base</td>
<td>-0.5 to 1</td>
</tr>
<tr>
<td></td>
<td>Sini</td>
<td>m</td>
<td>Initial water content in the production reservoir</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>RIni</td>
<td>m</td>
<td>Initial water level in the routing reservoir</td>
<td>-</td>
</tr>
</tbody>
</table>

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.

\[
P_n = \frac{P - ETP}{En} \quad \text{if} \quad P \geq ETP
\]

\[
P_n = 0 \quad \text{if} \quad P < ETP
\]

\[\text{with } P_n: \text{ net rain intensity } [\text{L/T}]; \quad \text{Peq: Equivalent precipitation } [\text{L/T}]; \quad \text{ETP: potential evapotranspiration } [\text{L/T}]; \quad \text{En: net evapotranspiration } [\text{L/T}].\]

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.

\[
Ps = \frac{d}{dt} \left(1 - \left(\frac{S}{X_1}\right)^2\right) \cdot \tanh \left(\frac{P \cdot dt}{X_1}\right)
\]

\[
Es = \frac{d}{dt} \left(2 - \left(\frac{S}{X_1}\right)^2\right) \cdot \tanh \left(\frac{E \cdot dt}{X_1}\right)
\]

\[\text{with } Ps: \text{ rain intensity feeding the production reservoir } [\text{L/T}]; \quad X_1: \text{ maximum capacity of production reservoir } S [\text{L}]; \quad S: \text{ water content in the production reservoir } [\text{L}]; \quad Es: \text{ evapotranspiration from the production reservoir } [\text{L/T}].\]

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

\[
iPerc = \frac{d}{dt} \left(S + (Ps - Es) \cdot dt\right) \cdot \left(1 - \left(1 + \left(\frac{4}{9} \left(\frac{S + (Ps - Es) \cdot dt}{X_1}\right)\right)^2\right)^{-\frac{1}{4}}\right)
\]

\[\text{with } iPerc: \text{ percolation intensity } [\text{L/T}].\]

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

\[
dS/dt = Ps - Es - iPerc
\]

\[S \geq 0\]
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.

\[
SH_{1r} = \begin{cases} 
0 & \text{if } t \leq 0 \\
\left(\frac{t}{X_4}\right)^{\frac{5}{2}} & \text{if } 0 < t < X_4 \\
1 & \text{if } t \geq X_4
\end{cases} \tag{I.7}
\]

\[
SH_{2r} = \begin{cases} 
0 & \text{if } t \leq 0 \\
\frac{1}{2} \cdot \left(\frac{t}{X_4}\right)^{\frac{5}{2}} & \text{if } 0 < t < X_4 \\
\frac{1}{2} \cdot \left(2 - \frac{t}{X_4}\right)^{\frac{5}{2}} & \text{if } X_4 < t < 2 \cdot X_4 \\
1 & \text{if } t \geq 2 \cdot X_4
\end{cases} \tag{I.8}
\]

\[
UH_{1j} = SH_{1j} - SH_{1j-1} \tag{I.9}
\]

\[
UH_{2j} = SH_{2j} - SH_{2j-1} \tag{I.10}
\]

with \(SH_1\) : hydrograph 1 from \(S\); \(SH_2\) : hydrograph 2 from \(S\); \(t\) : time step [T]; \(X_4\) : base time [T]; \(UH_1\) : hydrograph from \(SH_1\); \(UH_2\) : hydrograph from \(SH_2\).

Then, at each time step \(k\), outflows \(Q_9\) and \(Q_1\) 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.

\[
Q_9 = 0.9 \cdot \sum_{j=1}^{i} (UH_{1j} \cdot iPr_{k-j-1}) \tag{I.11}
\]

\[
Q_1 = 0.1 \cdot \sum_{j=1}^{m} (UH_{2j} \cdot iPr_{k-j-1}) \tag{I.12}
\]

with \(Q_9\) : Unitary flow from the UH1 hydrograph; [L/T]; \(Q_1\) : Unitary flow from the UH2 hydrograph [L/T]; \(M\) : day (integer) [T]; \(l\) = integer value of \(X_4 + 1\) [T]; \(m\) = integer value of \(2 \cdot X_4 + 1\) [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.

\[
iF = \frac{d}{dt} \left(\frac{X_2}{X_3}\right)^{\frac{7}{2}} \tag{I.13}
\]
\[
Q_r' = \frac{d}{dt} (R + (Q_9 + iF) \cdot dt) \cdot \left(1 - \left(1 + \left(\frac{R + (Q_9 + iF) \cdot dt}{X_3}\right)^4\right)^{\frac{1}{4}}\right)
\]  

(I.14)

*with* \( iF \): exchange function \([L/T]\); \( X_2 \): 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 \([L]\); \( X_3 \): one day capacity of routing reservoir \([L]\); \( Q_r' \): unitary routing reservoir outflow \([L/T]\).

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

\[
dR/dt = Q_9 - iF - Q_r'
\]

\[ R \geq 0 \]  

(I.15)

The outflow \( Q_1 \) from hydrograph UH2 has also the same exchange for providing the complementary flow \( Q_d' \):

\[
Q_d' = Q_1 + iF
\]  

(I.16)

*with* \( Q_d' \): 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, \( Q_{tot} \), as presented in equation I.19:

\[
Q_r = Q_r' \cdot A
\]  

(I.17)

\[
Q_d = Q_d' \cdot A
\]  

(I.18)

\[
Q_{tot} = Q_r + Q_d
\]  

(I.19)

*with* \( Q_r \): outflow from the routing reservoir \([L^3/T]\); \( A \): Basin surface \([L^2]\); \( Q_d \): complementary outflow \([L^3/T]\); \( Q_{tot} \): total outflow \([L^3/T]\).

The variables associated to this model are \( S_{ini} \) and \( R_{ini} \). The parameters to adjust are \( X_1 \), \( X_2 \), \( X_3 \), and \( X_4 \). 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.

![SAC-SMA model](Figure 9 SAC-SMA model)
Table 8: List of parameters and initial conditions for the SAC-SMA model

<table>
<thead>
<tr>
<th>Object</th>
<th>Name</th>
<th>Units</th>
<th>Description</th>
<th>Regular Range</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>m²</td>
<td>Surface of the basin</td>
<td>&gt;0</td>
</tr>
<tr>
<td></td>
<td>Adimp</td>
<td>-</td>
<td>Maximum fraction of an additional impervious area due to saturation</td>
<td>0 to 0.2</td>
</tr>
<tr>
<td></td>
<td>Pctim</td>
<td>-</td>
<td>Permanent impervious area fraction</td>
<td>0 to 0.05</td>
</tr>
<tr>
<td></td>
<td>Riva</td>
<td>-</td>
<td>Riparian vegetation area fraction</td>
<td>0 to 0.2</td>
</tr>
<tr>
<td></td>
<td>UztwMax</td>
<td>m</td>
<td>Upper Zone Tension Water capacity</td>
<td>0.01 to 0.15</td>
</tr>
<tr>
<td></td>
<td>UzfwMax</td>
<td>m</td>
<td>Upper Zone Free Water capacity</td>
<td>0.005 to 0.10</td>
</tr>
<tr>
<td></td>
<td>Uzk</td>
<td>1/d</td>
<td>Interflow depletion rate from the Upper Zone Free Water storage</td>
<td>0.10 to 0.75</td>
</tr>
<tr>
<td>SAC-SMA</td>
<td>Zperc</td>
<td>-</td>
<td>Ratio of maximum and minimum percolation rates</td>
<td>10 to 350</td>
</tr>
<tr>
<td></td>
<td>Rexp</td>
<td>-</td>
<td>Shape parameter of the percolation curve</td>
<td>1 to 4</td>
</tr>
<tr>
<td></td>
<td>Pfree</td>
<td>-</td>
<td>Percolation fraction that goes directly to the Lower Zone Free Water storage</td>
<td>0 to 0.6</td>
</tr>
<tr>
<td></td>
<td>LztwMax</td>
<td>m</td>
<td>The Lower Zone Tension Water capacity</td>
<td>0.05 to 0.40</td>
</tr>
<tr>
<td></td>
<td>LzfpMax</td>
<td>m</td>
<td>The Lower Zone primary Free Water capacity</td>
<td>0.03 to 0.80</td>
</tr>
<tr>
<td></td>
<td>LzfsMax</td>
<td>m</td>
<td>The Lower Zone supplementary Free Water capacity</td>
<td>0.01 to 0.40</td>
</tr>
<tr>
<td></td>
<td>Rserv</td>
<td>-</td>
<td>Fraction of Lower Zone Free Water not transferable to Lower Zone Tension Water</td>
<td>0 to 1</td>
</tr>
<tr>
<td></td>
<td>Lzpk</td>
<td>1/d</td>
<td>Depletion rate of the Lower Zone primary Free Water storage</td>
<td>0.001 to 0.03</td>
</tr>
<tr>
<td></td>
<td>Lzsk</td>
<td>1/d</td>
<td>Depletion rate of the Lower Zone supplemental Free Water storage</td>
<td>0.02 to 0.3</td>
</tr>
<tr>
<td></td>
<td>Side</td>
<td>-</td>
<td>Ratio of deep percolation from Lower Zone Free Water storages</td>
<td>0 to 0.5</td>
</tr>
<tr>
<td></td>
<td>AdimIni</td>
<td>m</td>
<td>Initial Tension Water content of the Adimp area</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>UztwIni</td>
<td>m</td>
<td>Initial Upper Zone Tension Water content</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>UzfwIni</td>
<td>m</td>
<td>Initial Upper Zone Free Water content</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>LztwIni</td>
<td>m</td>
<td>Initial Lower Zone Tension Water content</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>LzfpIni</td>
<td>m</td>
<td>Initial Lower Zone Free supplementary content</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>LzfsIni</td>
<td>m</td>
<td>Initial Lower Zone Free primary content</td>
<td>-</td>
</tr>
</tbody>
</table>

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 $Q_{\text{DirectRunoff}}$ 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

\[
Parea = 1 - (Pctim + Adimp)
\]

\(Parea\) : 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:

\[
T = \frac{UztwC}{UztwMax}
\]

\(UztwC\) (J.2)

\[
F = \frac{UzfwC}{UzfwMax}
\]

\(UzfwC\) (J.3)

\[
Uztr = d \left( \frac{UzfwC \cdot UztwMax - UztwC \cdot UzfwMax}{UzfwMax + UztwMax} \right) / dt \quad \text{if } T < F
\]

\(Uztr\) (J.4)

\[
Uztr = 0 \quad \text{if } T \geq F
\]

\(Uztr\) (J.5)
Chapter 2. Hydrological and hydraulics models description

with \( T \): relative content of Upper Zone Tension Water [-]; \( F \): relative content of Upper Zone Free Water [-]; \( Uzt \): 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.

\[
PBase = LzfpMax \cdot Lzpk + LzfsMax \cdot Lzsk
\]

\[
DEFR = \frac{(LztwMax - LztwC) + (LzfpMax - LzfpC) + (LzfsMax - LzfsC)}{LztwMax + LzfpMax + LzfsMax}
\]

\[
DDA = PBase \cdot (1 + Zperc \cdot DEFR^{exp})
\]

percolation rates [-]; DEF: 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):

\[
PERC = DDA \frac{UzfwC}{UzfwMax} \quad (J.9)
\]

\[
PERC \leq d(UzfwC)/dt \quad (J.10)
\]

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

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:

\[
Q_{interflow} = Uzk \cdot UzfwC \cdot Parea \cdot A \quad (J.12)
\]

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. 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:

\[
RFW = R_{serv} \cdot (LzfpMax + LzfsMax) \\
T = \frac{LztwC}{LztwMax} \\
F = \frac{LzfpC + LzfsC - RFW}{LzfpMax + LzfsMax - RFW}
\]

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]; \( R_{serv} \): 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:

\[
R = \frac{LzfpC + LzfsC + LztwC - RFW}{LzfpMax + LzfsMax + LztwMax - RFW} \\
DEL = LztwMax \cdot \frac{d(R - T)}{dt}
\]

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 \( P_{free} \) 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 \(Q_{\text{primary}}\) and supplementary flow \(Q_{\text{supplementary}}\)) follows the Darcy's law and forms the baseflow of the **Lower Zone** \(Q_{\text{Lower Zone}}\) as shown in equations J.18 to J.20:

\[
Q_{\text{primary}} = Lzpk \cdot LzfpC \cdot Parea \cdot A \tag{J.18}
\]

\[
Q_{\text{supplementary}} = Lzsk \cdot LzfsC \cdot Parea \cdot A \tag{J.19}
\]

\[
Q_{\text{Lower Zone}} = Q_{\text{primary}} + Q_{\text{supplementary}} \tag{J.20}
\]

*with* \(Q_{\text{primary}}\): primary baseflow \([L^3/T]\); \(Q_{\text{supplementary}}\): supplementary baseflow \([L^3/T]\); \(Q_{\text{Lower Zone}}\): 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 \(Q_{\text{SubSurf}}\), 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:

\[
Q_{\text{SubSurf}} = \text{Side} \cdot Q_{\text{baseflow}} \tag{J.21}
\]

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

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

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

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

**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:

\[
RED = ETP - ETR1
\]

\[
d(UztwC)/dt = -ETR1
\]  \hspace{1cm} (J.25) \hspace{1cm} (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:

\[
ETR2 = \min\{RED; d(UzfwC)/dt\}
\]

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

\[
REDLz = RED - ETR2
\]

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

\[
d(UzfwC)/dt = -ETR2
\]

- **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:

\[
ETR3 = REDLz \cdot \frac{LztwC}{UztwMax + LztwMax} \leq d(LztwC)/dt
\]

- **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:

\[
ETR4 = ETP - ETR1 - ETR2 - ETR3
\]

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

\[
ETR5 = ETR1 + (RED + ETR2) \cdot \frac{AdimC - UztwC - V_{ETR1}}{UztwMax + LztwMax}
\]

\[
V_{ETR1} = \int_{t1}^{t2} ETR1 \cdot dt
\]

*with AdimC: sum of the Upper Zone and Lower Zone Tension Water Content in the Adimp area [-]; \(V_{ETR1}\): evapotranspired 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:

$$ETR_{\text{total}} = [(ETR1 + ETR2 + ETR3) \cdot Parea + ETR4 \cdot Riva + ETR5 \cdot Adimp] \cdot A$$  \hspace{1cm} \text{(J.34)}

with $ETR_{\text{total}}$: total evapotranspiration $[\text{L}^3/\text{T}]$; $Parea$: 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 $[\text{L}^2]$.

A more detailed guide about the SAC-SMA equations can be found on the following National Weather Service link: \url{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

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

St. Venant routing

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

\[
\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0
\]  \hfill (K.1)

\[
\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x} \left( \frac{Q^2}{A} + g \cdot I_1 \right) = g \cdot A \cdot (J_0 - J_f) + g \cdot I_2
\]  \hfill (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:

\[
I_1 = \int_0^h (h - \eta) \cdot b(\eta) \, d\eta
\]  \hfill (K.3)

The term \(b\) represents the cross sectional variation for the level \(\eta\) 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 \(I_1\)](image1)

![Figure 11 Transversal profile available for the computation of channel routing](image2)
For this trapezoidal section, the equation defining $I_1$ is reduced to:

$$I_1 = \frac{B \cdot h^2}{2} + \frac{m \cdot h^3}{3}$$  \hspace{1cm} (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:

$$J_f = \frac{Q \cdot |Q|}{A^2 \cdot K^2 \cdot R_h^{4/3}}$$  \hspace{1cm} (K.5)

$$R_h = \frac{A}{L_p}$$  \hspace{1cm} (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:

$$I_2 = \int_0^h (h-\eta) \frac{\partial b}{\partial x} \Big|_{h=\eta}$$  \hspace{1cm} (K.7)

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

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

$$Q_{j+1}^{n+1} = Q_{j+1}^n - \Delta t \frac{\Delta x}{A_{j+1}^n} \left( \frac{(Q_{j+1}^{n+1})^2}{A_{j+1}^n} - \frac{(Q_j^n)^2}{A_j^n} + g \cdot I_{1,j+1}^n - g \cdot I_{1,j}^n \right) + \Delta t \cdot g \cdot \frac{(A_j^n + A_{j+1}^n)}{2} \cdot J_0 - \Delta t \cdot g \cdot A_{j+1}^n \cdot J_{f_{j+1}}^n$$  \hspace{1cm} (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]; $\Delta x$: 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:

$$\frac{\partial I_1}{\partial x} = A \cdot (J_0 - J_f) + I_2$$ \hspace{1cm} (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:

$$\frac{\partial Q}{\partial t} + \left( \frac{Q}{B \cdot D \cdot dh} \right) \cdot \frac{\partial Q}{\partial x} - \frac{D^2}{2 \cdot B \cdot |Q|} \cdot \frac{\partial^2 Q}{\partial x^2} = 0$$ \hspace{1cm} (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:

$$Q = D \cdot J_0^{1/2}$$ \hspace{1cm} (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 $\delta$ (equation K.14):

$$c = \frac{Q}{B \cdot D} \cdot \frac{dD}{dh}$$ \hspace{1cm} (K.13)

$$\delta = \frac{D^2}{2B \cdot |Q|}$$ \hspace{1cm} (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:

$$\frac{\partial Q}{\partial t} + \left( \frac{dQ}{dA} \right)_{x_0} \cdot \frac{\partial Q}{\partial x} = 0$$ \hspace{1cm} (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:

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

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

Applying this scheme to equation K.15 yields:

$$\frac{X \cdot (Q_j^{n+1} - Q_j^n) + (1 + X) \cdot (Q_{j+1}^{n+1} - Q_{j+1}^n)}{\Delta t} + \frac{1}{2} \cdot (Q_{j+1}^{n+1} - Q_{j+1}^n) + \frac{1}{2} \cdot (Q_{j+1}^n - Q_j^n) = 0$$ \hspace{1cm} (K.18)
We can express the solution of this equation as a function of the unknown variable $Q_{j+1}^{n+1}$ namely:

$$Q_{j+1}^{n+1} = C_1 \cdot Q_j^{n+1} + C_2 \cdot Q_j^n + C_3 \cdot Q_{j+1}^n$$  \hspace{1cm} (K.19)

with:

$$C_1 = -\frac{KX - \frac{\Delta t}{2}}{K(1 - X) + \frac{\Delta t}{2}}$$  \hspace{1cm} (K.20)

$$C_2 = \frac{KX + \frac{\Delta t}{2}}{K(1 - X) + \frac{\Delta t}{2}}$$

$$C_3 = \frac{K(1 - X) - \frac{\Delta t}{2}}{K(1 - X) + \frac{\Delta t}{2}}$$

$$K = \frac{\Delta x}{c}$$  \hspace{1cm} (K.21)

$$c = \frac{Q_{j+1}^n - Q_j^n}{A_{j+1}^n - A_j^n}$$  \hspace{1cm} (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 $\Delta x/\Delta t = c$ and neglecting the quadratic terms ($\Delta x^2$) the equation can be written as follows:

$$\frac{\partial Q}{\partial t} + c \cdot \frac{\partial Q}{\partial x} - c \cdot B \cdot \frac{\partial^2 Q}{\partial x^2} = 0$$  \hspace{1cm} (K.23)

$$B = \Delta x \cdot \left(\frac{1}{X} - X\right)$$  \hspace{1cm} (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:

$$X = \frac{1}{2} - \frac{D^3}{2 \cdot \Delta x \cdot |Q| \cdot \frac{dD}{dh}}$$  \hspace{1cm} (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.
Chapter 2. Hydrological and hydraulics models description

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:

\[
\frac{\partial Q}{\partial t} + \left(\frac{dQ}{dA}\right)_{x_0} \cdot \frac{\partial Q}{\partial x} = 0
\]  
(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:

\[
c = \frac{\partial Q}{\partial A}
\]  
(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:

\[
c = \frac{Q_{j+1}^{n} - Q_{j}^{n}}{A_{j+1}^{n} - A_{j}^{n}}
\]  
(K.28)

\[
\alpha = c \cdot \frac{\Delta t}{\Delta L}
\]  
(K.29)

\[
Q_{j+1}^{n+1} = \alpha \cdot Q_{j}^{n} + (1 - \alpha) \cdot Q_{j+1}^{n}
\]  
\[\text{si} \quad \alpha \leq 1\]

\[
Q_{j+1}^{n+1} = \frac{1}{\alpha} \cdot Q_{j}^{n} + \frac{1}{\alpha} \cdot Q_{j+1}^{n+1}
\]  
\[\text{si} \quad \alpha > 1\]

(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

<table>
<thead>
<tr>
<th>Object</th>
<th>Name</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Series</td>
<td>Series (paired</td>
<td>s - (depending on the series)</td>
<td>Time – Value series</td>
</tr>
<tr>
<td></td>
<td>data)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.12. Reservoir

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

$$\frac{d\forall}{dt} = Q_e - Q_s$$

(L.1)

with $\forall$: volume in the reservoir [L$^3$]; $Q_e$: inflow in the reservoir [L$^3$/T]; $Q_s$: 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 $\forall$ 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

<table>
<thead>
<tr>
<th>Object</th>
<th>Name</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reservoir</td>
<td>$H-V$ (paired data)</td>
<td>masl - m$^3$</td>
<td>Level - Volume relation</td>
</tr>
<tr>
<td></td>
<td>$H_{ini}$</td>
<td>masl</td>
<td>Initial level in the reservoir</td>
</tr>
</tbody>
</table>
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:

\[
Q_{\text{outflow}}^n = 0 \quad \text{if} \quad H^n < H_1
\]

\[
Q_{\text{outflow}}^n = Q_{\text{dev},i} + (H_{i+1} - H^n) \cdot \frac{Q_{\text{outflow},i+1} - Q_{\text{outflow},i}}{H_{i+1} - H_i} \quad \text{if} \quad H_i < H^n < H_{i+1}
\]

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

Table 12 HQ paired data required

<table>
<thead>
<tr>
<th>Object</th>
<th>Name</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HQ</td>
<td>H-Q (paired data)</td>
<td>masl - m³/s</td>
<td>Level - Discharge relation</td>
</tr>
</tbody>
</table>
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:

\[
Q_{\text{turbine}}^n = Q_{\text{wanted}}^n \quad \text{if} \quad (H^n > H_{\text{on}}) \quad \text{or} \quad (H_{\text{off}} \leq H^n \leq H_{\text{on}} \quad \text{and} \quad \text{IsOperating} = 1)
\]
\[
Q_{\text{turbine}}^n = 0 \quad \text{if} \quad (H^n < H_{\text{off}}) \quad \text{or} \quad (H_{\text{off}} \leq H^n \leq H_{\text{on}} \quad \text{and} \quad \text{IsOperating} = 0)
\]

(N.1)

with \( Q_{\text{turbine}}^n \): discharge at instant \( n \) [L^3/T]; \( H^n \): water level at instant \( n \) [L]; \( Q_{\text{wanted}}^n \): discharge flow wanted at instant \( n \) [L^3/T]; \( H_{\text{on}} \): Threshold in the level of the reservoir to start the turbine cycle [L]; \( H_{\text{off}} \): Threshold in the level of the reservoir to stop the turbine cycle [L]; \( \text{IsOperating} \): Planned turbine cycle at instant \( t \) [0/1].

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

<table>
<thead>
<tr>
<th>Object</th>
<th>Name</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HQ</td>
<td>H-Q</td>
<td>masl - m^3/s</td>
<td>Level - Discharge relation</td>
</tr>
<tr>
<td></td>
<td>(paired data)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HQ</td>
<td>Hon</td>
<td>masl</td>
<td>Threshold in the level of the reservoir to start the turbine cycle</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(turbine starts when the threshold is exceeded)</td>
</tr>
<tr>
<td>HQ</td>
<td>Hoff</td>
<td>masl</td>
<td>Threshold in the level of the reservoir to stop the turbine cycle</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(turbine stops when level go below the threshold)</td>
</tr>
<tr>
<td></td>
<td>IsOperating\INI</td>
<td>0/1</td>
<td>First suggested value for the turbine cycle</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0 = not turbine; 1 = turbine). Only taken into account if ( H_{\text{off}} &gt; H &gt; H_{\text{on}} )</td>
</tr>
</tbody>
</table>

Figure 12 Time-H, turbine 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 (\(\nu\)).

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).

\[
Z_{net}^n = (Z_{water}^n - Z_{central}) - f \cdot \frac{8 \cdot L \cdot Q_n^2}{g \cdot \pi^2 \cdot D^5} \quad (O.1)
\]

\[
Power^n = \eta^n \cdot 1000 \cdot Q^n \cdot g \cdot Z_{net}^n \quad (O.2)
\]

\[
Revenue^n = Power^n \cdot Price^n \quad (O.3)
\]

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]; \(\eta\): 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-\(\eta\) of this object, the maximum discharge of the paired data Q-\(\eta\) is taken for the energy production calculations.

**Table 14** Hydropower paired data and parameters required

<table>
<thead>
<tr>
<th>Object</th>
<th>Name</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hydropower</td>
<td>Q-(\eta) (paired data)</td>
<td>m(^3/s) - %</td>
<td>Discharge-Performance relation</td>
</tr>
<tr>
<td>Z_{plant}</td>
<td>masl</td>
<td></td>
<td>Hydropower plant altitude</td>
</tr>
<tr>
<td>L</td>
<td>m</td>
<td></td>
<td>Length of the pipe</td>
</tr>
<tr>
<td>D</td>
<td>m</td>
<td></td>
<td>Diameter of the pipe</td>
</tr>
<tr>
<td>K</td>
<td>m</td>
<td></td>
<td>Roughness</td>
</tr>
<tr>
<td>(\nu)</td>
<td>m(^2/s)</td>
<td></td>
<td>Kinematic viscosity</td>
</tr>
<tr>
<td>Default Price</td>
<td>euro/Kwh</td>
<td></td>
<td>Default price, only used if no data exists in the database</td>
</tr>
</tbody>
</table>

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 \(\lambda\) represents \(f\). In addition, the Reynolds number, \(Re\), is presented in equation N.5.

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

\[
Re = \frac{\nu \cdot D}{\nu} \quad (O.5)
\]
with $\lambda$: friction factor [-]; $k$: roughness [L]; $Re$: Reynolds number [-]; $v$: velocity [L/T]; $\nu$: kinematic viscosity [L$^2$/T]

The equation N.4 is solved for the range of discharges of the paired data $Q-\eta$, providing the paired data relation « Discharge ($Q$) – Friction factor ($\lambda$) » 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:

\[
Q_{up}^n = Q_{down}^n + Q_{diverted}^n
\]  
(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

<table>
<thead>
<tr>
<th>Object</th>
<th>Name</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diversion</td>
<td>( Q_{up} ), ( Q_{diverted} ) (paired data)</td>
<td>m(^3)/s - m(^3)/s</td>
<td>Upstream flow - Diverted flow relation</td>
</tr>
</tbody>
</table>
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.

\[
Q_{supplied}^n = \min \left( \frac{Q_{demand}^n}{1 - \text{LossRate}}, Q_{up}^n \right) \quad (Q.1)
\]

\[
Q_{stream}^n = Q_{up}^n - Q_{supplied}^n \quad (Q.2)
\]

\[
Q_{losses}^n = Q_{supplied}^n \cdot \text{LossRate} \quad (Q.3)
\]

\[
Q_{delivered}^n = Q_{supplied}^n \cdot (1 - \text{LossRate}) \quad (Q.4)
\]

\[
Q_{UnmetDemand}^n = Q_{demand}^n - Q_{delivered}^n \quad (Q.5)
\]

\[
\text{DemandCoverage}^n = \frac{Q_{demand}^n - Q_{delivered}^n}{Q_{demand}^n} \cdot 100 \quad (Q.6)
\]

\[
Q_{consumed}^n = Q_{delivered}^n \cdot \text{ConsumptionRate} \quad (Q.7)
\]

\[
Q_{return}^n = Q_{delivered}^n \cdot (1 - \text{ConsumptionRate}) \quad (Q.8)
\]

\[
Q_{down}^n = Q_{up}^n - Q_{losses}^n - Q_{consumed}^n \quad (Q.9)
\]

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

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

### Figure 13 Distribution of discharges in the Consumer object.

- $Q_{\text{losses}}$ = $f(\text{LossRate})$  
  - $Q_{\text{supplied}}$ = $f(\text{Qdemand, LossRate})$  
  - $Q_{\text{delivered}}$  
  - $Q_{\text{consumed}}$  
  - $Q_{\text{return}}$  
  - $Q_{\text{return}}$  
  - $Q_{\text{stream}}$  
  - $Q_{\text{down}}$
2.18. Structure efficiency

This object needs an efficiency coefficient as information to describe the efficiency of a structure such as 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:

\[
Q_{\text{down}}^n = Q_{\text{up}}^n \cdot \text{Efficiency} \quad \text{(R.1)}
\]

\[
Q_{\text{lost}}^n = Q_{\text{up}}^n \cdot (1 - \text{Efficiency}) \quad \text{(R.2)}
\]

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

Table 17 Structure efficiency parameter required

<table>
<thead>
<tr>
<th>Object</th>
<th>Name</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure</td>
<td>Efficiency</td>
<td>-</td>
<td>Efficiency of the structure</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Combined condition is satisfied when</th>
</tr>
</thead>
<tbody>
<tr>
<td>OR</td>
<td>Left condition OR Right condition are satisfied</td>
</tr>
<tr>
<td>AND</td>
<td>Left condition AND Right condition are satisfied</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Object</th>
<th>Name</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planner</td>
<td>Rule</td>
<td>-</td>
<td>Rule to be applied regarding at least one condition</td>
</tr>
<tr>
<td>Rules</td>
<td>Condition</td>
<td>-</td>
<td>Expression to be evaluated and if it is satisfied the parent rule will be applied</td>
</tr>
</tbody>
</table>

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

Table 20 Operators for rule conditions.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;</td>
<td>Bigger than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Bigger than or equal to</td>
</tr>
<tr>
<td>=</td>
<td>Equal to</td>
</tr>
<tr>
<td>&lt; =</td>
<td>Smaller than or equal to</td>
</tr>
<tr>
<td>&lt;</td>
<td>Smaller than</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Object</th>
<th>Name</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comparator</td>
<td>Reference threshold</td>
<td>(depends on the variable)</td>
<td>Threshold used by threshold-based indicators (returns TRUE if reference value &gt; threshold)</td>
</tr>
<tr>
<td>Comparator</td>
<td>Simulation threshold</td>
<td>(depends on the variable)</td>
<td>Threshold used by threshold-based indicators (returns TRUE if simulation value &gt; threshold)</td>
</tr>
<tr>
<td>Comparator</td>
<td>Warm up period</td>
<td>Days</td>
<td>Warm up period, not used for indicators computation</td>
</tr>
</tbody>
</table>
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.

\[
Nash = 1 - \frac{\sum_{t=t_1}^{t_f}(X_{sim,t} - X_{ref,t})^2}{\sum_{t=t_1}^{t_f}(X_{ref,t} - \bar{X}_{ref})^2}
\]  

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 \(t\); \(\bar{X}_{ref}\): average observed variable (discharge \([L^3/T]\) or height \([L]\)) for the considered period.

It varies from \(-\infty\) 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.

\[
Nash-ln = 1 - \frac{\sum_{t=t_1}^{t_f}((\ln(X_{sim,t}) - \ln(X_{ref,t}))^2}{\sum_{t=t_1}^{t_f}((\ln(X_{ref,t}) - \ln(\bar{X}_{ref}))^2}
\]  

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

It varies from \(-\infty\) to 1, with 1 representing the best performance of the model.

3.3. Pearson Correlation Coefficient

The Pearson correlation coefficient shows the covaribility 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.

\[
Pearson = \frac{\sum_{t=t_1}^{t_f}(X_{sim,t} - \bar{X}_{sim}) \cdot (X_{ref,t} - \bar{X}_{ref})}{\sqrt{\sum_{t=t_1}^{t_f}(X_{sim,t} - \bar{X}_{sim})^2 \cdot \sum_{t=t_1}^{t_f}(X_{ref,t} - \bar{X}_{ref})^2}}
\]  

with Pearson: Pearson Correlation Coefficient [-]; \(\bar{X}_{sim}\): 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.
Chapter 3. Performance indicators

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):

\[
KGE' = 1 - \sqrt{(r - 1)^2 + (\beta - 1)^2 + (\gamma - 1)^2}
\]

\[\beta = \frac{\mu_s}{\mu_o}\]

\[\gamma = \frac{CV_s}{CV_o} = \frac{\sigma_s/\mu_s}{\sigma_o/\mu_o}\]

*with* \(KGE'\): modified KGE-statistic [-]; \(r\): correlation coefficient between simulated and reference values [-]; \(\beta\): bias ratio; \(\gamma\): variability ratio [-]; \(\mu\): mean discharge \([L^3/T]\); \(CV\): coefficient of variation [-]; \(\sigma\): 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.

\[
BS = 1 - \left[ \max \left( \frac{X_{sim}}{X_{ref}}, \frac{X_{ref}}{X_{sim}} \right) - 1 \right]^2
\]

*with* \(BS\): Bias Score [-].

It varies from \(-\infty\) 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.

\[
RRMSE = \sqrt{\frac{\sum_{t=1}^{n} (X_{sim,t} - X_{ref,t})^2}{n \cdot X_{ref}^2}}
\]

*with* \(RRMSE\): relative RMSE [-]; \(n\): number of values [-].

It varies from 0 to \(+\infty\). 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.

\[
RVB = \frac{\sum_{t=t_i}^{t_f} (X_{sim,t} - X_{ref,t})}{\sum_{t=t_i}^{t_f} (X_{ref,t})}
\]

*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 +\(\infty\). 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.

\[
NPE = \frac{S_{max} - R_{max}}{R_{max}}
\]

\[
S_{max} = \bigvee_{t=t_i}^{t_f} Q_{sim,t}
\]

\[
R_{max} = \bigvee_{t=t_i}^{t_f} Q_{ref,t}
\]

*with NPE: relative error between simulated and observed peak value [-]; S_{max}: maximum simulated value (discharge [L³/T] or height [L]) for the studied period; R_{max}: maximum observed value (discharge [L³/T] or height [L]) for the studied period.*

The NPE varies from -1 to +\(\infty\). 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 defining the number of cases where the simulation and the observation exceed or not the threshold, the PSS is computed according to IND.13.
\[ PSS = \frac{ad - bc}{(a + c)(b + d)} \]  

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 observation (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 observation are below the threshold (nonevent).

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

### 3.10. Overall Accuracy

The Overall Accuracy (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 \( \text{IND.14} \).

\[ OA = \frac{a + d}{a + b + c + d} \]  

\( \text{IND.14} \)
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 satisfactorily 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 Adaptive 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 with the 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

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

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.

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

4.3. Shuffled Complex Evolution – 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.
Chapter 4. Expert module - Calibration algorithms

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.
Chapter 4. Expert module - Calibration algorithms

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](image)

**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.
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 $N_{opt}$ parameters is given by the user or is assumed as random depending on the user’s needs and the used hydrological models. The other $N_{PT}-1$ 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\(\cdot\)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\(\cdot\)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

<table>
<thead>
<tr>
<th>Object</th>
<th>Name</th>
<th>Units</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCE-UA</td>
<td>MAXM</td>
<td>-</td>
<td>Maximum number of iterations</td>
<td>10000</td>
</tr>
<tr>
<td></td>
<td>NGS</td>
<td>-</td>
<td>Number of complexes</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>KSTOP</td>
<td>-</td>
<td>Number of shuffling loops</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>PCENTO</td>
<td>-</td>
<td>Criterion value on shuffling loops</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>PEPS</td>
<td>-</td>
<td>Convergence parameter</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>SEED</td>
<td>-</td>
<td>Seed value</td>
<td>Random</td>
</tr>
</tbody>
</table>

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 \texttt{COEFRANG}.

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

Table 24 Parameters of the UAMC algorithm

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

Figure 17 Flow chart of the UAMC algorithm

\[
\text{New}_{\text{min}} = \min_{\text{numbest}} - (\max_{\text{numbest}} - \min_{\text{numbest}}) \times \text{COEFRANG}, \\
\text{New}_{\text{max}} = \max_{\text{numbest}} + (\max_{\text{numbest}} - \min_{\text{numbest}}) \times \text{COEFRANG}
\]
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.

![Latin Hypercube](image)

**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.
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.

\[
\frac{MaxParameterValue_i - MinParameterValue_i}{STEPROS} = \Delta_i
\]

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

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

![Probability distributions](image)

**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].
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 that 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 it is more probable to find a X value.

Probability density function:

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

with $\mu$: is mean or expectation of the distribution (and also its median and mode); $\sigma$: is variance; $\sigma$: is standard deviation.

Cumulative distribution function:

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

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.

![Log normal distribution. Probability density function (left), cumulative distribution function (right).](image)

**Probability density function:**

$$
\varphi = f(x \mid \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(\frac{-(\ln x - \mu)^2}{2\sigma^2}\right)
$$

with $\mu$: is mean or expectation of the distribution (and also its median and mode); $\sigma^2$: is variance; $\sigma$ is standard deviation.

**Cumulative distribution function:**

$$
\Phi = F(x \mid \mu, \sigma) = \frac{1}{2} \left[1 + \frac{1}{\sqrt{\pi}} \int_{-\infty}^{(\ln x - \mu)/\sigma\sqrt{2}} e^{-t^2} \, dt\right]
$$

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.
Chapter 5. Expert module - Stochastic equations

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

**Probability density function:**

\[ \phi = f(x | \lambda) = \lambda e^{-\lambda x} \]  

with \( \lambda \): is the parameter of the distribution, often called the rate parameter \([\lambda > 0]\).

**Cumulative distribution function:**

\[ \Phi = F(x | \lambda) = 1 - e^{-\lambda x} \]  

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.

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

The Figure 27 shows probability density function for the 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:

```vbscript
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:

```vbscript
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

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>modelFullPath</td>
<td>String. Full path of the hydrological model. The file extension of the model has to be <em>.rsm</em>; otherwise, the script reports an error.</td>
</tr>
<tr>
<td>Debug</td>
<td>Boolean (optional). If TRUE, the script writes a text file with the error logs.</td>
</tr>
</tbody>
</table>
## Errors

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argument Null Exception</td>
<td>“Model file not defined”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Model file not found”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Model file not valid”</td>
</tr>
</tbody>
</table>
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”
**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**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseDatasetFullPath</td>
<td>String. Full path of the base dataset.</td>
</tr>
<tr>
<td>saveDataInDstFile</td>
<td>Boolean. If true the dataset values are saved in text format (<em>.dst), if false in binary format (</em>.dsb).</td>
</tr>
</tbody>
</table>

**Errors**

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid Operation Exception</td>
<td>“Input dataset not loaded, failed to merge with base dataset”</td>
</tr>
<tr>
<td>Argument Null Exception</td>
<td>“Input dataset not defined”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Fullpath of base dataset not valid”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Failed to create new base dataset”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Base dataset not valid”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Failed to merge input dataset with base dataset”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Failed to save changes in base dataset”</td>
</tr>
</tbody>
</table>

**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**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>icFileFullPath</td>
<td>String. Full path of the IC file. Acceptable file extensions are *.txt and *.xlsx.</td>
</tr>
</tbody>
</table>
Chapter 6. Visual Basic Scripts

Errors

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Error Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid Operation Exception</td>
<td>“Model not loaded”</td>
</tr>
<tr>
<td>Argument Null Exception</td>
<td>“IC file not defined”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“IC file not found”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Failed to import initial conditions from IC file”</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>icDatasetFullPath</td>
<td>String. Full path of the dataset from where to extract the initial conditions.</td>
</tr>
</tbody>
</table>

Errors

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Error Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid Operation Exception</td>
<td>“Model not loaded”</td>
</tr>
<tr>
<td>Argument Null Exception</td>
<td>“IC dataset not defined”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“IC dataset not found”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“IC dataset not valid”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Failed to load results from IC dataset”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Failed to initialize the model with IC dataset”</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>icFileFullPath</td>
<td>String. Full path of the IC file. Acceptable file extensions are *.txt and *.xlsx.</td>
</tr>
<tr>
<td>saveFolder</td>
<td>String. Full path of the save folder. If not specified, the current working directory is used.</td>
</tr>
</tbody>
</table>
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

- 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**
- **paramFileFullPath**  
  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”
Chapter 6. Visual Basic Scripts

**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**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>startDate</td>
<td>Date. Start date for the simulation period. The format of the date has to be: dd.MM.yyyy HH:mm:ss.</td>
</tr>
<tr>
<td>endDate</td>
<td>Date. End date for the simulation period. The format of the date has to be: dd.MM.yyyy HH:mm:ss.</td>
</tr>
</tbody>
</table>

**Errors**

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid Operation Exception</td>
<td>“Model not loaded”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Start Date not valid, unable to convert 'startDate' to a Date”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“End Date not valid, unable to convert 'endDate' to a Date”</td>
</tr>
</tbody>
</table>

### SetStartDate

**Setting of the start date**

**Description**

This procedure sets the simulation period start date.

**Syntax**

SetStartDate (startDate)

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>startDate</td>
<td>Date. Start date for the simulation period. The format of the date has to be: dd.MM.yyyy HH:mm:ss.</td>
</tr>
</tbody>
</table>

**Errors**

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid Operation Exception</td>
<td>“Model not loaded”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Start Date not valid, unable to convert 'startDate' to a Date”</td>
</tr>
</tbody>
</table>

### SetEndDate

**Setting of the end date**

**Description**

This procedure sets the simulation period end date.

**Syntax**

SetEndDate (endDate)

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>endDate</td>
<td>Date. End date for the simulation period. The format of the date has to be: dd.MM.yyyy HH:mm:ss.</td>
</tr>
</tbody>
</table>
Chapter 6. Visual Basic Scripts

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. See Table 27.

**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.
<table>
<thead>
<tr>
<th><strong>value</strong></th>
<th>Double. Value of the property.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>zone = null</strong></td>
<td>String (optional). Zone where the objects are.</td>
</tr>
</tbody>
</table>

**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

**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: \( Q = A \times \text{Output} + B \) for the definition of a particular Rule output at a specified Planner.

**Syntax**

\[
\text{SetOutputRatioCoeffsInPlanner (plannerName, ruleName, coefA, coefB)}
\]

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plannerName</td>
<td>String. Name of the Planner object concerned.</td>
</tr>
<tr>
<td>ruleName</td>
<td>String. Name of the Rule (defined within the above Planner) for which the output expression’s coefficients are to be defined.</td>
</tr>
<tr>
<td>coefA</td>
<td>Double. Value of the coefficient A in the output expression ( Q = A \times \text{Output} + B )</td>
</tr>
<tr>
<td>coefB</td>
<td>Double. Value of the coefficient B in the output expression ( Q = A \times \text{Output} + B )</td>
</tr>
</tbody>
</table>

**Errors**

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argument Null Exception</td>
<td>“Planner name not defined”</td>
</tr>
<tr>
<td>Argument Null Exception</td>
<td>“Rule name not defined”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Ratio Coeff. A is not valid, unable to convert to a Double”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Ratio Coeff. B is not valid, unable to convert to a Double”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“No Planner with this name”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“No Rule with this name”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Failed to define Ratio Coefficient in Planner”</td>
</tr>
</tbody>
</table>

---

#### 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.
Chapter 6. Visual Basic Scripts

Syntax

SetConditionThresholdInPlanner (plannerName, ruleName, conditionName, value)

Parameters

<table>
<thead>
<tr>
<th>RegulationName</th>
<th>String. Name of the Planner object concerned.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleName</td>
<td>String. Name of the Rule, defined within the above Planner.</td>
</tr>
<tr>
<td>ConditionName</td>
<td>String. Name of the Condition, defined within the above Rule, for which the threshold is to be defined.</td>
</tr>
<tr>
<td>Value</td>
<td>Double. Value of the threshold of the specified condition.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>PlannerName</th>
<th>String. Name of the Planner object concerned.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleName</td>
<td>String. Name of the Rule, defined within the above Planner.</td>
</tr>
<tr>
<td>Day1, ..., Day7</td>
<td>Boolean. Whether the rule is active for the n° day of the week (day1=Monday, day2=Tuesday...).</td>
</tr>
</tbody>
</table>

Errors

| Argument Null Exception | “Planner name not defined” |
| Argument Null Exception | “Rule name not defined” |
Chapter 6. Visual Basic Scripts

<table>
<thead>
<tr>
<th>Argument Exception</th>
<th>“No Planner with name plannerName”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argument Exception</td>
<td>“No Rule with name ruleName”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Failed to define rule days in Planner”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>SetHoursInPlanner</strong></th>
<th><strong>Setting of the hours of a rule in a specific Planner</strong></th>
</tr>
</thead>
</table>

**Description**

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

**Syntax**

SetHoursInPlanner (plannerName, ruleName, startHour, endHour)

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plannerName</td>
<td>String. Name of the Planner object concerned.</td>
</tr>
<tr>
<td>ruleName</td>
<td>String. Name of the Rule, defined within the above Planner.</td>
</tr>
<tr>
<td>startHour</td>
<td>String. Initial hour of the period for which the rule is active.</td>
</tr>
<tr>
<td>endHour</td>
<td>String. Final hour of the period for which the rule is active.</td>
</tr>
</tbody>
</table>

**Errors**

<table>
<thead>
<tr>
<th>Error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argument Null Exception</td>
<td>“Planner name not defined”</td>
</tr>
<tr>
<td>Argument Null Exception</td>
<td>“Rule name not defined”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“No Planner with name plannerName”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“No Rule with name ruleName”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Failed to define hours in Planner”</td>
</tr>
</tbody>
</table>
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.
- *showInitialDateInFileName*: Boolean (optional). Adds the StartDate of the simulation to the file name if True (e.g. *yyyyMMddHH_referenceName.txt*).

**Errors**
- **Invalid Operation Exception**: “Model not loaded”
- **Argument Null Exception**: “Reference name not defined”
Chapter 6. Visual Basic Scripts

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

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SaveModelAs</strong></td>
<td>Saving of the model</td>
</tr>
</tbody>
</table>

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

**Syntax**

```vbs
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”

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SaveModel</strong></td>
<td>Saving of the model</td>
</tr>
</tbody>
</table>

This procedure saves the loaded model at its current location.

**Syntax**

```vbs
SaveModel ()
```

**Errors**

- **Invalid Operation Exception**: “Model not loaded”
- **Invalid Operation Exception**: “Failed to save model”

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SaveInitialConditionsAs</strong></td>
<td>Saving of the initial conditions</td>
</tr>
</tbody>
</table>

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**

```vbs
SaveInitialConditionsAs (initialConditionsFullPath)
```
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”

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

**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

**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”
<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argument Null Exception</td>
<td>“Save folder not defined”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Save folder 'saveFolder' not valid”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Failed to save parameters file”</td>
</tr>
</tbody>
</table>
6.7. Model results

<table>
<thead>
<tr>
<th><strong>SaveFullResultsAs</strong></th>
<th>Saving of all the results</th>
</tr>
</thead>
</table>

**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”

<table>
<thead>
<tr>
<th><strong>SaveFullResults</strong></th>
<th>Saving of all the results</th>
</tr>
</thead>
</table>

**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.
Chapter 6. Visual Basic Scripts

**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**

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid Operation Exception</td>
<td>“Model not loaded”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Model 'hydrologicalmodel' has no results”</td>
</tr>
<tr>
<td>Argument Null Exception</td>
<td>“Reference name not defined”</td>
</tr>
<tr>
<td>Argument Null Exception</td>
<td>“Save folder not defined”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Save folder 'saveFolder' not valid”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Failed to save full result”</td>
</tr>
</tbody>
</table>

**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**

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid Operation Exception</td>
<td>“Model not loaded”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Model 'hydrologicalmodel' has no results”</td>
</tr>
<tr>
<td>Argument Null Exception</td>
<td>“Fullpath of selection results dataset not defined”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Fullpath of selection results dataset contains invalid characters”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Filename extension of selection results dataset not valid”</td>
</tr>
</tbody>
</table>
Chapter 6. Visual Basic Scripts

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argument Exception</td>
<td>“Fullpath of selection results dataset not rooted”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Selection 'selectionname' not valid”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Failed to create and save selection results dataset”</td>
</tr>
</tbody>
</table>

### SaveSelectionResults

**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**

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

**Parameters**

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

**Errors**

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid Operation Exception</td>
<td>“Model not loaded”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Model 'hydrologicalmodel' has no results”</td>
</tr>
<tr>
<td>Argument Null Exception</td>
<td>“Reference name not defined”</td>
</tr>
<tr>
<td>Argument Null Exception</td>
<td>“Save folder not defined”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Save folder 'saveFolder' not valid”</td>
</tr>
<tr>
<td>Argument Null Exception</td>
<td>“Selection not defined”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Selection 'selectionName' not valid”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Failed to save selection results”</td>
</tr>
</tbody>
</table>
### 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**

```vba
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**

```vba
SaveInitializationDataset (resultsDatasetFullPath, saveDataInDstFile)
```

**Parameters**

- **resultsDatasetFullPath**  String. Dataset path.
- **saveDataInDstFile**  Boolean. If true the dataset values are saved in text format (*.dst), if false in binary format (*.dsb).
Chapter 6. Visual Basic Scripts

<table>
<thead>
<tr>
<th>resultsDatasetFullPath</th>
<th>String. Full path of the file where is the full results dataset. The file extension has to be *.dsx.</th>
</tr>
</thead>
<tbody>
<tr>
<td>saveDataInDstFile</td>
<td>Boolean. If true the dataset values are saved in text format (<em>.dst), if false in binary format (</em>.dsb).</td>
</tr>
</tbody>
</table>

**Errors**

<table>
<thead>
<tr>
<th>Invalid Operation Exception</th>
<th>“Model not loaded”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid Operation Exception</td>
<td>“Model has no results”</td>
</tr>
<tr>
<td>Argument Null Exception</td>
<td>“Fullpath of full results dataset not defined”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Fullpath of full results dataset contains invalid characters”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Filename extension of full results dataset not valid”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Fullpath of full results dataset not rooted”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Failed to create and save full results dataset”</td>
</tr>
</tbody>
</table>

**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**

<table>
<thead>
<tr>
<th>selectionResultsFileFullPath</th>
<th>String. Full path of the selection results file. Acceptable file extensions are *.csv and *.xlsx.</th>
</tr>
</thead>
<tbody>
<tr>
<td>selectionName</td>
<td>String. Name of the selection of results.</td>
</tr>
</tbody>
</table>

**Errors**

<table>
<thead>
<tr>
<th>Invalid Operation Exception</th>
<th>“Model not loaded”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid Operation Exception</td>
<td>“Model 'hydrologicalmodel' has no results”</td>
</tr>
<tr>
<td>Argument Null Exception</td>
<td>“Fullpath not defined”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Fullpath of selection results contains invalid characters”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Fullpath of selection results dataset not rooted”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Selection 'selectionName' not valid”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Fullpath extension not valid”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Failed to create and save selection results”</td>
</tr>
</tbody>
</table>
**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: “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**
- selectionResultsDatasetFullPath: 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”

<table>
<thead>
<tr>
<th><strong>MergeDatasetsAs</strong></th>
<th><strong>Merging of a source dataset with another dataset</strong></th>
</tr>
</thead>
</table>

**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
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**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>objectType</td>
<td>String. Object type (e.g. SOCONT, GSM, Reservoir, etc.). See Table 27.</td>
</tr>
<tr>
<td>propertyName</td>
<td>String. Property name (e.g. HGR3Max). See Table 27.</td>
</tr>
<tr>
<td>zone</td>
<td>String (optional). Zone where the objects are.</td>
</tr>
</tbody>
</table>

**Returned value**

property value

It returns the property value (parameter or initial condition) of the object.

**Errors**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Error Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid Operation Exception</td>
<td>“Model not loaded”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Object type 'objectType' not valid”</td>
</tr>
<tr>
<td>Argument Null Exception</td>
<td>“Property name not defined”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Zone 'zone' not found in 'hydrologicalmodel'”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Failed to provide model property”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“Model property not found”</td>
</tr>
</tbody>
</table>

### 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**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>indicatorName</td>
<td>String. Indicator name (e.g. HGR3Max)</td>
</tr>
<tr>
<td>objectName</td>
<td>String. Object name (e.g. SOCONT)</td>
</tr>
</tbody>
</table>
### Chapter 6. Visual Basic Scripts

<table>
<thead>
<tr>
<th><strong>indicatorName</strong></th>
<th>String. Indicator name. <strong>Indicators:</strong> Nash, Nash-ln, Pearson Correlation Coeff, Kling-Gupta Efficiency, Bias Score, RRMSE, Relative Volume Bias, Normalized Peak Error</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>objectName</strong></td>
<td>String. User-defined object name.</td>
</tr>
</tbody>
</table>

**Returned value**

| **indicator value** | It returns the value of the indicator.                                                                                                                                                     |

**Errors**

<table>
<thead>
<tr>
<th><strong>Invalid Operation Exception</strong></th>
<th>“Model not loaded”</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Argument Exception</strong></td>
<td>“Comparator 'objectName' not found”</td>
</tr>
<tr>
<td><strong>Argument Exception</strong></td>
<td>“Indicator name 'indicatorName' not valid”</td>
</tr>
</tbody>
</table>
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

### 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

### 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

### Description
This procedure exports the results of a specified calibration, to a `.txt` file.
## Syntax

### ExportCalibrationResults (configName, resultFilePath)

#### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>configName</td>
<td>String. Name given in the Calibrator module of RS MINERVE to the calibration configuration to be executed.</td>
</tr>
<tr>
<td>resultFilePath</td>
<td>String. Full path of the file where to save the calibration results. The file extension has to be *.txt.</td>
</tr>
</tbody>
</table>

#### Errors

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Error Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argument Null Exception</td>
<td>“The name of the calibration configuration is not defined”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“The file path for the results is not defined”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“No calibration configuration called configName”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“There are no results to export in calibration configName”</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resultFolderPath</td>
<td>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.</td>
</tr>
</tbody>
</table>

#### Errors

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Error Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argument Null Exception</td>
<td>“The folder path for the results is not defined”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“Selected folder does not exist”</td>
</tr>
<tr>
<td>Invalid Operation Exception</td>
<td>“There are no results to export for some calibrations”</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>configName</th>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td>startDate</td>
<td>Date. Start date for the calibration period. The format of the date has to be: dd.MM.yyyy HH:mm:ss.</td>
</tr>
<tr>
<td>endDate</td>
<td>Date. End date for the calibration period. The format of the date has to be: dd.MM.yyyy HH:mm:ss.</td>
</tr>
</tbody>
</table>

### Errors

<table>
<thead>
<tr>
<th>Error</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid Operation Exception</td>
<td>“Model not loaded”</td>
</tr>
<tr>
<td>Argument Null Exception</td>
<td>“The name of the calibration configuration is not defined”</td>
</tr>
<tr>
<td>Argument Exception</td>
<td>“No calibration configuration called configName”</td>
</tr>
</tbody>
</table>
6.9. **Application Stop**

<table>
<thead>
<tr>
<th>Stop</th>
<th>Stop of the execution</th>
</tr>
</thead>
</table>

**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 removes 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>
<thead>
<tr>
<th>Object class</th>
<th>Names to be used in VBScripts</th>
<th>parameters</th>
<th>Initial conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Station</td>
<td>X, Y, Z, Search Radius, No. min. of stations, Gradient P, Gradient T, Gradient ETP, Coeff P, Coeff T, Coeff ETP</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SnowSD</td>
<td>S, SInt, SMin, SPh, ThetaCri, bp, Tcp1, Tcp2, Tcf</td>
<td>SWEIni, ThetaIni</td>
<td>-</td>
</tr>
<tr>
<td>SwMM</td>
<td>A, L, J0, K</td>
<td>HIni</td>
<td>-</td>
</tr>
<tr>
<td>GSM</td>
<td>A, S, SInt, SMin, SPh, ThetaCri, bp, Tcp1, Tcp2, Tcf, G, GInt, GMin, Tcg, Kg1, Ksn</td>
<td>SWEIni, ThetaIni, QsnowIni, QglacierIni, HglacierIni</td>
<td>-</td>
</tr>
<tr>
<td>SOCONT</td>
<td>A, S, SInt, SMin, SPh, ThetaCri, bp, Tcp1, Tcp2, Tcf, G, GInt, GMin, Tcg, Kg1, Ksn</td>
<td>SWEIni, ThetaIni, HGR3Ini, HrIni</td>
<td>-</td>
</tr>
<tr>
<td>HBV</td>
<td>A, CFMax, CFR, CWH, TT, TTInt, TTSM, Beta, FC, PWP, SUMax, Kr, Ku, Kl, Kperc</td>
<td>SWEIni, WHIni, HumIni, SUIni, SLIni</td>
<td>-</td>
</tr>
<tr>
<td>GR4J</td>
<td>A, X1, X2, X3, X4</td>
<td>SIni, RIni</td>
<td>-</td>
</tr>
<tr>
<td>SAC</td>
<td>A, Adimp, Pctim, Riva, UztwMax, UzfWMax, Uzk, Zperc, Rexp, Pfree, LztwMax, LzfWMax, LzfsMax, Rserv, Lzp, Lzsk, Side</td>
<td>AdimIni, UztwIni, UzfWIni, LztwIni, LzfWIni, LzfsIni</td>
<td>-</td>
</tr>
<tr>
<td>River</td>
<td>L, B0, m, J0, K, lag</td>
<td>QIni</td>
<td>-</td>
</tr>
<tr>
<td>Kinematic</td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>MuskingumCunge</td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>StVenant</td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Reservoir</td>
<td></td>
<td>HIni, HvarIni</td>
<td>-</td>
</tr>
<tr>
<td>Hydropower</td>
<td>Zplant, L, D, K, v, Default Price</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Consumer</td>
<td>Default QDemand, LossRate, ConsumptionRate</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>StructureEfficiency</td>
<td>Efficiency</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
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.

![Figure 28 Example of the parameters of the GSM hydrological models in an Excel file](image)

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.

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,)
  - Contains one or more datasets

- **Dataset**
  - Set of data of common type (Meteo data, Flow data,...)
  - 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

<table>
<thead>
<tr>
<th><strong>Node</strong></th>
<th><strong>Name</strong></th>
<th><strong>Description</strong></th>
<th><strong>Example</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td><strong>Description</strong></td>
<td>Name of the database</td>
<td><strong>Database_tutorial</strong></td>
</tr>
<tr>
<td>Date creation</td>
<td>Date of creation of database</td>
<td>14.12.2012 13:32</td>
<td></td>
</tr>
<tr>
<td>Date modification</td>
<td>Date of last modification</td>
<td>04.01.2013 14:30</td>
<td></td>
</tr>
<tr>
<td><strong>Time zone</strong></td>
<td><strong>Date scale of the data</strong></td>
<td>UTC+01:00</td>
<td></td>
</tr>
<tr>
<td>Filename</td>
<td>File name and path</td>
<td>(Full path of file)</td>
<td></td>
</tr>
<tr>
<td>XML file</td>
<td>XML file size</td>
<td>21 Ko</td>
<td></td>
</tr>
<tr>
<td>Data file</td>
<td>Data file size</td>
<td>161 Ko</td>
<td></td>
</tr>
<tr>
<td><strong>Group</strong></td>
<td><strong>Description</strong></td>
<td>Name of the group</td>
<td><strong>Measure</strong></td>
</tr>
<tr>
<td><strong>Category</strong></td>
<td><strong>Type of data (Inputs, Outputs or None)</strong></td>
<td><strong>Inputs</strong></td>
<td></td>
</tr>
<tr>
<td>Date capture</td>
<td>Date of last dataset capture date in the group</td>
<td>18/12/2012 10:54</td>
<td></td>
</tr>
<tr>
<td><strong>Dataset</strong></td>
<td><strong>Description</strong></td>
<td>Name of the dataset</td>
<td><strong>Meteo_data</strong></td>
</tr>
<tr>
<td>Dataset creation</td>
<td>Dataset creation date</td>
<td>14.12.2012 00:00</td>
<td></td>
</tr>
<tr>
<td>Dataset reception</td>
<td>Date of receipt of data</td>
<td>14.12.2012 03:00</td>
<td></td>
</tr>
<tr>
<td>Data modification</td>
<td>Date of last modification</td>
<td>14.12.2012 03:00</td>
<td></td>
</tr>
<tr>
<td><strong>Data capture</strong></td>
<td><strong>First date of the data (editable)</strong></td>
<td>14.12.2012 00:00</td>
<td></td>
</tr>
<tr>
<td>Time zone</td>
<td>Tine scale of the data</td>
<td>UTC+01:00</td>
<td></td>
</tr>
<tr>
<td>Filename</td>
<td>File name and path</td>
<td>(Full path of file)</td>
<td></td>
</tr>
<tr>
<td>XML file</td>
<td>XML file size</td>
<td>21 Ko</td>
<td></td>
</tr>
<tr>
<td>Data file</td>
<td>Data file size</td>
<td>161 Ko</td>
<td></td>
</tr>
<tr>
<td><strong>Station</strong></td>
<td><strong>Name</strong></td>
<td>Name of the station</td>
<td><strong>Station_meteo_1</strong></td>
</tr>
<tr>
<td><strong>X</strong></td>
<td>Coordinate X in meter</td>
<td>650000</td>
<td></td>
</tr>
<tr>
<td><strong>Y</strong></td>
<td>Coordinate Y in meter</td>
<td>130000</td>
<td></td>
</tr>
<tr>
<td><strong>Z</strong></td>
<td>Altitude in meter above sea level</td>
<td>1957</td>
<td></td>
</tr>
<tr>
<td><strong>Sensor</strong></td>
<td><strong>Description</strong></td>
<td>Name of the variable</td>
<td><strong>Precipitation</strong></td>
</tr>
<tr>
<td><strong>Category</strong></td>
<td><strong>Variable type</strong></td>
<td><strong>Precipitation</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Unit</strong></td>
<td><strong>Variable unit</strong></td>
<td><strong>MillimetersPerHour</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Interpolation</strong></td>
<td><strong>Data assimilation method</strong></td>
<td><strong>Linear</strong></td>
<td></td>
</tr>
<tr>
<td>Initial date</td>
<td>The first date of the time series</td>
<td>01/01/2011 00:00:00</td>
<td></td>
</tr>
<tr>
<td>Final date</td>
<td>The last date of the time series</td>
<td>31/12/2013 00:00:00</td>
<td></td>
</tr>
<tr>
<td>Min value</td>
<td>Minimum value of the series</td>
<td>0 m³/s</td>
<td></td>
</tr>
<tr>
<td>Average value</td>
<td>Average value of the series</td>
<td>1.1884 m³/s</td>
<td></td>
</tr>
<tr>
<td>Max value</td>
<td>Maximum value of the series</td>
<td>3.2485 m³/s</td>
<td></td>
</tr>
</tbody>
</table>

---

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), 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.
Chapter 8. Database formats

* `.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.

**xml file**

```xml
<Database>
  <DateCreated>2013-02-22T12:26:47.5426565+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>Databasetutorial</Name>
  <LastModifiedDate>2013-02-22T12:27:16.0912894+01:00</LastModifiedDate>
  <Groups>
    <Group>
      <Key>ad506514-af29-467c-9eef-4fa788c038c8</Key>
      <Name>Group1</Name>
      <DataSets>
        <DataSet>
          <Version>4.001</Version>
          <Key/>
          <ID/>
          <LastComputer/>
          <LastUser/>
          <Name>Dataset1</Name>
          <LastModifiedDate>2013-02-22T12:27:04.4256221+01:00</LastModifiedDate>
          <DateReceived>2013-02-22T12:27:04.4356227+01:00</DateReceived>
          <LastDateModified>2013-02-22T12:27:16.0912894+01:00</LastDateModified>
          <DateCapture>2013-02-22T12:27:04.4256221+01:00</DateCapture>
          <Stations>
            <Station>
              <Key/>
              <Name>EVO</Name>
              <Sensors>
                <Sensor>
                  <Type>Inputs</Type>
                  <Name>P</Name>
                  <Category>Precipitation</Category>
                  <Unit>MillimetersPerHour</Unit>
                  <InterpolationMode>Linear</InterpolationMode>
                </Sensor>
                <Sensor>
                  <Type>Inputs</Type>
                  <Name>T</Name>
                  <Category>Temperature</Category>
                  <Unit>DegreeCelsius</Unit>
                  <InterpolationMode>Linear</InterpolationMode>
                </Sensor>
              </Sensors>
            </Station>
          </Stations>
        </DataSet>
      </DataSets>
    </Group>
  </Groups>
</Database>
```

**Figure 31** Example of the database XML file
Chapter 8. Database formats

**txt file**

```
Group\Dataset\Station\Sensor
27.11.2012 00:00:00 0
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
Group\Dataset\Semester
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
```

*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

```xml
<DataSet>
  <Version>4.001</Version>
  <Key></Key>
  <ID></ID>
  <LastComputer/>
  <LastUser/>
  <Name>Dataset</Name>
  <DateCreated>2014-04-15T11:15:22.3963143+02:00</DateCreated>
  <DateReceived>2014-04-15T11:22.3963343+02:00</DateReceived>
  <LastDateModified>2014-04-15T11:28.9044057+02:00</LastDateModified>
  <DateCapture>2014-04-15T11:22.3973144+01:00</DateCapture>
  <Stations>
    <Station>
      <Key></Key>
      <Name>EVO</Name>
      <Sensors>
        <Sensor>
          <Key></Key>
          <Type>Inputs</Type>
          <Name>P</Name>
          <Category>Precipitation</Category>
          <Unit>Millimeters Per Hour</Unit>
          <InterpolationMode>Linear</InterpolationMode>
        </Sensor>
        <Sensor>
          <Key></Key>
          <Type>Inputs</Type>
          <Name>T</Name>
          <Category>Temperature</Category>
          <Unit>Degree Celsius</Unit>
          <InterpolationMode>Linear</InterpolationMode>
        </Sensor>
      </Sensors>
    </Station>
    <Station>
      <Key></Key>
      <Name>AIG</Name>
      <Sensors>
        <Sensor>
          <Key></Key>
          <Type>Inputs</Type>
          <Name>P</Name>
          <Category>Precipitation</Category>
          <Unit>Millimeters Per Hour</Unit>
          <InterpolationMode>Linear</InterpolationMode>
        </Sensor>
        <Sensor>
          <Key></Key>
          <Type>Inputs</Type>
          <Name>T</Name>
          <Category>Temperature</Category>
          <Unit>Degree Celsius</Unit>
          <InterpolationMode>Linear</InterpolationMode>
        </Sensor>
      </Sensors>
    </Station>
  </Stations>
</DataSet>
```

**Figure 34** Example of the dataset XML file
**txt file**

<table>
<thead>
<tr>
<th>Station</th>
<th>Sensor</th>
<th>Date (JJ.MM.YYYY hh:mm:ss)</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVO</td>
<td>P</td>
<td>27.11.2012 00:00</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 01:00</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 02:00</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 03:00</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 04:00</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 05:00</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 06:00</td>
<td>0</td>
</tr>
<tr>
<td>EVO</td>
<td>T</td>
<td>27.11.2012 00:00</td>
<td>2.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 01:00</td>
<td>5.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 02:00</td>
<td>4.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 03:00</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 04:00</td>
<td>4.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 05:00</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 06:00</td>
<td>0.6</td>
</tr>
<tr>
<td>AIG</td>
<td>P</td>
<td>27.11.2012 00:00:00</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 01:00:00</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 02:00:00</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 03:00:00</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 04:00:00</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 05:00:00</td>
<td>0.9</td>
</tr>
<tr>
<td>AIG</td>
<td>T</td>
<td>27.11.2012 00:00:00</td>
<td>8.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 01:00:00</td>
<td>8.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 02:00:00</td>
<td>8.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 03:00:00</td>
<td>7.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 04:00:00</td>
<td>7.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 05:00:00</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.11.2012 06:00:00</td>
<td>6.3</td>
</tr>
</tbody>
</table>

**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.
Chapter 8. Database formats

Figure 36 Structure of CSV file in RSMINERVE.

And Figure 37 shows an example of CSV file:
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

<table>
<thead>
<tr>
<th>Node</th>
<th>Required</th>
<th>Suggested</th>
<th>Example</th>
<th>Purpose</th>
<th>Visible</th>
<th>Editable</th>
<th>Non editable</th>
<th>Visible</th>
<th>Non visible</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataBase</td>
<td>X</td>
<td></td>
<td>&lt;Database&gt;2014-02-05T17:10:10.1625449+01:00&lt;/Database&gt;</td>
<td>Starts the database node</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DateCreated</td>
<td>X</td>
<td></td>
<td>&lt;DateCreated&gt;2014-02-05T17:10:10.1625449+01:00&lt;/DateCreated&gt;</td>
<td>Database description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Version</td>
<td>X</td>
<td></td>
<td>&lt;Version&gt;4.001&lt;/Version&gt;</td>
<td>To verify RS Database version</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>X</td>
<td></td>
<td>&lt;Key&gt;22348d22-767-4d77-af95-564f029a46e6&lt;/Key&gt;</td>
<td>To distinguish different databases</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>X</td>
<td></td>
<td>&lt;ID&gt;d1fa701-0e05-4833-847e-4188a8b808f1&lt;/ID&gt;</td>
<td>To distinguish databases with identical Key</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LastComputer</td>
<td>X</td>
<td></td>
<td>&lt;LastComputer&gt;D29493&lt;/LastComputer&gt;</td>
<td>Computer origin</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LastUser</td>
<td>X</td>
<td></td>
<td>&lt;LastUser&gt;D29493&lt;/LastUser&gt;</td>
<td>User</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>X</td>
<td></td>
<td>&lt;Name&gt;Database&lt;/Name&gt;</td>
<td>Database description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LastDateModified</td>
<td>X</td>
<td></td>
<td>&lt;LastDateModified&gt;2014-04-10T15:31:46.2862121+02:00&lt;/LastDateModified&gt;</td>
<td>Database description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TimeOffset</td>
<td>X</td>
<td></td>
<td>&lt;TimeOffset&gt;360000000000&lt;/TimeOffset&gt;</td>
<td>Database description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Groups</td>
<td>X</td>
<td></td>
<td>&lt;Groups&gt;</td>
<td>Starts the groups node</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group</td>
<td>X</td>
<td></td>
<td>&lt;Group&gt;</td>
<td>Starts the description of a group</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>X</td>
<td></td>
<td>&lt;Key&gt;763d446-5eb7-4d75-8d2a-e84c79abebf9&lt;/Key&gt;</td>
<td>To distinguish groups in the database</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>X</td>
<td></td>
<td>&lt;Name&gt;Measure&lt;/Name&gt;</td>
<td>Group description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DataSets</td>
<td>X</td>
<td></td>
<td>&lt;DataSets&gt;</td>
<td>Starts the datasets node</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dataset</td>
<td>X</td>
<td></td>
<td>&lt;Dataset&gt;</td>
<td>Starts the description of a dataset</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Version</td>
<td>X</td>
<td></td>
<td>&lt;Version&gt;4.001&lt;/Version&gt;</td>
<td>To verify RS Database version</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>X</td>
<td></td>
<td>&lt;Key&gt;a3e0e759-6f10-4a93-8aa1-cb213ac767f1&lt;/Key&gt;</td>
<td>To distinguish different datasets</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>X</td>
<td></td>
<td>&lt;ID&gt;b26e741977c6ec&lt;/ID&gt;</td>
<td>To distinguish databases with identical Key</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LastComputer</td>
<td>X</td>
<td></td>
<td>&lt;LastComputer&gt;vmcrealp&lt;/LastComputer&gt;</td>
<td>Computer origin</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LastUser</td>
<td>X</td>
<td></td>
<td>&lt;LastUser&gt;vmcrealp&lt;/LastUser&gt;</td>
<td>User</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>X</td>
<td></td>
<td>&lt;Name&gt;201404090700_Measure&lt;/Name&gt;</td>
<td>Dataset description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DateCreated</td>
<td>X</td>
<td></td>
<td>&lt;DateCreated&gt;2014-04-10T10:00:12+02:00&lt;/DateCreated&gt;</td>
<td>Dataset description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DateRecept</td>
<td>X</td>
<td></td>
<td>&lt;DateRecept&gt;10T10:00:12&lt;/DateRecept&gt;</td>
<td>Dataset description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LastDateModified</td>
<td>X</td>
<td></td>
<td>&lt;LastDateModified&gt;2014-04-10T15:31:46.2862121+02:00&lt;/LastDateModified&gt;</td>
<td>Dataset description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DateCapture</td>
<td>X</td>
<td></td>
<td>&lt;DateCapture&gt;2014-04-09T07:00:00.000000000+02:00&lt;/DateCapture&gt;</td>
<td>Dataset description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stations</td>
<td>X</td>
<td></td>
<td>&lt;Stations&gt;</td>
<td>Starts the stations node</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Station</td>
<td>X</td>
<td></td>
<td>&lt;Station&gt;</td>
<td>Starts the description of a station</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>X</td>
<td></td>
<td>&lt;Key&gt;AIG&lt;/Key&gt;</td>
<td>To distinguish stations in the database</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>X</td>
<td></td>
<td>&lt;Name&gt;Aigle&lt;/Name&gt;</td>
<td>Station description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensors</td>
<td>X</td>
<td></td>
<td>&lt;Sensors&gt;</td>
<td>Starts the sensors node</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensor</td>
<td>X</td>
<td></td>
<td>&lt;Sensor&gt;</td>
<td>Starts the description of a sensor</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>X</td>
<td></td>
<td>&lt;Key&gt;AIG(P)/Key&lt;/Key&gt;</td>
<td>To distinguish sensors in the database</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>X</td>
<td></td>
<td>&lt;Type&gt;Inputs&lt;/Type&gt;</td>
<td>Type of data</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>X</td>
<td></td>
<td>&lt;Name&gt;(P)/Name&lt;/Name&gt;</td>
<td>Sensor description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>X</td>
<td></td>
<td>&lt;Category&gt;Precipitation&lt;/Category&gt;</td>
<td>Sensor description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit</td>
<td>X</td>
<td></td>
<td>&lt;Unit&gt;MillimetersPerHour&lt;/Unit&gt;</td>
<td>Sensor description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>InterpolationMode</td>
<td>X</td>
<td></td>
<td>&lt;InterpolationMode&gt;ConstantBefore&lt;/InterpolationMode&gt;</td>
<td>Sensor description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensor</td>
<td>X</td>
<td></td>
<td>&lt;Sensor&gt;</td>
<td>Finish the description of a sensor</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensors</td>
<td>X</td>
<td></td>
<td>&lt;Sensors&gt;</td>
<td>Finish the sensors node</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td></td>
<td>&lt;X&gt;560402&lt;/X&gt;</td>
<td>Station description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>X</td>
<td></td>
<td>&lt;Y&gt;13073&lt;/Y&gt;</td>
<td>Station description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>X</td>
<td></td>
<td>&lt;Z&gt;3815&lt;/Z&gt;</td>
<td>Station description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Station</td>
<td>X</td>
<td></td>
<td>&lt;Station&gt;</td>
<td>Finish the description of a station</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stations</td>
<td>X</td>
<td></td>
<td>&lt;Stations&gt;</td>
<td>Finish the stations node</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DataSets</td>
<td>X</td>
<td></td>
<td>&lt;DataSets&gt;</td>
<td>Finish the datasets node</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dataset</td>
<td>X</td>
<td></td>
<td>&lt;Dataset&gt;</td>
<td>Finish the description of a dataset</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>X</td>
<td></td>
<td>&lt;Category&gt;Outputs&lt;/Category&gt;</td>
<td>Group description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group</td>
<td>X</td>
<td></td>
<td>&lt;Group&gt;</td>
<td>Finish the description of a group</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Groups</td>
<td>X</td>
<td></td>
<td>&lt;Groups&gt;</td>
<td>Finish the groups node</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TimeOffset</td>
<td>X</td>
<td></td>
<td>&lt;TimeOffset&gt;3600000000000&lt;/TimeOffset&gt;</td>
<td>Database description</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database</td>
<td>X</td>
<td></td>
<td>&lt;Database&gt;</td>
<td>Finish the database node</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 30 Description of all the possible nodes in a dataset XML file

<table>
<thead>
<tr>
<th>Node</th>
<th>Required</th>
<th>Suggested</th>
<th>Example</th>
<th>Purpose</th>
<th>Visible</th>
<th>Editable</th>
<th>Non editable</th>
<th>Non visible</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataSets</td>
<td>X</td>
<td></td>
<td><code>&lt;DataSets&gt;</code></td>
<td>Starts the datasets node</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DataSet</td>
<td>X</td>
<td></td>
<td><code>&lt;DataSet&gt;</code></td>
<td>Starts the description of a dataset</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Version</td>
<td>X</td>
<td></td>
<td><code>&lt;Version&gt;</code>4.0.01&lt;/Version&gt;</td>
<td>To verify RS Database version</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>X</td>
<td></td>
<td><code>&lt;Key&gt;a3e0e759-6f10-4a93-8aa1-d0215e9d71f&lt;/Key&gt;</code></td>
<td>To distinguish different datasets</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>X</td>
<td></td>
<td><code>&lt;ID&gt;a0a3179-3eb0-47ab-892-580e471977c&lt;/ID&gt;</code></td>
<td>To distinguish datasets with identical Key</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LastComputer</td>
<td>X</td>
<td></td>
<td><code>&lt;LastComputer&gt;vmcrealp&lt;/LastComputer&gt;</code></td>
<td>Computer origin</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LastUser</td>
<td>X</td>
<td></td>
<td><code>&lt;LastUser&gt;crealp&lt;/LastUser&gt;</code></td>
<td>User</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>X</td>
<td></td>
<td><code>&lt;Name&gt;201404090700_Measure&lt;/Name&gt;</code></td>
<td>Dataset description</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DateCreated</td>
<td>X</td>
<td></td>
<td><code>&lt;DateCreated&gt;2014-04-10T00:12:02.000&lt;/DateCreated&gt;</code></td>
<td>Dataset description</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DateRecept</td>
<td>X</td>
<td></td>
<td><code>&lt;DateRecept&gt;2014-04-10T15:29:36.0541902&lt;/DateRecept&gt;</code></td>
<td>Dataset description</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LastDateModified</td>
<td>X</td>
<td></td>
<td><code>&lt;LastDateModified&gt;2014-04-10T15:29:36.0531901&lt;/LastDateModified&gt;</code></td>
<td>Dataset description</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DateCapture</td>
<td>X</td>
<td></td>
<td><code>&lt;DateCapture&gt;2014-04-09T07:00:00.0000000&lt;/DateCapture&gt;</code></td>
<td>Dataset description</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stations</td>
<td>X</td>
<td></td>
<td><code>&lt;Stations&gt;</code></td>
<td>Starts the stations node</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Station</td>
<td>X</td>
<td></td>
<td><code>&lt;Station&gt;</code></td>
<td>Starts the description of a station</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>X</td>
<td></td>
<td><code>&lt;Key&gt;AIG&lt;/Key&gt;</code></td>
<td>To distinguish stations in the database</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>X</td>
<td></td>
<td><code>&lt;Name&gt;Aigle&lt;/Name&gt;</code></td>
<td>Station description</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensors</td>
<td>X</td>
<td></td>
<td><code>&lt;Sensors&gt;</code></td>
<td>Starts the sensors node</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensor</td>
<td>X</td>
<td></td>
<td><code>&lt;Sensor&gt;</code></td>
<td>Starts the description of a sensor in the database</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>X</td>
<td></td>
<td><code>&lt;Key&gt;AIG\P&lt;/Key&gt;</code></td>
<td>To distinguish sensors in the database</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>X</td>
<td></td>
<td><code>&lt;Name&gt;\P&lt;/Name&gt;</code></td>
<td>Sensor description</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>X</td>
<td></td>
<td><code>&lt;Category&gt;Precipitation&lt;/Category&gt;</code></td>
<td>Sensor description</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit</td>
<td>X</td>
<td></td>
<td><code>&lt;Unit&gt;MillimetersPerHour&lt;/Unit&gt;</code></td>
<td>Sensor description</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>InterpolationMode</td>
<td>X</td>
<td></td>
<td><code>&lt;InterpolationMode&gt;ConstantBefore&lt;/InterpolationMode&gt;</code></td>
<td>Sensor description</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensor</td>
<td>X</td>
<td></td>
<td><code>&lt;Sensor&gt;</code></td>
<td>Finish the description of a sensor in the database</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensors</td>
<td>X</td>
<td></td>
<td><code>&lt;Sensors&gt;</code></td>
<td>Finish the sensors node</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td></td>
<td><code>&lt;X&gt;560401&lt;/X&gt;</code></td>
<td>Station description</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>X</td>
<td></td>
<td><code>&lt;Y&gt;130713&lt;/Y&gt;</code></td>
<td>Station description</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>X</td>
<td></td>
<td><code>&lt;Z&gt;381&lt;/Z&gt;</code></td>
<td>Station description</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Station</td>
<td>X</td>
<td></td>
<td><code>&lt;Station&gt;</code></td>
<td>Finish the description of a station</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stations</td>
<td>X</td>
<td></td>
<td><code>&lt;Stations&gt;</code></td>
<td>Finish the stations node</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DataSet</td>
<td>X</td>
<td></td>
<td><code>&lt;DataSet&gt;</code></td>
<td>Finish the description of a dataset</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DataSets</td>
<td>X</td>
<td></td>
<td><code>&lt;DataSets&gt;</code></td>
<td>Finish the datasets node</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Formats</th>
<th>Extensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vector</td>
<td>*.shp</td>
</tr>
<tr>
<td>Raster</td>
<td>*.bgd</td>
</tr>
<tr>
<td>DotSpatial.Shapefile</td>
<td>*.shp</td>
</tr>
<tr>
<td>DotSpatial – Binary file</td>
<td>*.bgd</td>
</tr>
<tr>
<td>DotNet Image - Bitmap</td>
<td>*.bmp</td>
</tr>
</tbody>
</table>
Bibliography


Bibliography


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 Micheloud¹, Néstor Lerma², Alex Dionisio-Calado¹, Nicolas Rey³, Edgar Belda², Samuel Alesina¹, Aurélien Claude¹ and Javier Fluixá Sanmartín¹.

¹ Centre de recherche sur l’environnement alpin - CREALP
² Université Polytechnique de Valence - UPV
³ HydroCosmos SA
Appendix

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**

<table>
<thead>
<tr>
<th>Warning</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td>No precipitation found within search radius of virtual station 'Station 1'. Radius extended to satisfy No. min of stations.</td>
</tr>
</tbody>
</table>

**Snow-SD model description**

**Pre-simulation report**

**Fatal**

<table>
<thead>
<tr>
<th>Fatal</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>If this object exists and is connected to other objects, but not all Snow-SD model inputs are available.</td>
<td>The object 'Snow-SD 1' has no Precipitation 'i' / Temperature 'T' as input.</td>
</tr>
</tbody>
</table>

**Runoff (SWMM) model description**

**Pre-simulation report**

**Fatal**

<table>
<thead>
<tr>
<th>Fatal</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>If the object exists and is not isolated, but has no intensity as input.</td>
<td>The object 'SWMM 1' has no Precipitation 'i' as input.</td>
</tr>
</tbody>
</table>

**GSM model description**

**Pre-simulation report**

**Fatal**

<table>
<thead>
<tr>
<th>Fatal</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>If this object exists and is not isolated, but not all GSM model inputs are available.</td>
<td>The object 'GSM 1' has no Precipitation 'P' / Temperature 'T' as input.</td>
</tr>
</tbody>
</table>
### Appendix

#### SOCONT model description

**Pre-simulation report**

<table>
<thead>
<tr>
<th>Fatal</th>
<th>If this object exists and is not isolated, but not all SOCONT model inputs are available.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The object 'SOCONT 1' has no Precipitation 'P' / Temperature 'T'/ETP 'ETP' as input.</td>
</tr>
</tbody>
</table>

#### HBV model description

**Pre-simulation report**

<table>
<thead>
<tr>
<th>Fatal</th>
<th>If this object exists and is not isolated, but not all HBV model inputs are available.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The object 'HBV 1' has no Precipitation 'P'/Temperature 'T'/ETP 'ETP' as input.</td>
</tr>
</tbody>
</table>

#### GR4J model description

**Pre-simulation report**

<table>
<thead>
<tr>
<th>Fatal</th>
<th>If this object exists and is not isolated, but not all HBV model inputs are available.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The object 'GR4J' has no Precipitation 'P'/ETP 'ETP' as input.</td>
</tr>
</tbody>
</table>

#### SAC model description

**Pre-simulation report**

<table>
<thead>
<tr>
<th>Fatal</th>
<th>If this object exists and is not isolated, but not all SAC model inputs are available.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The object 'SAC' has no Precipitation 'P'/ETP 'ETP' as input.</td>
</tr>
</tbody>
</table>

#### 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

<table>
<thead>
<tr>
<th>Error</th>
<th>Description</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fatal</td>
<td>If this object if used and is not isolated, but the H-Q paired data is not provided.</td>
<td>Times Series 'Time series 1' is missing paired data values.</td>
</tr>
</tbody>
</table>

Source

Pre-simulation report

<table>
<thead>
<tr>
<th>Error</th>
<th>Description</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warning</td>
<td>If a source object exists and no database is connected to the model.</td>
<td>Missing database to get source data.</td>
</tr>
<tr>
<td>Warning</td>
<td>If no group is selected in the Data source for sources.</td>
<td>Missing database group to source data.</td>
</tr>
<tr>
<td>Warning</td>
<td>If this object is used and is not isolated, but the source has no a station from database to provide data series.</td>
<td>No station or sensor for source 'Source 1'.</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fatal</strong></td>
<td>If this object is used and is not isolated, but the H-V paired data is not provided.</td>
<td>Reservoir 'Reservoir 1' is missing H-V paired data.</td>
</tr>
<tr>
<td><strong>Fatal</strong></td>
<td>If this object is used and is not isolated, but does not have an inflow.</td>
<td>The object 'Reservoir 1' has no Flow 'Qe' as input.</td>
</tr>
<tr>
<td><strong>Warning</strong></td>
<td>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.</td>
<td>Hini of reservoir 'Reservoir 1' is out of range.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>If a Reservoir object exists but no database is connected to the model.</td>
<td>Missing database to get initial height of reservoirs.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>If no group is selected in the Data source for reservoirs.</td>
<td>Missing database group to get HPP data.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>If the reservoir uses the initial condition proposed by the user (and not data from database as initial condition).</td>
<td>Hini of reservoir 'Reservoir 1' used as initial height.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>If the reservoir uses the initial condition from the database.</td>
<td>Initial height of reservoir 'Reservoir 1' taken from database.</td>
</tr>
</tbody>
</table>

#### Post-Simulation Report

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning</strong></td>
<td>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).</td>
<td>Volume of reservoir 'Reservoir 1' contains negative values.</td>
</tr>
<tr>
<td><strong>Warning</strong></td>
<td>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).</td>
<td>Volume of reservoir 'Reservoir 1' exceeds its capacity.</td>
</tr>
</tbody>
</table>
## 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\textsubscript{wanted} data as input. | The object 'Turbine 1' has no Flow 'Q\textsubscript{wanted}' 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

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fatal</strong></td>
<td>If this object if used and is not isolated, but the Q-η paired data is not provided.</td>
<td>Hydropower 'Hydropower 1' is missing paired data values.</td>
</tr>
<tr>
<td><strong>Warning</strong></td>
<td>If a source object exists and no database is connected to the model.</td>
<td>Missing database to get Hydropower data. Default Price value(s) will be used.</td>
</tr>
<tr>
<td><strong>Warning</strong></td>
<td>If no group is selected in the Data source for sources.</td>
<td>Missing database group to get Hydropower data. Default Price value(s) will be used.</td>
</tr>
<tr>
<td><strong>Warning</strong></td>
<td>If this object is used and is not isolated, but the consumer has not a station from database to provide data series.</td>
<td>No station or sensor for Hydropower 'Hydropower 1'. Price defined from Default Price.</td>
</tr>
</tbody>
</table>

### Diversion

#### Pre-simulation report

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fatal</strong></td>
<td>If this object if used and is not isolated, but the Q_{Up}-Q_{Diverted} paired data is not provided.</td>
<td>Diversion 'Diversion 1' is missing Q_{Up}-Q_{Diverted} paired data.</td>
</tr>
</tbody>
</table>

### Consumer

#### Pre-simulation report

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning</strong></td>
<td>If a source object exists and no database is connected to the model.</td>
<td>Missing database to get Consumer data. Default QDemand values will be used.</td>
</tr>
<tr>
<td><strong>Warning</strong></td>
<td>If no group is selected in the Data source for sources.</td>
<td>Missing database group to consumer data. Default QDemand values will be used.</td>
</tr>
<tr>
<td><strong>Warning</strong></td>
<td>If this object is used and is not isolated, but the consumer has not a station from database to provide data series.</td>
<td>No station or sensor for Consumer 'Consumer 1'. Consumption defined from Default QDemand.</td>
</tr>
</tbody>
</table>
A.6. Planner object

Planner

Pre-simulation report

<table>
<thead>
<tr>
<th>Fatal Error</th>
<th>A rule is required.</th>
<th>The planner ‘Planner 1’ has no rules defined.</th>
</tr>
</thead>
</table>

Rules

Pre-simulation report

<table>
<thead>
<tr>
<th>Fatal Error</th>
<th>A condition is required.</th>
<th>The rule ‘Rule 1’ in planner ‘Planner 1’ has no condition defined.</th>
</tr>
</thead>
</table>

Conditions

Pre-simulation report

<table>
<thead>
<tr>
<th>Fatal Error</th>
<th>An input is required.</th>
<th>The condition 'Condition 1' in ‘Rule 1’ of ‘Planner 10’ has no input defined.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fatal Error</td>
<td>An object is required.</td>
<td>The condition 'Condition 1' in ‘Rule 1’ of ‘Planner 10’ has no object defined.</td>
</tr>
<tr>
<td>Fatal Error</td>
<td>A result is required.</td>
<td>The condition 'Condition 1' in ‘Rule 1’ of ‘Planner 10’ has no result defined.</td>
</tr>
</tbody>
</table>
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