ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA
SCUOLA DI INGEGNERIA E ARCHITETTURA
DIPARTIMENTO DI INGEGNERIA CIVILE, CHIMICA, AMBIENTALE E DEI MATERIALI
MASTER COURSE IN CIVIL ENGINEERING - INFRASTRUCTURE DESIGN IN RIVER BASINS

TESI DI LAUREA MAGISTRALE
IN
"FLOOD AND DROUGHT RISK MANAGEMENT"

Analysis of the importance of the snow module and of the simulation of extreme streamflows in the presence of a dam using the "GR" hydrological models

---

Author:
Chiara Cesarini

Supervisor:
Prof.ssa Elena Toth
Co-supervisor:
Dr. Guillaume Thirel

ACCADEMIC YEAR 2017-2018

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Life and human dependence on water is well known all around the world. Pressures on water resources are increasing due to different drivers such as expanding populations, urban development, over-exploitation of groundwater, pollution and the uncertain threat of climate changes: as a consequence, improving the management of water resources turs to be essential.
A fundamental step in order to reach this goal is becoming able to understand the hydrologic cycle and estimating the fluxes among the water volumes collected in the earth crust and in the atmosphere (See Fig.1.1). In particular, knowing how rivers do behave is a fundamental information for many reasons, such as managing extremes events, dimensioning reservoirs, but also for preserving the environmental flow in order to sustain natural ecosystems and for designing structures for river training.

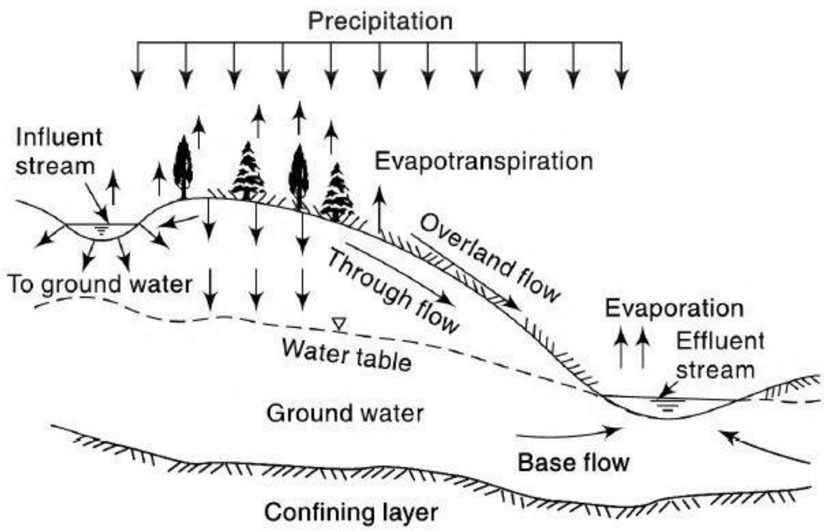

FIGURE 1.1: Water fluxes
(www.ques10.com/p/29581/explain-run-off-in-hydrology-1/)

The water cycle includes processes (precipitation, interception, evapotranspiration, infiltration, percolation underground...) which are complex and interact one to another. Moreover, since they can occur underground, they are not always measurable. This leads to the need of a tool for estimating these processes, namely the hydrologic models. In particular, the rainfall-runoff models are used to assess the streamflows on river catchments.

1.1 Objectives of the analysis

The purpose of this report is to show some analyses performed in order to become familiar with hydrological modelling. The study has been done on three basins, in France and in Austria, with different hydro-climatic regimes in order to understand the impact of the different model components (calibration and validation periods, objective functions, criteria...) on the model efficiency.
The main parts of the analysis will focus on the following topics:

- Performance of some calibration experiments - to familiarize with the hydrological modelling in general
- Snow analysis - to understand the importance of coupling the rainfall-runoff model with a snow accounting model (Cemaneige) in case of a nival hydro-climate of the watershed
- Transformations versus extreme flows - in order to focus more on particular parts of the hydrograph
- Dam module - to study how the presence of a dam could affect the streamflow downstream of a reservoir

The study has been monitored by Dr. Guillaume Thirel (Hydrologic team of IRSTEA - Antony).

1.2 Introduction to the hydrological models

1.2.1 About hydrologic models in general

An hydrologic model can be defined as a *'simplification of a real-world system (e.g., surface water, soil water, wetland, groundwater, estuary) that aids in understanding, predicting, and managing water resources'* [Wik].
Models were developed in parallel with computing power progress. Event-based models originated in the 1930s and they can be used with hand calculation, while in the 1960s the first hydrologic models for continuous simulation

of rainfall-runoff processes emerged: the computing power was sufficient to represent many more processes in a simplified, “conceptual” way [Wheater, 2010].
A rainfall runoff-model is fed with a rainfall time-series (hyetograph) and does generate a temporal time-series of streamflows, namely an hydrograph. Other inputs required can be, for instance, temperatures or potential evapotranspiration. Rainfall-runoff models are mathematical models defined by a set of equations: the latter are function of a number of parameters that can assume different values in order to make the model flexible, namely suitable to reproduce the behaviour of different basins. The parameters are used for taking into account, for instance, different watershed characteristics, soil properties, vegetation cover, watershed topography, soil moisture content and characteristics of groundwater aquifer. They are introduced to optimize the model performance and they need to be estimated or, better, calibrated.
The calibration process is usually performed through an algorithm that optimizes the error criterion chosen as objective function (see chapters 1.2.2 and 1.2.4). The error criteria are measures of how far the simulated values of streamflow are from the observed ones: they are used during both the calibration process (taking the name "objective functions", as just said) and the validation one, being called goodness-of-fit criteria.
The objective functions and transformations that will be used for this study are listed in the sections 1.2.4 and 1.2.5.

1.2.2 Calibration and Validation

Usually the procedures of calibration by means of optimization are based on the comparison between observed and simulated data. Once the parameters are set, simulated discharges are computed and then compared with observed values by means of numerical error criteria: these criteria, when used in calibration, are called *"Objective functions"* (see section 1.2.4) and they aim at estimating the performances of the model and the goodness of the simulated streamflows.

The first calibration procedure adopted was the manual one, by which the parameters were just guessed, the model applied to the dataset and then the observed values compared to the simulated ones. However, from the nineties, with the spread of personal computers, optimization algorithms have made the trial and error calibration procedure automatic, and therefore less time consuming: their goal is to look for the maximization (or minimization, depending on the function) of the "Objective Function".

In order to get a robust model, namely a model reliable on different hydrologic conditions, one would need to calibrate it on a very wide observation period. Nevertheless, having very long data time series is quite infrequent in hydrology, where the samples are usually rather limited. In addition, the calibrated model, before being used in practice, has to be validated: validation allows to analyze the performance of a model, testing it on a discharge sample that differs from the one used for calibration. The same functions used as *"objective function"* during the calibration process, can be used with the name of *"criteria"* in the validation one.
Practically speaking, the observed data are split in two parts: the first one is used for calibration, whereas the second one for validation. Later, the test can be performed inverting the two groups in order to have performed the validation twice and checked the robustness of the model. In order to initialize the values of the state variables inside the model, a warm up period is taken into account just before the the calibration and the validation periods: usually it ranges from one to few years.

1.2.3 Lumped and distributed models

Hydrological models can be distinguished between physically based and lumped conceptually based.
The first ones are based on known scientific principles of energy and water fluxes, while the latter type considers three basic processes within a river basin: the loss of water from storage to atmosphere, storage of water (in soil, vegetation, aquifer, and rivers) and routing of flow over the surface.
In the conceptual models, the catchment is considered as a single unit, and the different parts of the model structure (like the parameters, the functions, the storages...) do not have a direct physical interpretation in the real world [Payan et al., 2008].
Regarding the distributed ones, the watershed can be divided in a rectangular grid mesh or it can be discretized into a limited number of sub-basins depending on the catchment elevations, topography and the drainage characteristics [Zahidul, 2011]. In the first case the model is fully distributed, whereas it is semi-distributed otherwise: on each one of the sub-catchment the model is applied in a lumped way.

1.2.4 Objective functions (and criteria)

As already mentioned, the maximization (or minimization) of the objective functions are what the optimization algorithm looks for during the calibration process in order to find the best parameters for the model. These functions are also calculated as criteria during the validation process in order to check the goodness of the model.
The four objective functions provided with the *airGR* package (see 2.1) and used as error criteria in the next chapters are the *Root Mean Square Error (RMSE)*, the *Nash-Sutcliffe Efficiency (NSE)*, the *Kling Gupta Efficiency (KGE)* and the *Modified Kling Gupta Efficiency (KGE′)* described below.

- Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{\sum_i (Q_{SIM}^{(i)} - Q_{OBS}^{(i)})^2}{N}} \tag{1.1}$$

- Nash-Sutcliffe Efficiency (NSE)

$$NSE = 1 - \frac{\sum_i (Q_{SIM}^{(i)} - Q_{OBS}^{(i)})^2}{\sum_i (Q_{OBS}^{(i)} - \mu(Q_{OBS}))^2} \tag{1.2}$$

  Where:

  - $Q_{OBS}^{(i)}$ is the observed flow at the i time-step
  - $Q_{SIM}^{(i)}$ is the simulated flow at the i time-step
  - N is the length of the sample (time-series)
  - $\mu(Q_{OBS})$ is the mean of the observed discharges

The value of NSE can vary in $]-\infty, 1]$, with the perfect model performances if $NSE = 1$, while if $NSE = 0$ the performances are equal to computing the mean of the observed discharges, and finally, if $NSE < 0$ then the model predictive power is worse than computing the simple mean.
Looking at the equations above it's clear that the $RMSE$ and NSE are closely related: in fact, they are two modified versions of the same *absolute* criteria *Mean Square Error* (see Eq. 1.3): the $RMSE$ is its squared root, while the NSE is its normalized version, therefore a *relative* criteria. A reason why $RMSE$ and MSE are not widely used criteria, is that they depend on the units of

measure of the predicted variable, so they do not allow to compare performances of models run to simulate different phenomena [Gupta et al., 2009].

$$MSE = \frac{\sum_i (Q_{SIM}^{(i)} - Q_{OBS}^{(i)})^2}{N} \tag{1.3}$$

The NSE criteria has been one of the most often used performance criteria in hydrology [Guse et al., 2017]: however, its suitability has been discussed by many authors (such as [Schaefli and Gupta, 2007]), mainly because it uses the mean of the observed streamflows as reference, which is not an appropriate benchmark model in case of basins with different hydrologic regimes: using the mean as baseline leads, for instance, to the overestimation of the model performances for strongly seasonal time series [Gupta et al., 2009]. From that, the need of look for better error criteria emerged.
With the goal of improving the estimation of the model performance, the NSE criterion can be split into separate components that can be directly related to the model error: this allows each of them to be analysed separately in order to understand how they contribute to the model performance. This approach results in the definition of a new criterion, namely the KGE one [Gupta et al., 2009].

- Kling Gupta Efficiency (KGE) [Gupta et al., 2009]

$$KGE = 1 - \sqrt{(r-1)^2 + (\alpha - 1)^2 + (\beta - 1)^2} \tag{1.4}$$

- Modified Kling Gupta Efficiency (KGE') [Kling et al., 2012]

$$KGE' = 1 - \sqrt{(r-1)^2 + (\gamma - 1)^2 + (\beta - 1)^2} \tag{1.5}$$

Where:

- $r = \frac{Cov(x,y)}{\sqrt{Var(x)Var(y)}}$ Pearson coefficient [-]
- $\alpha = \sigma(Q_{SIM})/\sigma(Q_{OBS})$ variability ratio used for KGE [-]
- $\beta = \mu(Q_{SIM})/\mu(Q_{OBS})$ bias ratio [-]
- $\gamma = CV(Q_{SIM})/CV(Q_{OBS})$ variability ratio used for KGE' [-]

r is the Pearson linear correlation coefficient of Q_{SIM} with respect to Q_{OBS}, σ is the standard deviation of the discharges distribution, μ is the average of the discharges, "CV" is the coefficient of variation of the observed discharges, namely σ/μ. The best values to aim at are RMSE = 0, NSE = 1, KGE = 1 and

KGE′ = 1.
The optimal value of r, α, β and γ is equal to 1.
Both α and γ are measures of the relative variability in the simulated and observed values.

With the *KGE* is possible to study the ability of the model to reproduce different aspects of the observed streamflow distribution (i.e. the mean, the standard deviation and the correlation): the optimal value of this criterion is reached when all these components assume their target values (namely 1); therefore the optimization of *KGE* consists in looking for the best combination of r, α and β [Guse et al., 2017]. Additionally, it is possible to put a different weight on the different components of the criterion using the formula 1.6 that includes the s_r, s_α and s_β scaling factors.

$$KGE = 1 - \sqrt{[s_r(r-1)^2] + [s_\alpha(\alpha-1)^2] + [s_\beta(\beta-1)^2]} \tag{1.6}$$

Moreover, in order to have the variability ratio α uncorrelated with the bias ratio β, the numerator and denominator of α can be divided by the numerator and denominator of β, leading to obtain γ as new variability ratio, and to get KGE' as a modified *Kling Gupta Efficiency* error criterion [Kling et al., 2012]. This turns α to be independent from the mean of the distribution and therefore helps the variation to be unaffected by the error that, for instance, could occur when the precipitation inputs are biased. As already said, the coefficients incorporated in the KGE and KGE′ formula, give different information on different aspects of the simulated hydrograph: the Pearson correlation coefficient r is a measure of the temporal dynamics (timing), while α (or γ) and β are indexes of how well the distribution of discharges is preserved [Kling et al., 2012]. In particular β tells if, on average, the discharges have been overestimated or underestimated with respect to the observed ones, whereas α and γ give information about the variation of the simulated discharges relatively to the one of the observed.

1.2.5 Transformations

In order to focus on particular ranges of simulated streamflows, some transformations can be applied to the discharge value. The three transformations of the output variable, which generally is the streamflow at the closure section, frequently used are:

- Square root function: $\sqrt{Q}$

- Inverse function: $1/Q$
- Logarithmic function $\log(Q)$

According to the target of the analysis, other hydrological variables can be transformed.
A different transformation can be used depending on the range of flows that have to be analyzed: in particular, the not-transformed streamflows are used to put more weight on high flows, the *square root function* does not favour high or low flows, whereas the inverse function puts more weight on low flows [Santos et al., 2018].
The logarithmic function $\log(Q)$ can be also used to analyse low discharges, but it won't be take into account in the following analysis.

Chapter 2

GR Models in airGR R-package and case study

2.1 Intro

In order to use the GR hydrological models, I worked in the airGR package framework, which proposes functions to run the different models, calibrations and transformations [Coron et al., 2017][Coron et al., 2018]. The version of the R package used is 1.0.15.2.

The first GR models were developed by Claude Michel at the beginning of the 1980s, starting from a simple module structure and increasing the complexity every time a better performance model was obtained [Coron et al., 2017].
The several existing models differ for their complexity, therefore for the time-step and the number of parameters, and they are used in different countries for various purposes, such as water resources assessment, flood and drought estimation and forecast, land use and climate change impact assessment [Coron et al., 2017].
GR stands for *Génie Rural*, literally *Rural Engineering*, and the letters highlighting the time-step of the models in its names are:

- *H* stands for *horaire* (hourly) time-step
- *J* stands for *journalier* (daily) time-step
- *M* stands for *mensuel* (monthly) time-step
- *A* stands for *annuel* (annual) time-step

The number present between "GR" and the time-step delineates the number of free parameters of the model, i.e. the number of parameters to calibrate.

2.2 GR4J

The GR4J model is a four-parameter daily lumped rainfall-runoff model, which belongs to the SMA (soil moisture accounting) models. The two inputs of the model are the precipitation depth P [mm] and the potential evapotranspiration PE [mm]; as first step the net rainfall P_n and the net evapotranspiration E_n are calculated [Perrin et al., 2003]. The model is built up with a soil moisture (or production) store with an evolution described by the power function of its storage S; this first reservoir is followed by a transfer function that, in the daily version of the model, splits the water in two fluxes equivalent to 90% and 10% of the total: the first one is led to the unit hydrograph UH1 and then to a non-linear routing store, whereas the second skips the store, being routed directly by UH2. UH1 and UH2 are used to simulate the time lag between the precipitation and the streamflow peak ([Perrin et al., 2003]). Moreover, a groundwater exchange function is applied to the two fluxes ($Q9$ and $Q1$) and is described by a power law of the routing store level R: finally, the two resulting streamflows Qr and Qd are summed up in order to obtain the final Q (see fig. 2.1).

The four parameters characterizing the GR4J model are:

- x_1 [mm]: maximum capacity of SMA (Soil Moisture Accounting) store
- x_2 [mm/day] : groundwater exchange coefficient
- x_3 [mm]: capacity of routing store
- x_4 [day]: time base of Unit Hydrographs

	Median	**80% conf. int.**
X1 (mm)	350	100 - 1200
X2 (mm/day)	0	-5 to 3
X3 (mm)	90	20 - 300
X4 (day)	1.7	1.1 - 2.9

TABLE 2.1: Statistics of GR4J parameters[Perrin et al., 2003]

In table 2.1 are reported the mean values and the 80% confidence intervals got from the 0.1 and 0.9 percentiles of the distribution of models run on a sample of 429 catchments in Australia, Brazil, France, the Ivory Coast and the United States [Perrin et al., 2001]. It has to be noted that parameters ranges

(between the maximum and minimum values) are not used in this case.
All the formula that build up the GR4J model are listed in fig 2.2 while its structure is reported in fig. 2.1.

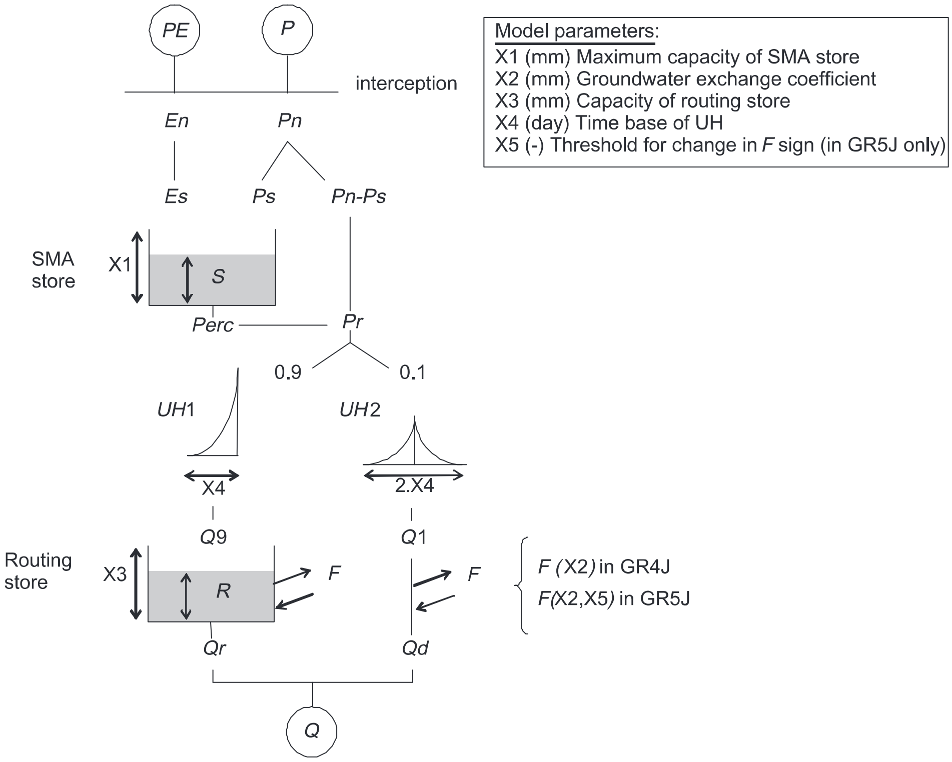

FIGURE 2.1: Structure of the GR4J and GR5J rainfall-runoff models [Pushpalatha et al., 2011]

$$\text{If } P \geq E, \quad \text{then } P_n = P - E \text{ and } E_n =$$

$$\text{otherwise } P_n = 0 \quad \text{and} \quad E_n = E -$$

$$P_s = \frac{x_1\left(1 - \left(\frac{S}{x_1}\right)^2\right)\tanh\left(\frac{P_n}{x_1}\right)}{1 + \frac{S}{x_1}\tanh\left(\frac{P_n}{x_1}\right)}$$

$$E_s = \frac{S\left(2 - \frac{S}{x_1}\right)\tanh\left(\frac{E_n}{x_1}\right)}{1 + \left(1 - \frac{S}{x_1}\right)\tanh\left(\frac{E_n}{x_1}\right)}$$

$$S = S - E_s + P_s$$

$$\text{Perc} = S\left\{1 - \left[1 + \left(\frac{4}{9}\frac{S}{x_1}\right)^4\right]^{-1/4}\right\}$$

$$S = S - \text{Perc}$$

$$P_r = \text{Perc} + (P_n - P_s)$$

$$\text{For } t \leq 0, \ \text{SH1}(t) = 0$$

$$\text{For } 0 < t < x_4, \ \text{SH1}(t) = \left(\frac{t}{x_4}\right)^{5/2}$$

$$\text{For } t \geq x_4, \ \text{SH1}(t) = 1$$

FIGURE 2.2: Formulas of the GR4J rainfall-runoff model [Perrin et al., 2003]

2.3 GR5J and GR6J

The need of getting improved low-flow simulations led to the necessity of upgrading the GR4J model. Typically, the water exchange between surface and groundwater is one of the main factors that affects the low-flow regime of the catchment [Pushpalatha et al., 2011], therefore a different modelling of the groundwater exchange function F has been taken into account. The GR5J and GR6J models resulted to be a valid solution for the low flows simulating issue; however, they are not supposed to alter the simulation of high flows [Pushpalatha et al., 2011]. The five parameters version of the GRJ model involves a new formulation of the groundwater exchange that includes x_5 as $F = x_2(\frac{R}{x_3} - x_5)$ (see fig.2.1). In the GR6J model the modelling of the groundwater exchange is further improved including an additional exponential store placed in parallel with the already existing routing one (see fig.2.3):

SC is the *splitting coefficient* of effective rainfall between the two stores [Pushpalatha et al., 2011]; it is a fixed parameter equal to 0.4. SC is equivalent to the variable "c" in the function *frun_GR6J.f* of the *airGR* package source.

The additional parameters of GR5J and GR6J with respect to GR4J are (see fig. 2.1 and fig. 2.3):

- x_5 [-]: threshold for change in F sign (in GR5J and GR6J only)
- x_6 [mm]: capacity of the new routing store (in GR6J only)

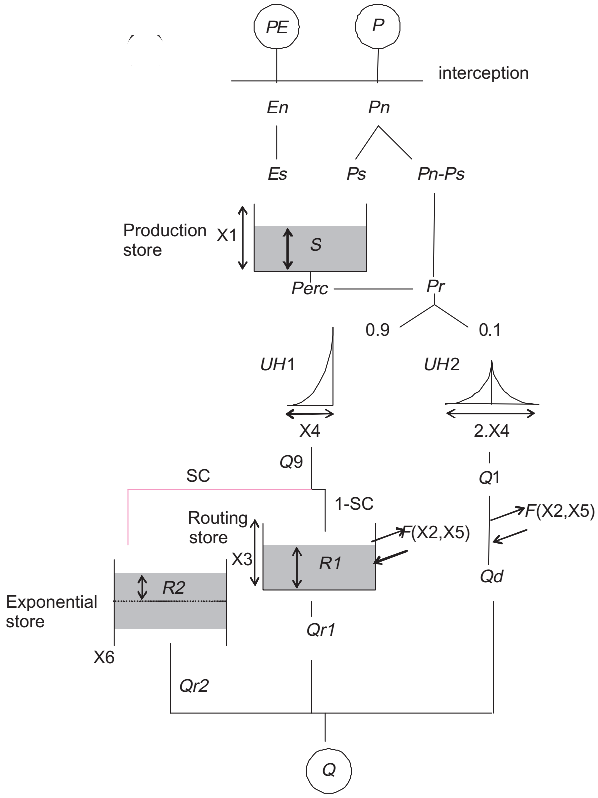

FIGURE 2.3: Structure of the GR6J rainfall-runoff model [Pushpalatha et al., 2011]

2.4 CemaNeige

CemaNeige is a semi-distributed model developed in order to improve the simulation of the streamflows in river catchments affected by the presence of snow. It is a two-parameter daily time-step snow accounting model and it gives as final output the simulated snowmelt, which is summed to rainfall. The main five functions of CemaNeige are [Valéry et al., 2014]:

- Determination of the solid fraction of precipitation
- Snow accumulation
- Updating of the snowpack cold-content
- Potential snow-melt computation
- Actual snow-melt computation

In particular the watershed is divided into different elevations sub-catchments with equal areas, namely *elevation zones*, which allow the model to simulate the evolution of the snow cover consistently with the different altitudes. The CemaNeige model needs precipitation and temperatures as input data: therefore, as second step, these data are estimated for each elevation zone. Successively, a differentiation between solid and liquid precipitation is performed depending on the average altitude of each sub-catchment. Then the snow is collected in a theoretical reservoir that represents the snow pack and is supplied only by the solid part of the precipitation: the snow pack, melting, will produce water that will be added to the liquid precipitation, and together they will make up the final streamflow. This melting process is simulated through the transfer function (see fig. 2.4). The melted snow output of CemaNeige then becomes one of the inputs of the rainfall-runoff model.

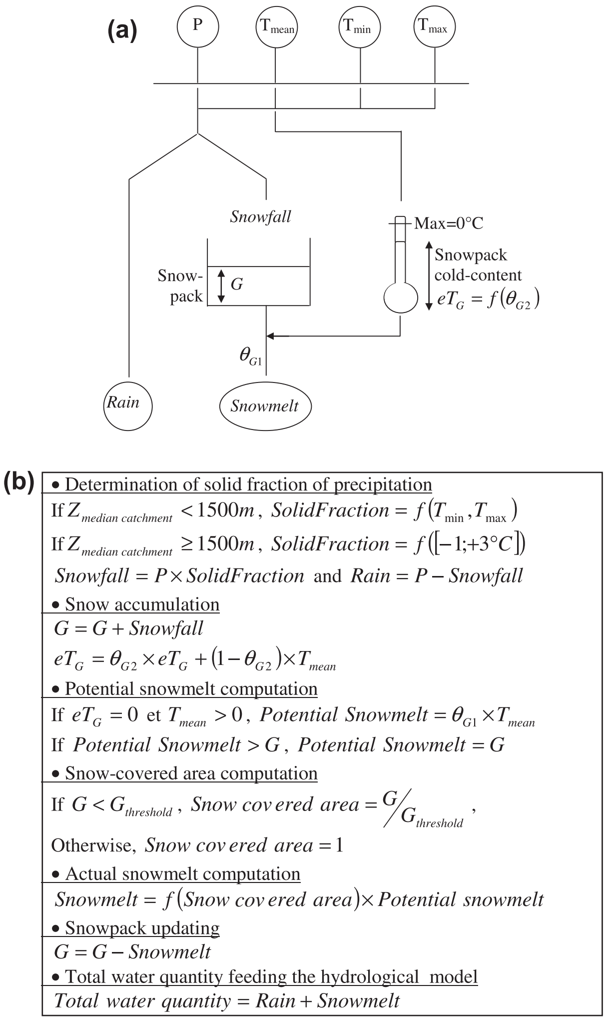

FIGURE 2.4: Structure of the CemaNeige snow accounting routine model [Valéry et al., 2014]

The two parameters to be calibrated in this case are:

- c_1 [-]: weighting coefficient for snow pack thermal state
- c_2 [mm/°C/day]: degree-day melt coefficient

Regarding the input data needed to run CemaNeige, also the *Zinputs* and the *HypsoData* vectors are needed.

- *Zinputs*: mean of the elevations at which input precipitation and mean air temperature are calculated for the considered basin
- *HypsoData*: it describes the hypsometric curve though a vector of 101 components, including the minimum, the maximum and the 99 quantiles (from q_{01} to q_{99}) for the distribution of the elevations in the considered basin.

This allows extrapolating the precipitation and temperature values at the altitudes of the chosen altitude layers, which are typically 5.

2.5 Calibration algorithm

The final goal of an optimization algorithm is the maximization or minimization of an objective function performed choosing its "best available" input values within an allowed set. The solution that minimizes (or maximize, depending on the target) the objective function is the *optimal solution*.
The calibration algorithm proposed in airGR to optimise the error criterion selected as objective function, is the procedure proposed by Claude Michel (through the "$Calibration\_Michel()$" function [Perrin, 2000]): this algorithm is the one used for all the analyses described in this report.
On the other hand, different optimization methods can be used combined with airGR, such as the "$nlminb()$" function, the Different Evolution strategy ("$DEoptim()$"), the Particle Swarm strategy ("$hydroPSO()$") and the MA-LS-Chains one ("$malschains()$"). How to use all these alternatives is explained in the two vignettes "Plugging in new calibration algorithms in airGR" and "Parameter estimation within a Bayesian MCMC framework" of the airGR package (see `https://cran.r-project.org/web/packages/airGR/index.html`).

In short, the Claude Michel algorithm starts with a screening to define the initial set of parameters: a screening is performed using either a rough predefined grid (considering various initial values for each parameter) or a list of initial parameter sets. Then a steepest descent local search algorithm is performed, starting from the result of the screening procedure (see function *Calibration_Micheal* in the airGR package [Coron et al., 2017]).
The first step of the iterative search consists in calculating the objective function for the initial vector of n parameters $x^0 = (x_1^0, x_2^0, ..., x_i^0, ..., x_n^0)$. Then, each i-parameter is subsequently modified adding and subtracting a Δx in order to get two new sets of parameters $(x_1^0, x_2^0, ..., x_i^0 - \Delta x, ..., x_n^0)$ and $(x_1^0, x_2^0, ..., x_i^0 + \Delta x, ..., x_n^0)$: for all the $2 * n$ new sets of parameters, the objective function is computed and the vector which gives the best improvement of the objective function is chosen as x^1, namely the initial vector for the following step. The objective function is computed again for the latter vector, hence the following iteration starts and so on. x^1 is identical to x^0 except for the component $x_i^0 - \Delta x$ or $x_i^0 + \Delta x$. The procedure stops once the Δx becomes smaller than a predefined threshold Δx_{min}: hence, the best vector of parameters x^* is found

[Perrin, 2000]. The objective function, the initial Δx_0 and the minimum Δx_{min} have to be chosen by the user: for instance, Perrin [2000] fixed $\Delta x_0 = 0.64$ and $\Delta x_{min} = 0.01$

2.6 Study regions

The analysis that follows will concern three different catchments: two in France and one in Austria. The first French basin is located in Brittany, close to the Atlantic Ocean, while the second one is in the south of France. The latter, together with the Austrian river, is located on the Alps (Fig. 2.5). Also a fourth basin will be described in this paragraph, but it will be only used to perform some quick analyses that will appear only in the last chapter of this report (chapter 6).

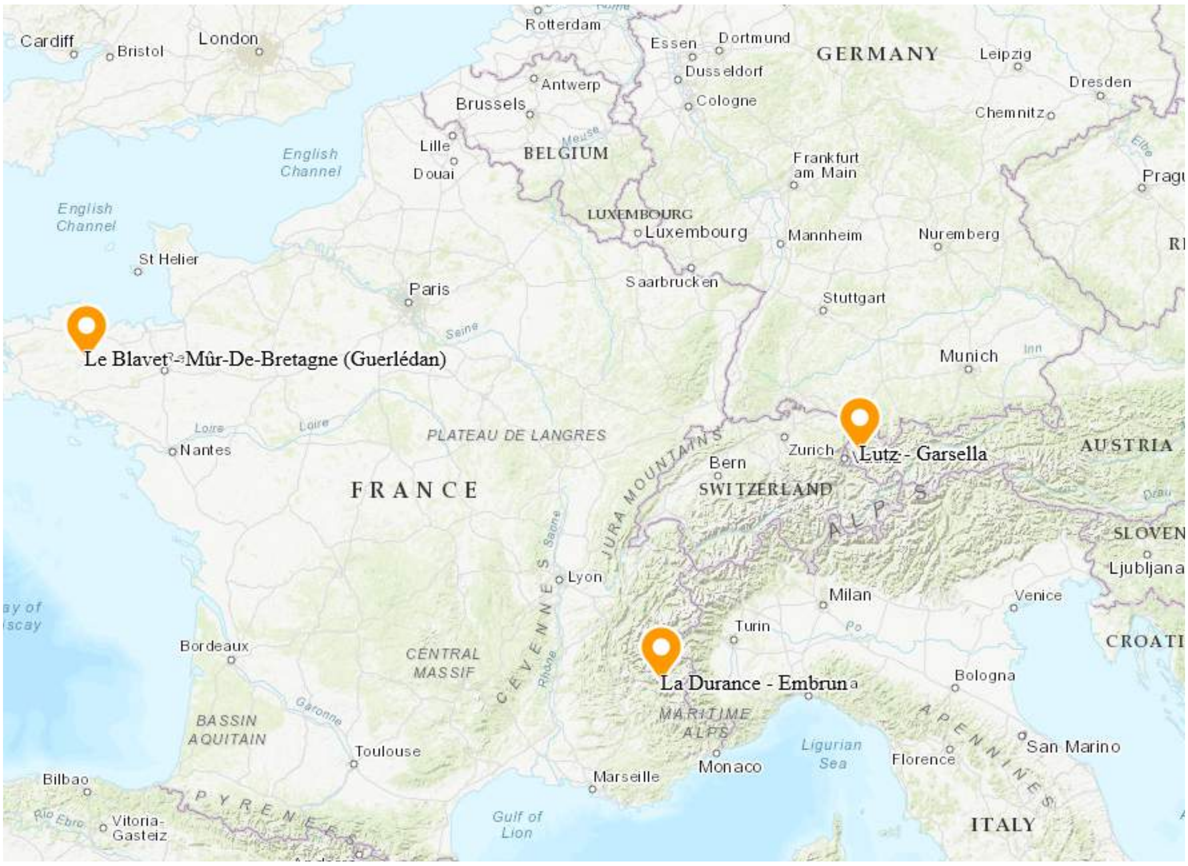

FIGURE 2.5: Location of the three basins (gauges)
www.geoportail.gouv.fr

2.6.1 Durance river at Embrun - X0310010

"La Durance" (see Fig. 2.6) is an alpine river, inflow of the Serre-Ponçon hydropower reservoir.
It has its sources on the Chenaillet mount, close to the Italian border and 2.634 m above the sea level. It ends in the Rhône, few kilometers from Avignon.

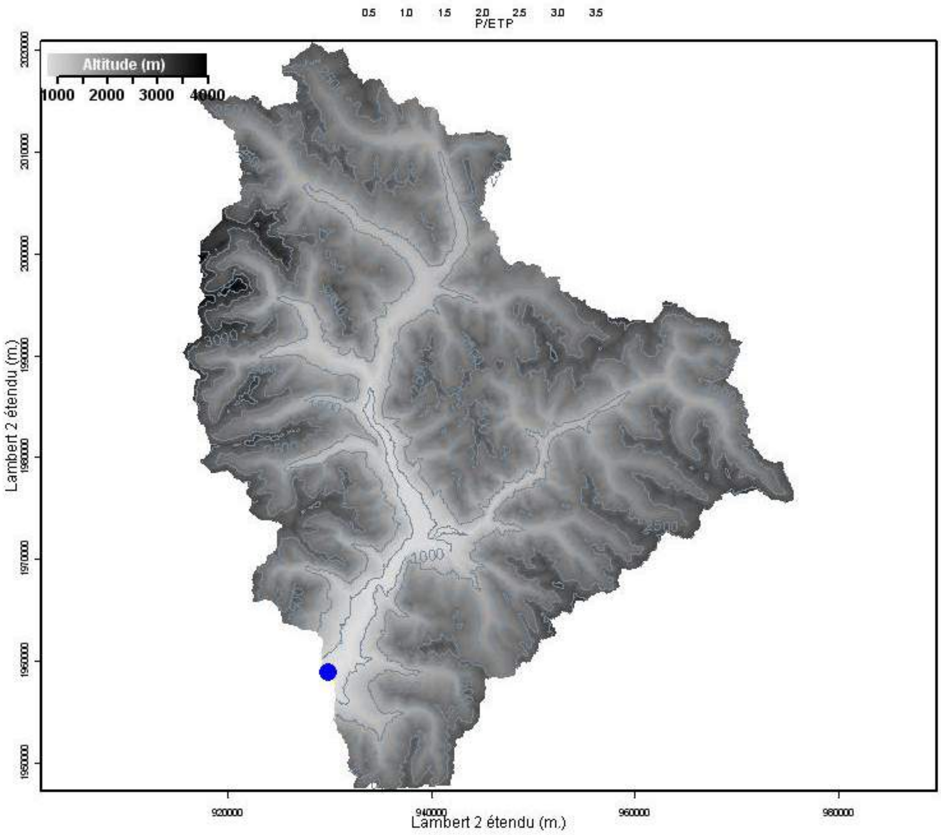

FIGURE 2.6: Map of the Durance river catchment `webgr.irstea.fr/wp-content/uploads/2016/10/X0310010.jpg`

	Area $[km^2]$	**Mean elevation** $[m]$
Basin X0310010	2282.76	2104.35

TABLE 2.2: Area and mean elevation of the Alpine French basin (X0310010)

2.6.2 Blavet river at Mûr-de-Bretagne (Guerlédan) - J5412110

"Le Blavet" flows in the French Brittany region mainly from north to south, up to the Atlantic Ocean. Getting closer to Mûr-de-Bretagne, the river turns into the man made Guerlédan lake: the reservoir was built during the 1930, in order to supply electricity to the central part of Brittany.

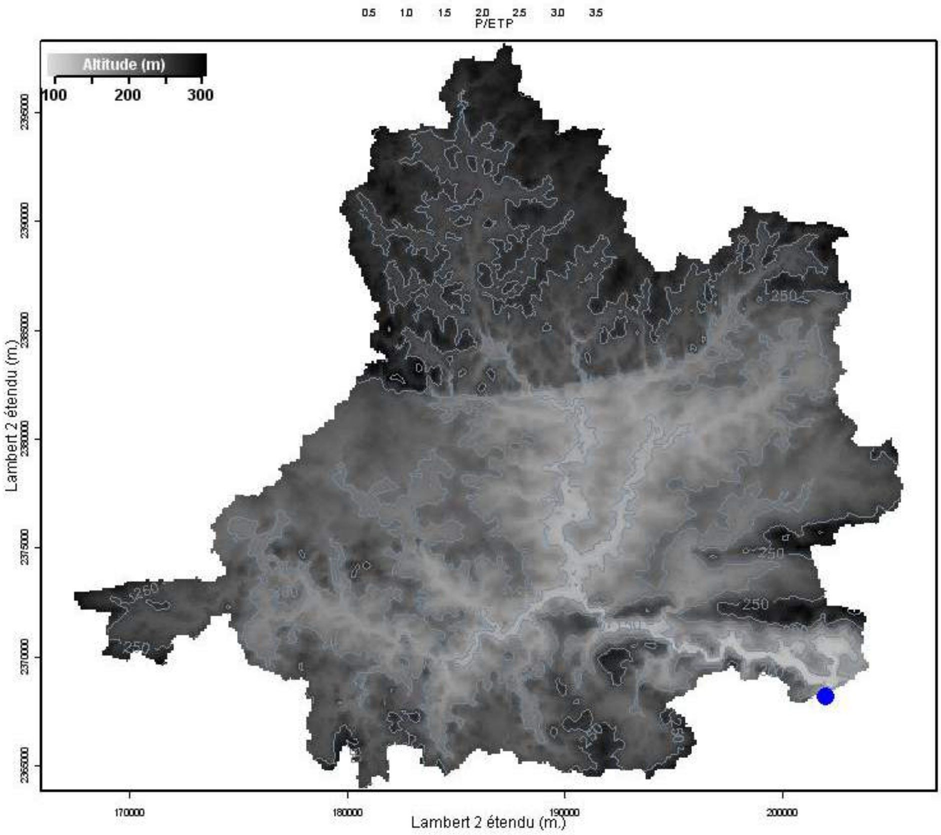

FIGURE 2.7: Map of the Blavet river (J5412110) catchment `webgr.irstea.fr/wp-content/uploads/2016/10/J5412110.jpg`

	Area $[km^2]$	**Mean elevation** $[m]$
Basin J5412110	675.64	214.53

TABLE 2.3: Area and mean elevation of the Breton French basin (J5412110)

2.6.3 Lutz river at Garsella - HZBnr. 200105

The Lutz river (gauge Garsella) flows through the Great Walser Valley (Großes Walsertal): it's an alpine saw-cut valley located in the northern part of the Limestone Alps, economically important also because source of drinking water.

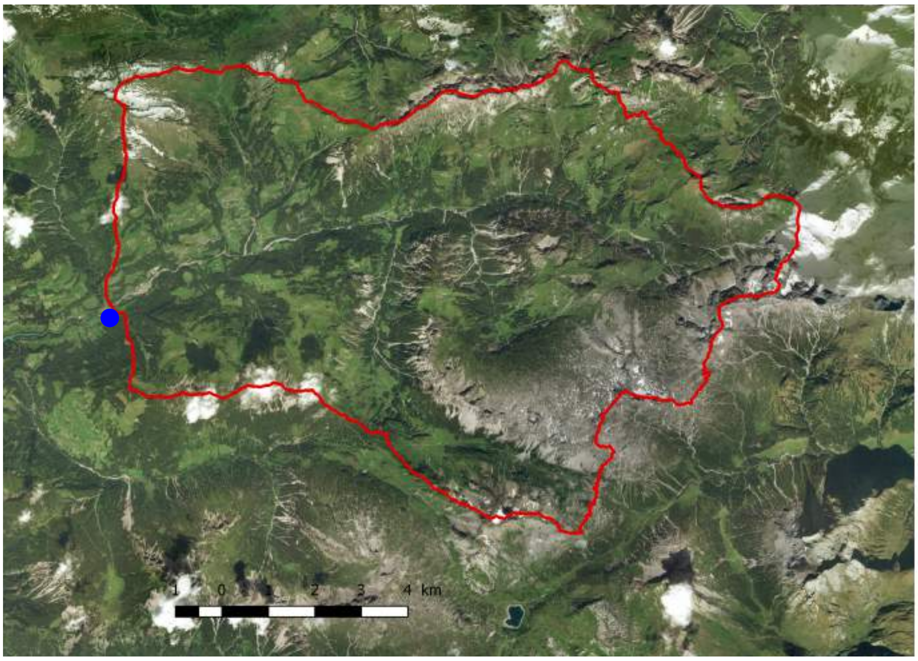

FIGURE 2.8: Map of the Lutz river (200105) catchment - Garsella gauge in blue

	Area $[km^2]$	Mean elevation $[m]$
Basin 200105	95.50	1600.24

TABLE 2.4: Area and mean elevation of the Alpine Austrian basin (200105)

2.6.4 Ill river at Oberhergheim - A1320310

The Ill river starts from the Jura mountains and flows in the Alsace region, in the north-eastern France.
As already introduced, this catchment will be used just for a quick final analysis described at the end of this report. In particular it will be used to study

the impact of a dam on a watershed: in this basin, in fact, a barrage diverts the streamflow of the Ill river to create a reservoir.

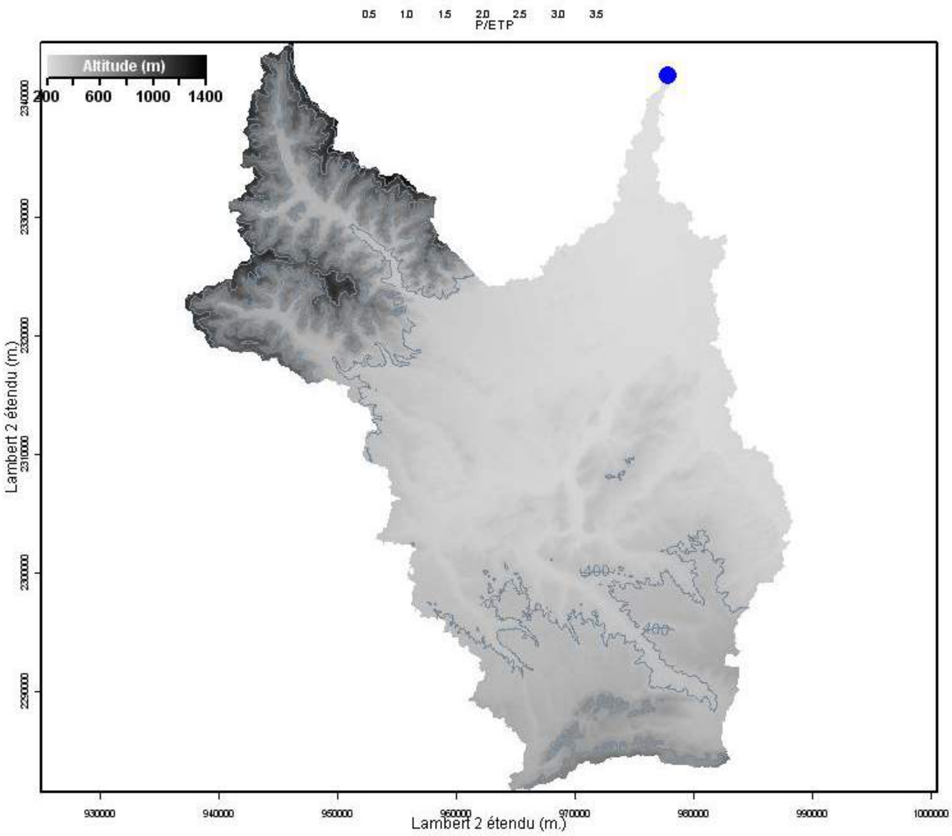

FIGURE 2.9: Map of the Lill river (A1320310) catchment `webgr.irstea.fr/wp-content/uploads/2016/10/J5412110.jpg`

2.6.5 Mean input data and areas of the three main basins

In order to highlight the different climates of the basins over the two different calibration used, the table in the following Tab.2.10 has been computed from the available time-series.

Basin	Ptot_Cal1 [mm/day]	Ptot_Cal2 [mm/day]	Temp_Cal1 [°C]	Temp_Cal2 [°C]	Qobs_Cal1 [mm/day]	Qobs_Cal2 [mm/day]
Mean_X0310010	2.85	2.82	2.47	2.95	2.05	1.86
Mean_J5412110	2.70	3.06	9.90	10.53	1.39	1.40
Mean_200105	5.78	5.53	3.54	4.05	5.97	5.55

FIGURE 2.10: Mean rainfall and temperature over the calibration periods ($Cal1$ and $Cal2$ are the 1^{st} and the 2^{nd} half of the sample time-series; see chapter 3.1 for the detailed description of the calibration periods)

2.7 Available data

The French data used for the analysis have been provided by Dr. Guillaume Thirel (IRSTEA), while the Austrian ones by TU Wien, inside a collaboration project with the University of Bologna.
They consist in time-series of daily precipitation, daily potential evapotranspiration, daily streamflows and daily temperature values.
Regarding France, the observed discharges source is the *Banque Hydro* (`www.hydro.eaufrance.fr`) while the meteo data come from *Météo-France* [Vidal et al., 2009].

The French basins meteo data are available from the 01/08/1958 to the 31/07/2010: the Q_{obs} values for "La Durance" are available from the 01/01/1960 to the 31/07/2010, while streamflows data of the Blavet river go from 01/08/1958 up to 31/12/2008.
Regarding the Austrian basin data, instead, they all go from the 01/01/1976 to the 31/12/2008.
In order to use the CemaNeige model, also the *Zinputs* and the *HypsoData* were required as input data (see paragraph 2.4).
The Austrian meteorological data were given per each altitudinal range, namely splitting the catchment area in zones, each of them spanning a 200 m elevation range. Due to the fact that the models have to be applied in a lumped way, all the Austrian data have been spatially aggregated by means of the weighted average with respect to the area of each elevation zone.

2.8 Analysis of the hydro-climatic regimes

In order to understand the hydro-climatic regimes of the three rivers object of this analysis, the monthly average of observed streamflows and observed rainfalls have been plotted. Additionally, for the two alpine basins, the monthly snowfall has been simulated with CemaNeige over the $Cal1$ (see paragraph 4.2 for details about the objective functions used); then it has been added to the graphs (see Fig.2.11 and 2.12). For the Brittany basin the estimation of solid precipitation has been avoided due to the already known non-nival hydroclimatic regime of the basin (see Fig. 2.13).

The values taken into account for the following study are just monthly averages, therefore the fluctuation of discharges would be greater over shorter periods and would change depending on the year considered.

2.8.1 Durance river at Embrun - X0310010

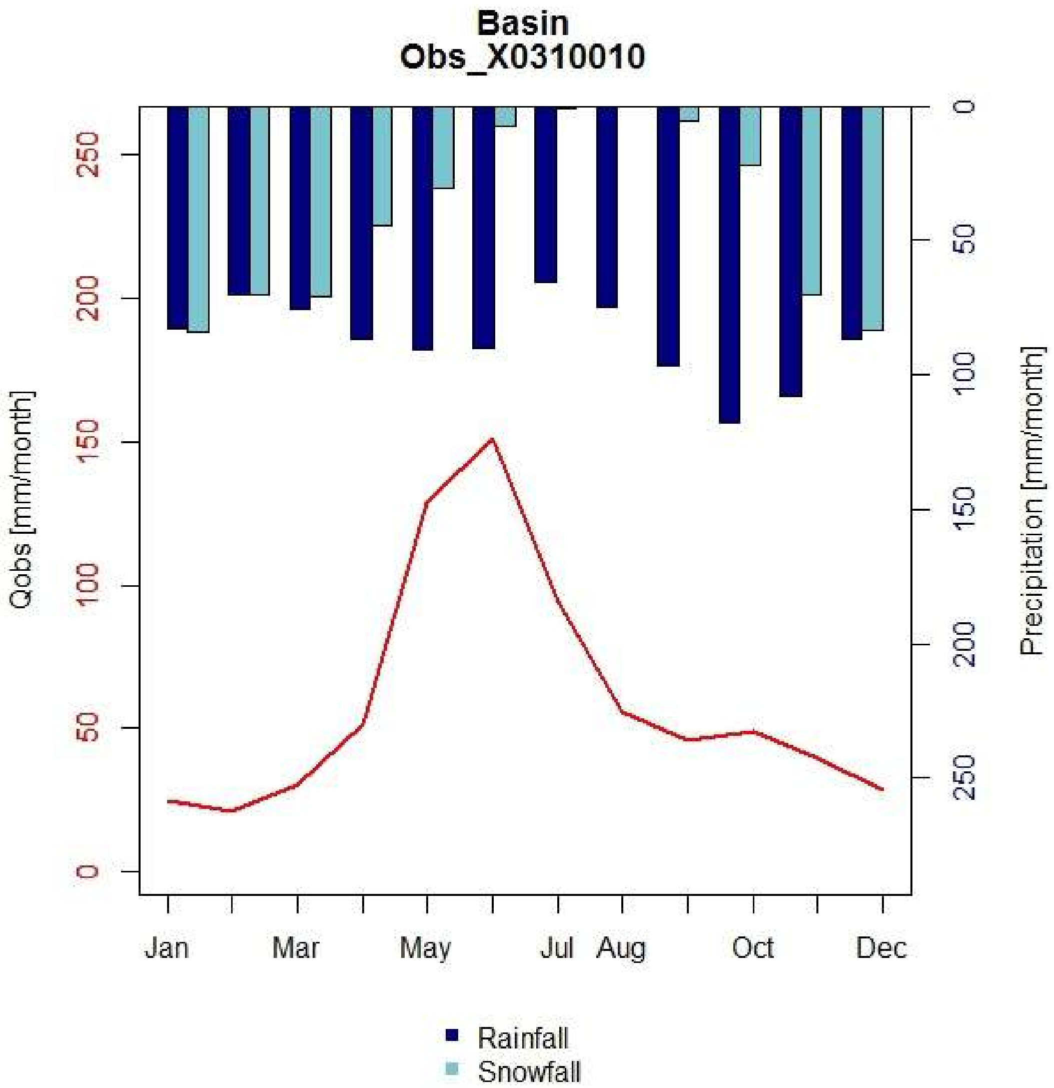

FIGURE 2.11: Hydro-climate basin X0310010

The presence of the Alps in the central European continent profoundly affects the climate of all the surrounding regions and areas, including the Durance basin: the latter is a nival regime basin [Wik]. From the trend that emerges in Fig.2.11 we can infer that water fallen as precipitation during autumn, winter and spring is stored in the form of snow and it's gradually released during the following summer: this can be understood focusing on the time lag present between the peaks of the precipitation and of the discharge: the biggest discharges occur in the early summer, with a peak in June, while the higher precipitation (rainfall and snowfall together) are the winter ones. It can also be observed that the streamflow is never zero: this can also be due to the accumulated snow (and glaciers) that melts and feeds the water course even in the hottest season, allowing to maintain a regular flow.

2.8.2 Lutz river at Garsella - 200105

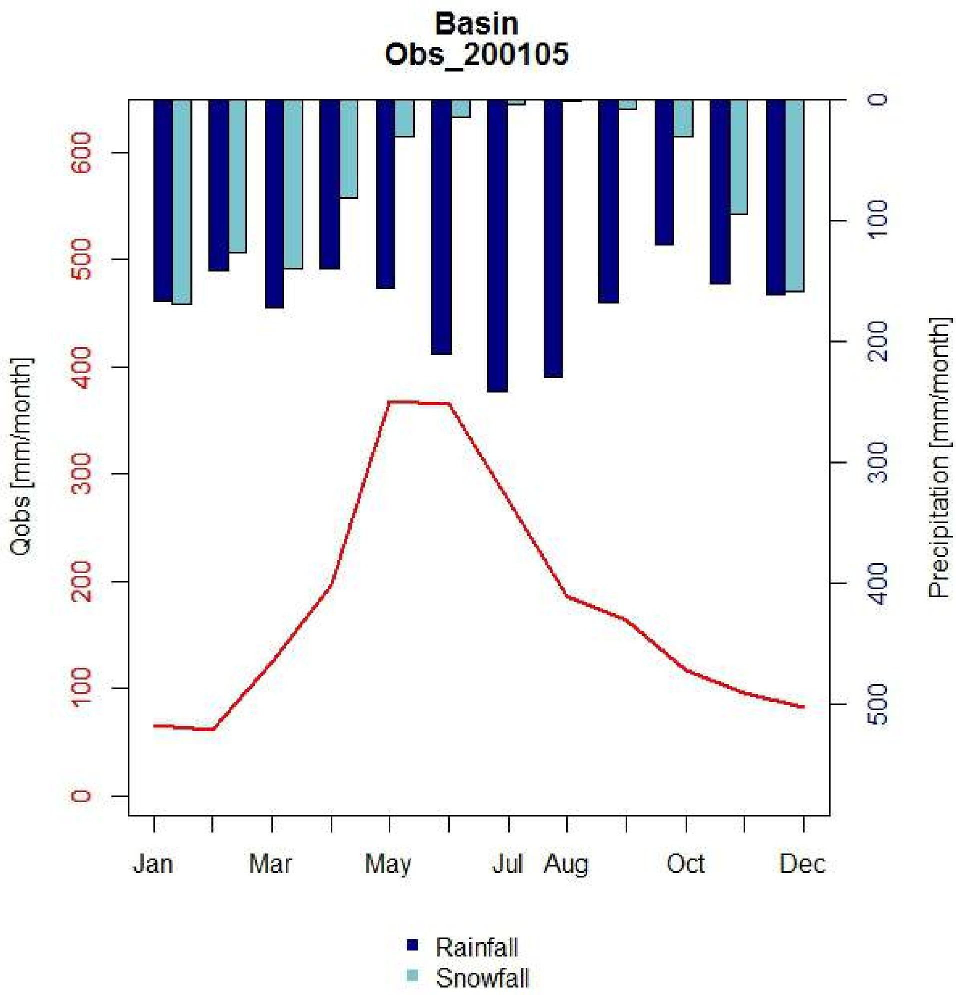

FIGURE 2.12: Hydro-climate basin 200105

Similar considerations to those done for the Alpine Durance nival basin can be done also in this case. With respect to "La Durance", though, it can be noted that both the precipitation and streamflow amounts are higher for the Austrian basin. Also in this case the peak of the discharge doesn't coincide perfectly with the one of the rainfall: this offset between them can be due to the fact that the snow storage delays the water supply to the river, shifting it towards the warmer months.

2.8.3 Blavet river at Mûr-de-Bretagne (Guerlédan) - J5412110

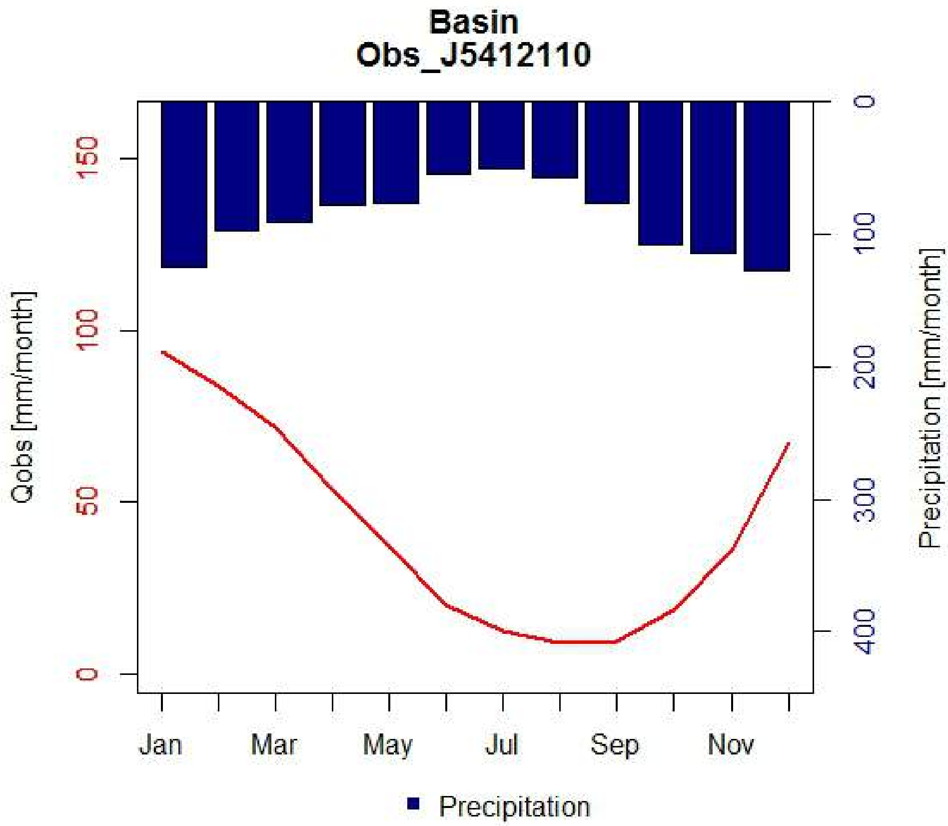

FIGURE 2.13: Hydro-climate basin J5412110

Looking at Fig.2.13 it can be noticed that streamflows oscillate between 20 mm and 100 mm per month: the highest discharges appear to occur between December and March, with a maximum in January. The driest period instead is the summer one, from the end of June to the last days of September, with a minimum in August-September; from October the streamflow starts increasing again.

With respect to the other two basins, it can be noticed that the difference between precipitation and discharge in this case is higher, probably due to the substantial amount of evapotranspiration. All these considerations are consistent with the Brittany region climate, which in fact is oceanic: given that precipitation is more homogeneously dispersed throughout the year, typically the dry season is missing, with at least 40 mm of precipitation per month. Usually rain is the most common form of precipitation for the majority of the year [Wik].

Chapter 3

Calibration experiments

3.1 Calibration experiments and analysis of performances in validation

As already mentioned in chapter 1, the main purpose of the implementation analysis is to test how the different aspects of the hydrological model can play a role in its efficiency: the structure of the model itself, the calibration and validation periods, the objective functions (including goodness-of-fit criteria and the transformations) can influence the simulations in different ways, changing drastically its quality. In particular the analysis can focus on high or on low flows only, therefore how the different transformations can be used for a more efficient study results to be crucial. Other factors to be studied are, for instance, the hydroclimate regime of the catchment and, possibly, the presence of snow on the basin: knowing that can be fundamental to run a good simulation. All these aspects will be discussed in details in the following chapters.

As first step of the analysis that follows, the GR4J, GR5J, GR6J and GR4J+CemaNeige models have been run over the three basins, combining different calibration periods, validation periods, objective functions and criteria, for a total of 192 simulations (3 basins, 4 models, 2 calibrations periods, 2 validation periods, 4 objective functions for calibration). The result of all the simulations is a dataframe including the objective function values, criteria values and parameters got for all the different combinations (see a part of it in fig. 3.1).
The several simulations were performed combining:

- Calibration periods: 1961-1985 ($Cal1$) and 1986-2010 ($Cal2$) for the two French basins, and 1978-1993 ($Cal1$) and 1994-2008 ($Cal2$) for the Austrian one

- Validation periods: 1961-1985 ($Val1$)and 1986-2010 ($Val2$) for the two French basins, and 1978-1993 and 1994-2008 for the Austrian one
- Objective functions (and criteria): Root Mean Square Error (RMSE), Nash-Sutcliffe Efficiency (NSE), Kling Gupta Efficiency (KGE and KGE')

	Basin	Model	Per_Cal	Per_Val	OF_Name	OF_Value	X1	X2	X3	X4	X5	X6	C1	C2	RMSE	NSE	KGE	KGE2
1	Obs_X0310010	RunModel_GR4J	1961−1985	1961−1985	RMSE[Q]	1.704	1671	4.141	4501.000	1.66	NA	NA	NA	NA	1.704	0.111	0.035	0.033
2	Obs_X0310010	RunModel_GR4J	1961−1985	1986−2010	RMSE[Q]	1.704	1671	4.141	4501.000	1.66	NA	NA	NA	NA	1.474	0.096	0.106	0.088
3	Obs_X0310010	RunModel_GR4J	1961−1985	1961−1985	NSE[Q]	0.111	1671	4.141	4501.000	1.66	NA	NA	NA	NA	1.704	0.111	0.035	0.033
4	Obs_X0310010	RunModel_GR4J	1961−1985	1986−2010	NSE[Q]	0.111	1671	4.141	4501.000	1.66	NA	NA	NA	NA	1.474	0.096	0.106	0.088
5	Obs_X0310010	RunModel_GR4J	1961−1985	1961−1985	KGE[Q]	0.132	2595	0.110	8.399	2.00	NA	NA	NA	NA	2.292	−0.609	0.132	0.130
6	Obs_X0310010	RunModel_GR4J	1961−1985	1986−2010	KGE[Q]	0.132	2595	0.110	8.399	2.00	NA	NA	NA	NA	2.037	−0.726	0.199	0.202
7	Obs_X0310010	RunModel_GR4J	1961−1985	1961−1985	KGE'[Q]	0.133	2723	0.000	8.313	2.00	NA	NA	NA	NA	2.262	−0.566	0.130	0.133
8	Obs_X0310010	RunModel_GR4J	1961−1985	1986−2010	KGE'[Q]	0.133	2723	0.000	8.313	2.00	NA	NA	NA	NA	1.998	−0.660	0.206	0.207
9	Obs_X0310010	RunModel_GR4J	1986−2010	1961−1985	RMSE[Q]	1.462	2702	−3.552	4772.815	1.63	NA	NA	NA	NA	1.716	0.099	−0.007	0.004
10	Obs_X0310010	RunModel_GR4J	1986−2010	1986−2010	RMSE[Q]	1.462	2702	−3.552	4772.815	1.63	NA	NA	NA	NA	1.462	0.110	0.069	0.070
11	Obs_X0310010	RunModel_GR4J	1986−2010	1961−1985	NSE[Q]	0.110	2702	−3.552	4772.815	1.63	NA	NA	NA	NA	1.716	0.099	−0.007	0.004
12	Obs_X0310010	RunModel_GR4J	1986−2010	1986−2010	NSE[Q]	0.110	2702	−3.552	4772.815	1.63	NA	NA	NA	NA	1.462	0.110	0.069	0.070
13	Obs_X0310010	RunModel_GR4J	1986−2010	1961−1985	KGE[Q]	0.219	2352	0.071	31.050	2.00	NA	NA	NA	NA	2.125	−0.383	0.113	0.117
14	Obs_X0310010	RunModel_GR4J	1986−2010	1986−2010	KGE[Q]	0.219	2352	0.071	31.050	2.00	NA	NA	NA	NA	1.849	−0.422	0.219	0.214
15	Obs_X0310010	RunModel_GR4J	1986−2010	1961−1985	KGE'[Q]	0.219	2368	−0.181	30.569	1.99	NA	NA	NA	NA	2.111	−0.365	0.102	0.120

FIGURE 3.1: Dataframe of results including (going from the column on the left to the one on the right hand side) the basin and the model names, the calibration and the validation periods, the objective function name and value, the model parameters values and finally the evaluation criteria names and values.

Then depending on the analysis that has to be performed, I extracted from the just mentioned dataframe the results needed (see Appendix B for the whole dataframe).

With the goal of underlining the role of the four coefficients used to compute the two Kling-Gupta criteria (KGE and KGE' - see chapter 1.2.4), I tabled the ones got after a series of simulations performed with GR4J + Cemaneige on the French Alpine basin (X0310010). I performed different calibrations changing calibration period and objective function; for each of them the performance of the model has then been evaluated both with KGE and KGE', extracting from the results got with *airGR* the values of the coefficients r, α, β and γ (see Fig. 3.2).
In order to highlight how these parameters affect the error criteria values, I computed the difference of both the criteria values (KGE and KGE') and the coefficients, with their optimal values, which is 1 in all the cases (see Fig. 3.3).

Per Cal	OF Name	Per Val	KGE Value	KGE2 Value	r	alpha	beta	gamma
1961-1985	KGE[Q]	1961-1985	0.928	0.928	0.928	0.999	1.001	0.998
1961-1985	KGE[Q]	1986-2010	0.843	0.888	0.93	1.127	1.059	1.063
1961-1985	KGE'[Q]	1961-1985	0.93	0.93	0.93	0.999	1.000	0.999
1961-1985	KGE'[Q]	1986-2010	0.827	0.874	0.924	1.144	1.057	1.083
1986-2010	KGE[Q]	1961-1985	0.857	0.891	0.925	0.892	0.944	0.945
1986-2010	KGE[Q]	1986-2010	0.926	0.926	0.926	1.005	0.997	1.008
1986-2010	KGE'[Q]	1961-1985	0.855	0.888	0.925	0.887	0.949	0.935
1986-2010	KGE'[Q]	1986-2010	0.926	0.926	0.926	1.006	1.002	1.004

FIGURE 3.2: Coefficients used to calculate KGE and KGE' error criteria values obtained for different calibration processes

Per Cal	OF Name	Per Val	Dev KGE Value	Dev KGE2 Value	Dev r	Dev alpha	Dev beta	Dev gamma
1961-1985	KGE[Q]	1961-1985	0.072	0.072	0.072	0.001	0.001	0.002
1961-1985	KGE[Q]	1986-2010	0.157	0.112	0.070	0.127	0.059	0.063
1961-1985	KGE'[Q]	1961-1985	0.070	0.070	0.070	0.001	0	0.001
1961-1985	KGE'[Q]	1986-2010	0.173	0.126	0.076	0.144	0.059	0.083
1986-2010	KGE[Q]	1961-1985	0.143	0.109	0.075	0.108	0.056	0.055
1986-2010	KGE[Q]	1986-2010	0.074	0.074	0.074	0.005	0.003	0.008
1986-2010	KGE'[Q]	1961-1985	0.145	0.112	0.075	0.113	0.051	0.065
1986-2010	KGE'[Q]	1986-2010	0.074	0.074	0.074	0.006	0.002	0.004
Abs_dev_min			0.070	0.070	0.070	0.001	0	0.001
Abs_dev_max			0.173	0.126	0.076	0.144	0.059	0.083

FIGURE 3.3: Absolute differences between KGE value, KGE' value, the coefficients used to calculate the just mentioned error criteria values, and their optimal value (i.e. 1), for different calibration processes. The last two rows highlight the minimum (best case) and the maximum (worst case) absolute value for each column.

As expected, the Fig.3.3 shows that the best KGE and KGE' criteria values (the ones closer to 1, therefore with minimum deviations) correspond to the best values of r, α, β and γ (see the blue rectangles in the figure), and vice versa happens for the worst KGE and KGE' with the related coefficients (red rectangles).

Comparing the last four columns of the table in Fig. 3.3, it emerges that the Pearson coefficient is the parameter that on average brings the biggest error in the computation of the two criteria values. The r value results to be the least variable one within the set of calibrations performed: its distance from 1 ranges between 0.07 and 0.08, which means that the level of agreement in

temporal dynamics between measured and simulated discharges time series is between 92% and 93%.
Analysing the variability coefficients (α and γ) with respect to β, it can be noted that the distance of the optimal value from β is always smaller than (or very similar to) the one of α and γ, meaning that the bias ratio affect the error of KGE and KGE′ less than what the variability coefficients do. Additionally it can be noted that when α is close to 1, also β it is: as a consequence, their ratio is close to the unit too, and therefore also γ is not far from 1. β is bigger than one in the majority of the cases, hinting that the discharges are mostly overestimated (positive bias).

3.2 Parameters versus objective functions (only with GR4J + CemaNeige)

In order to understand how much the objective function and the calibration period affect the optimization of the model parameters, the latter have been plotted for each basin as function of the objective functions and calibration periods. Only the GR4J model plus CemaNeige has been considered for this section, therefore the parameters analyzed are x_1, x_2, x_3, x_4, c_1 and c_2.
In the very first analysis, the plots were got taking into account RMSE, NSE, KGE and KGE′. However, after a first scan of the charts, it has been noticed that the plots of the parameters obtained with RMSE and NSE were overlapping; hence, the plots related to RMSE has been removed. The reason why this happened is because NSE is function of RMSE (see chapter 1.2.4).

The meaning of the parameters characterizing GR4J and CemaNeige has been already explained in chapter 2. For the following analysis a particular reference will be done to the mean values of temperature, precipitation and discharges over the two calibration periods listed in chapter 2.6.5: for all the three basins the mean temperature increases of about 0.5 - 0.6 °C between the first and the second calibration periods (see Fig.2.10).

Before performing the following analysis, I used *airGRteaching* [Delaigue et al., 2018b][Delaigue et al., 2018a] to study how each parameter affects the characteristics of the hydrograph. *airGRteaching* is an add-on package of *airGR* which allows to access to the *GR* models through an interface that helps to analyse data and model results in an immediate way and without the need of advanced programming skills.

3.2.1 Parameter x_1

From a visual sensitivity analysis on airGRteaching [Delaigue et al., 2018b][Delaigue et al., 2018a] (see Fig.3.4 and 3.5)...

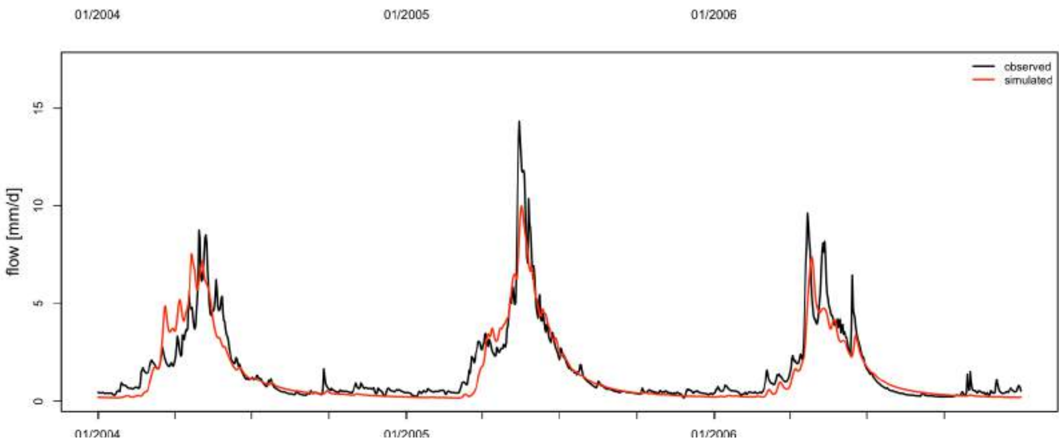

FIGURE 3.4: $x1 = 250mm$
Simulated hydrograph (in orange) of a random basin through *airGR teaching* to highlight how the parameter x_1 affects the simulated discharges

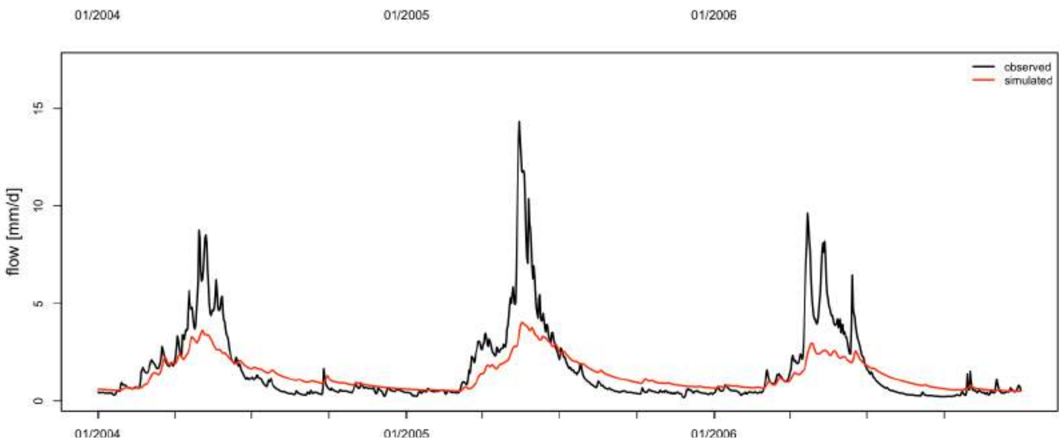

FIGURE 3.5: $x1 = 2200mm$
Simulated hydrograph (in orange) of a random basin through *airGR teaching* to highlight how the parameter x_1 affects the simulated discharges

For high values of x_1 the plot of the simulated discharges tends to flatten: both the high and the low flows smooth. Decreasing the parameter value, instead, increases the streamflow and in particular the higher flows are amplified.

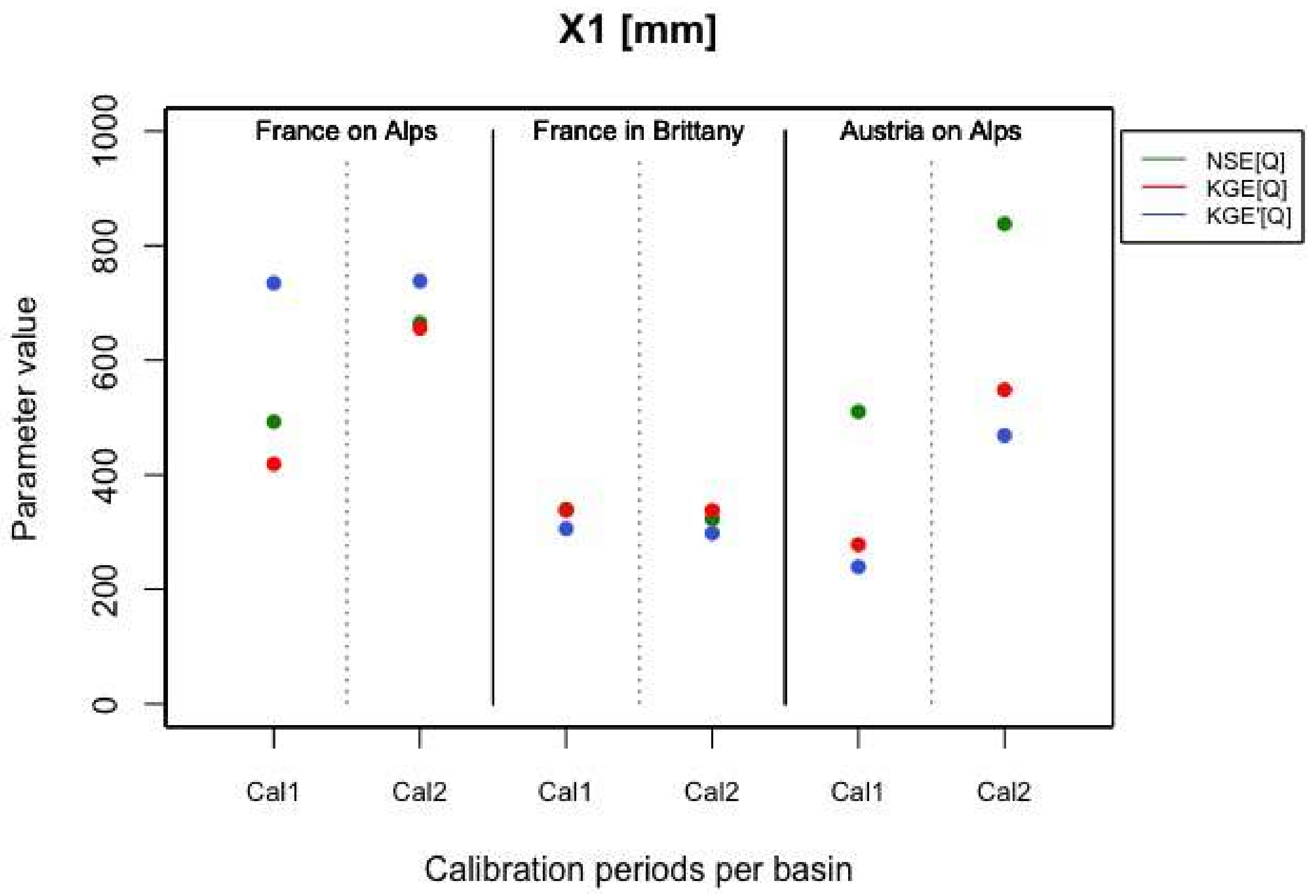

FIGURE 3.6: Parameter x_1 obtained with different objective functions and with two calibration periods for each basin ($Cal1$ and $Cal2$ are the 1^{st} and the 2^{nd} half of the sample time-series)

For both the first (X0310010 - Alps in France) and the third basin (200105 - Alps in Austria), x_1 increases visibly with the increasing of the temperature between $Cal1$ and $Cal2$ (Fig.3.6): a bigger x_1 means reduced high flows, which could be consistent with a warmer climate. Additionally, for the Austrian basin, also the precipitation average over the two calibration periods decreases from $Cal1$ to $Cal2$, leading to a drier condition, and probably causing a bigger gap between the x_1 value for the two calibration periods.
Regarding the Brittany basin, despite the increasing temperature, the value of the parameter concerned remains more or less the same: this could be justified with the higher mean of rainfall that, increasing the water input in

the catchment, could compensate in this case the probable increase of evapotransporation caused by the increasing T.

3.2.2 Parameter x_2

When x_2 is negative, the model is simulating the import of water in the underground (deep aquifers or surrounding catchment), and viceversa if positive; if its value is around zero, there is not exchange of water.

From a visual sensitivity analysis on airGRteaching [Delaigue et al., 2018b][Delaigue et al., 2018a] (see Fig.3.7 and 3.8)...

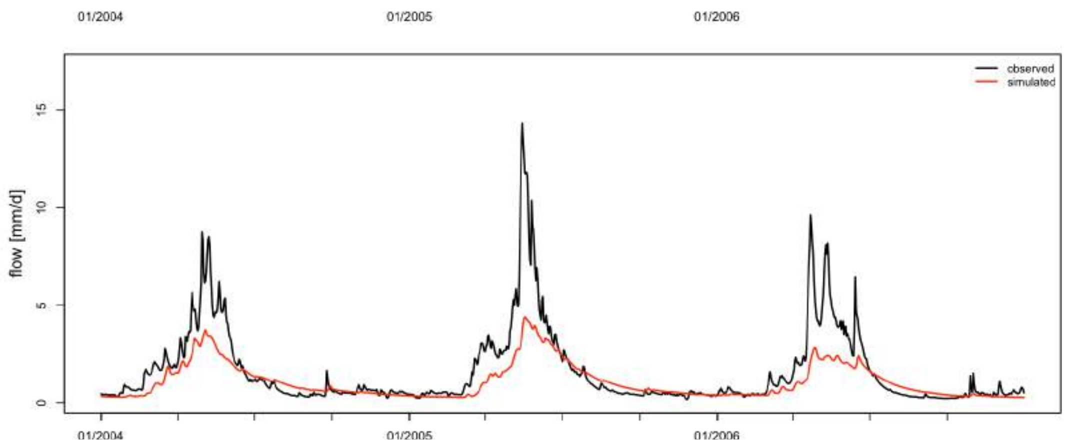

FIGURE 3.7: $x2 = -4mm/day$
Simulated hydrograph (in orange) of a random basin through *airGR teaching* to highlight how the parameter x_2 affects the simulated discharges

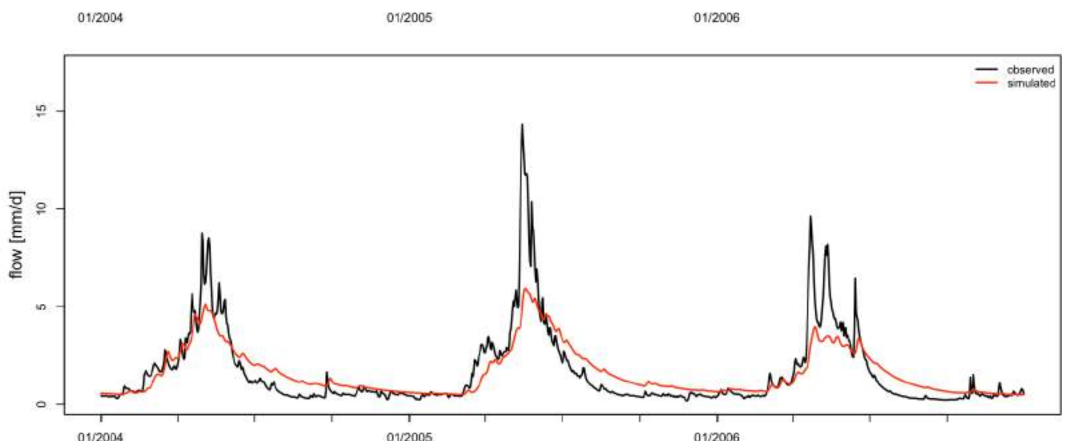

FIGURE 3.8: $x2 = 4mm/day$
Simulated hydrograph (in orange) of a random basin through *airGR teaching* to highlight how the parameter x_2 affects the simulated discharges

Decreasing the parameter values below zero leads to smoother and down-shifted simulated hydrographs. On the contrary, for increasing values of the parameter above zero, the plot turns to move upward, gaining a more peaky aspect.

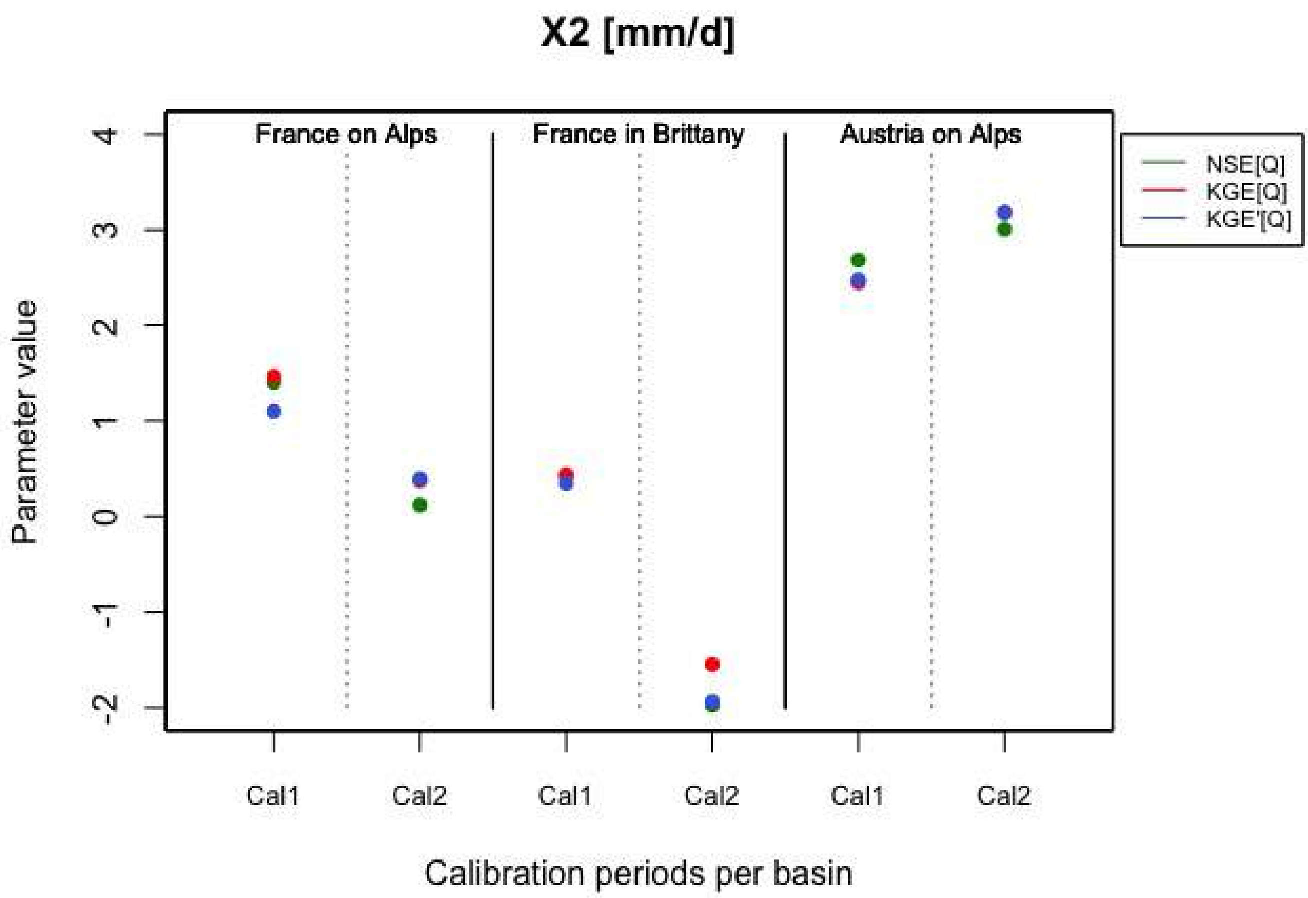

FIGURE 3.9: Parameter x_2 obtained with different objective functions and with two calibration periods for each basin ($Cal1$ and $Cal2$ are the 1^{st} and the 2^{nd} half of the sample time-series)

x_2 decreases for the first and second basins (see Fig. 3.9): in the latter case, though, it turns to be negative, indicating an "outflow" from the considered watershed: a reason for this could be that the lack of snow and the higher temperature, and therefore evapotranspiration, in Brittany do not help the river to recharge and the global water-balance results to be an outgoing flow.

	Cal1	***Cal2***
Q_{obs}/P_{obs}	0.52	0.46

TABLE 3.1: Water yield calculated over each one of the two calibration periods ($Cal1$ and $Cal2$)

The computation of the water yield (see Tab.3.1), namely the ratio between the mean of the observed streamflow and rainfall values, can help to

understand better what really happens. The value related to the second calibration period results indeed to be lower than the one of $Cal1$: from Tab. 2.10 can indeed be seen that the discharge mean values are almost the same for the two periods, despite the fact that the the precipitations increase. This suggests an higher evapotranspiration and a more arid climate for the second part of the time-series, which reflects in the negative value of x_2.
For the Alpine French basin instead, the decrease of x_2 means a simulated hydrograph moved downward, therefore with lower discharges, consistently with the increase of mean temperature between the two calibration periods. On the other hand, for the Austrian catchment, the parameter value grows from about 2.5 to about 3.5 [mm/day]: since the temperature increases as in the first basin, the trend expected would be the same as that one observed in the $Basin1$: the fact that the trend instead is not the same could be due to a different compensation from the other parameters on the two basins.

3.2.3 Parameter x_3

From a visual sensitivity analysis on airGRteaching [Delaigue et al., 2018b][Delaigue et al., 2018a] (see Fig.3.10 and 3.11)...

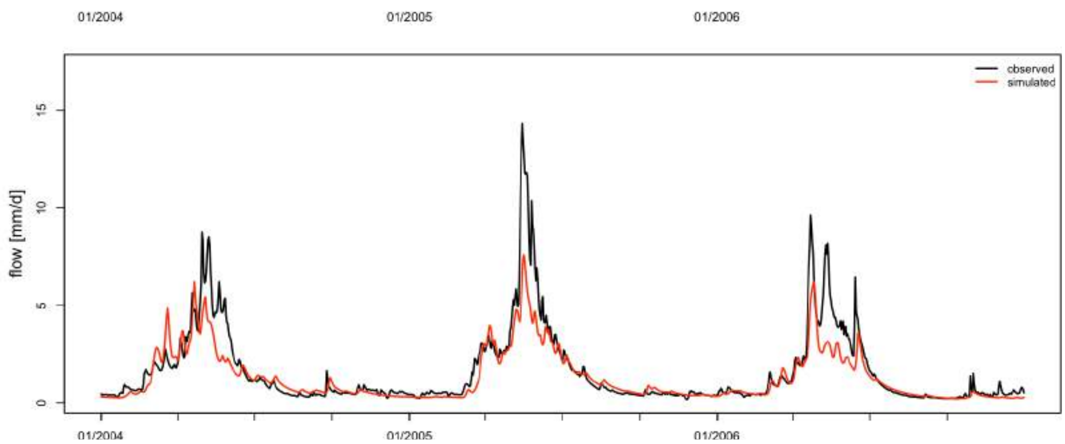

FIGURE 3.10: $x3 = 100mm$
Simulated hydrograph (in orange) of a random basin through *airGR teaching* to highlight how the parameter x_3 affects the simulated discharges

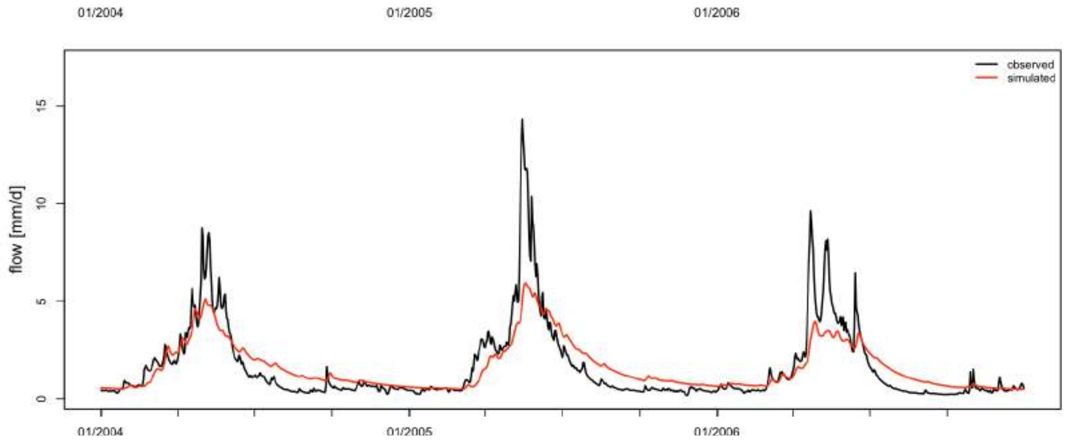

FIGURE 3.11: $x3 = 900mm$
Simulated hydrograph (in orange) of a random basin through *airGR teaching* to highlight how the parameter x_3 affects the simulated discharges

x_3 affects the simulated hydrograph amplifying the peaks in case of low values of the parameter, and flatting them in case of higher values.

The low value of x_3 for the Austrian basin (see Fig. 3.12) suggests an hydrograph with evident peaks: this could be due, for instance, to the smaller dimension of the river that turns it to be more sensitive to rainfall and related flash flood events.

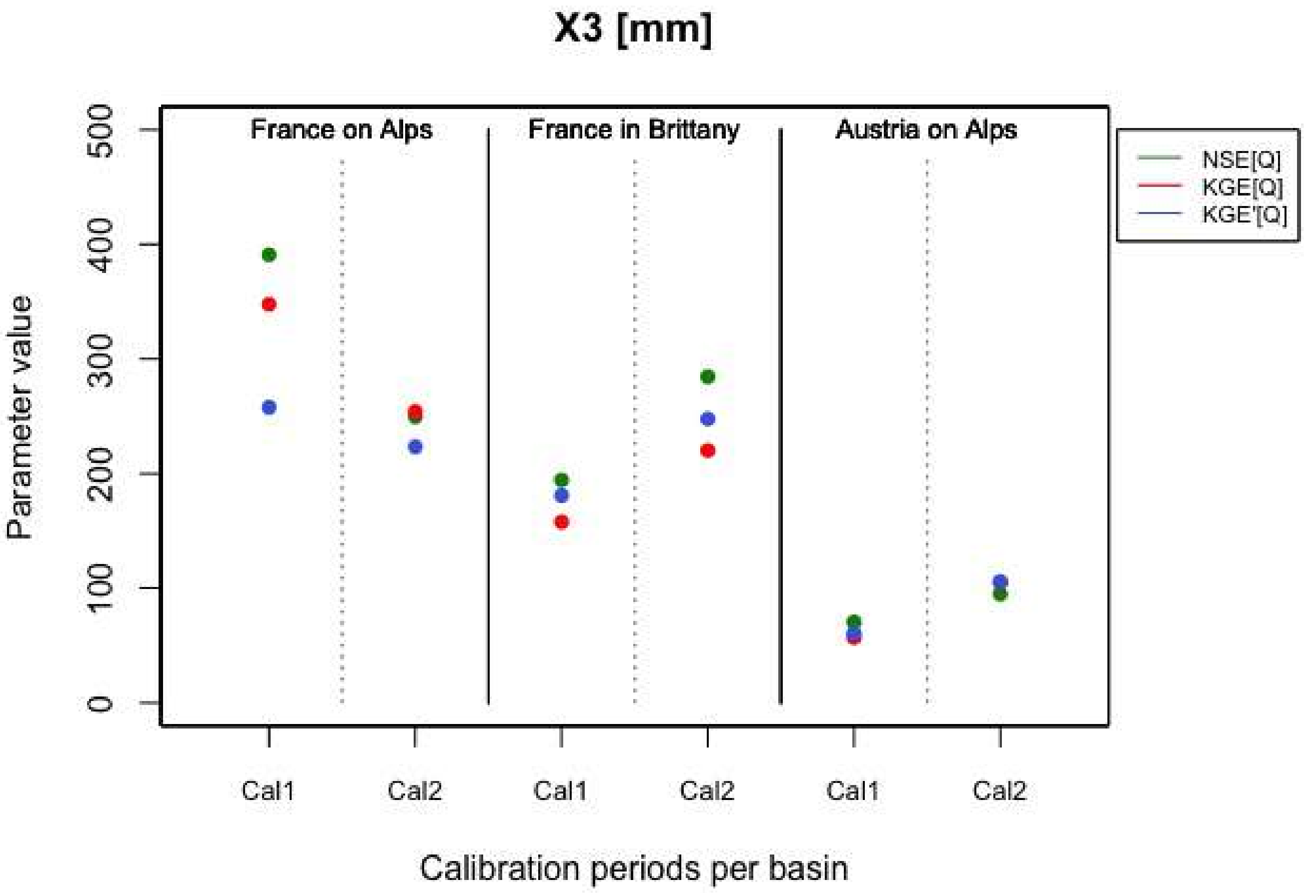

FIGURE 3.12: Parameter x_3 obtained with different objective functions and with two calibration periods for each basin ($Cal1$ and $Cal2$ are the 1^{st} and the 2^{nd} half of the sample time-series)

3.2.4 Parameter x_4

From a visual sensitivity analysis on airGRteaching [Delaigue et al., 2018b][Delaigue et al., 2018a](see Fig.3.13 and 3.14)...

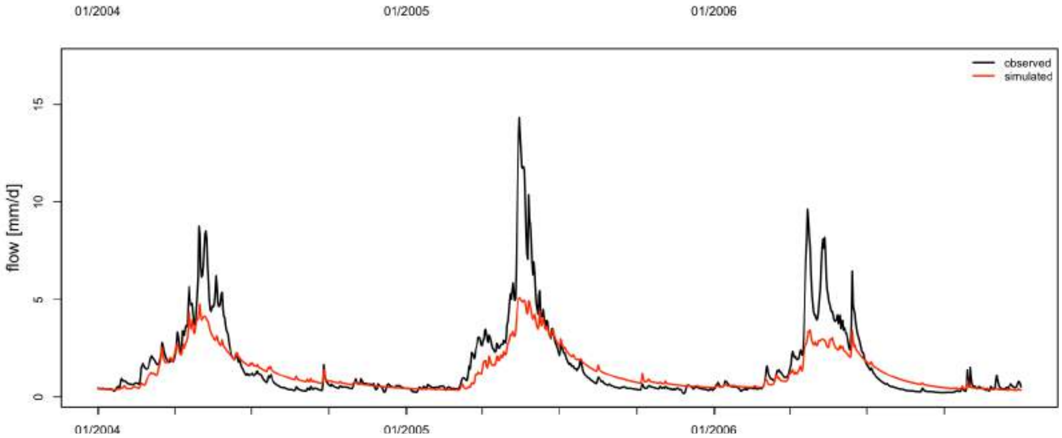

FIGURE 3.13: $x4 = 1.5day$
Simulated hydrograph (in orange) of a random basin through *airGR teaching* to highlight how the parameter x_4 affects the simulated discharges

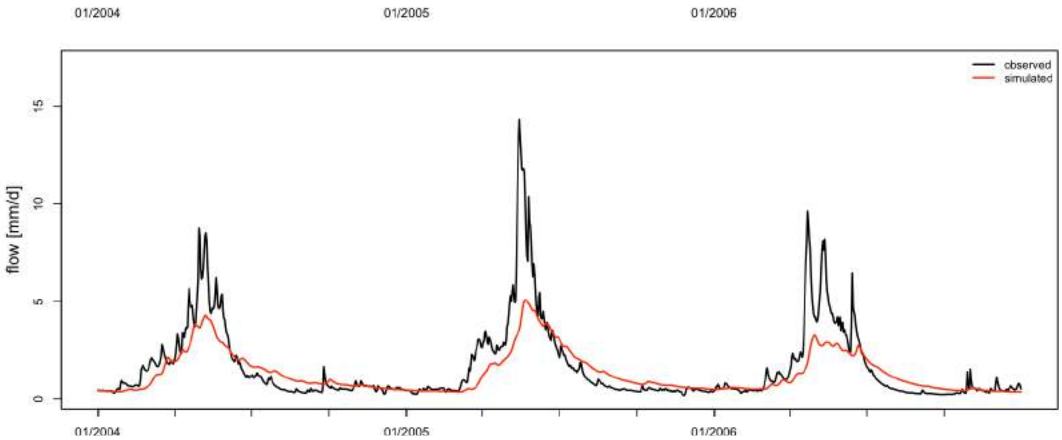

FIGURE 3.14: $x4 = 8.5day$
Simulated hydrograph (in orange) of a random basin through *airGR teaching* to highlight how the parameter x_4 affects the simulated discharges

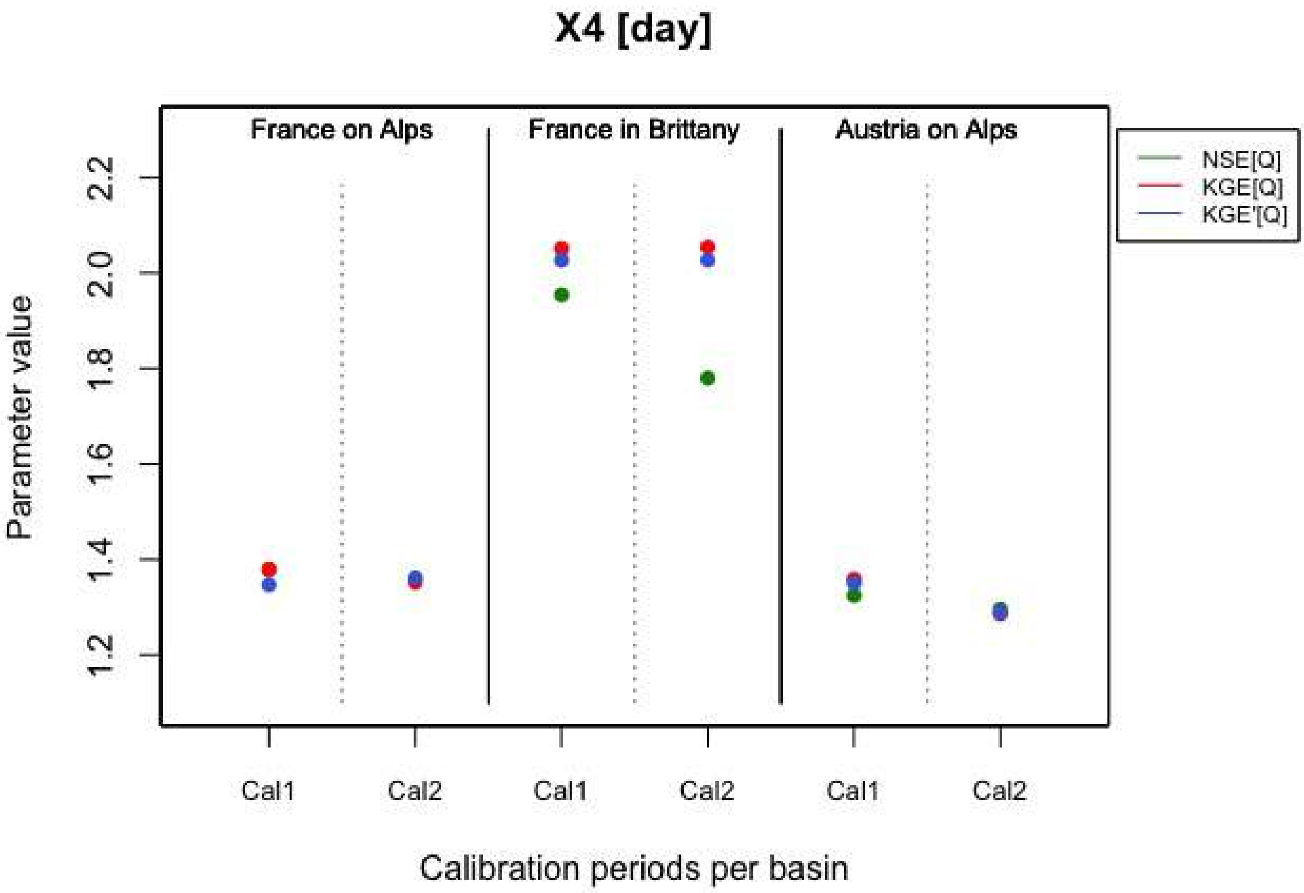

FIGURE 3.15: Parameter x_4 obtained with different objective functions and with two calibration periods for each basin ($Cal1$ and $Cal2$ are the 1^{st} and the 2^{nd} half of the sample time-series)

This parameter affects the smoothness of the simulated hydrograph and it slightly governs its horizontal position, i.e. delay, with respect to the observed streamflows: for bigger values of x_4, the plot results to be less peaky and shifted on the right; viceversa happens for smaller values.
The x_4 value results to be visibly higher in the Breton basin than in the other two ones (see Fig. 3.15): therefore in this case the delay between input of water in the catchment and discharge is higher.

3.2.5 Parameter c_1

From a visual sensitivity analysis on airGRteaching [Delaigue et al., 2018b][Delaigue et al., 2018a] (see Fig.3.16 and 3.17)...

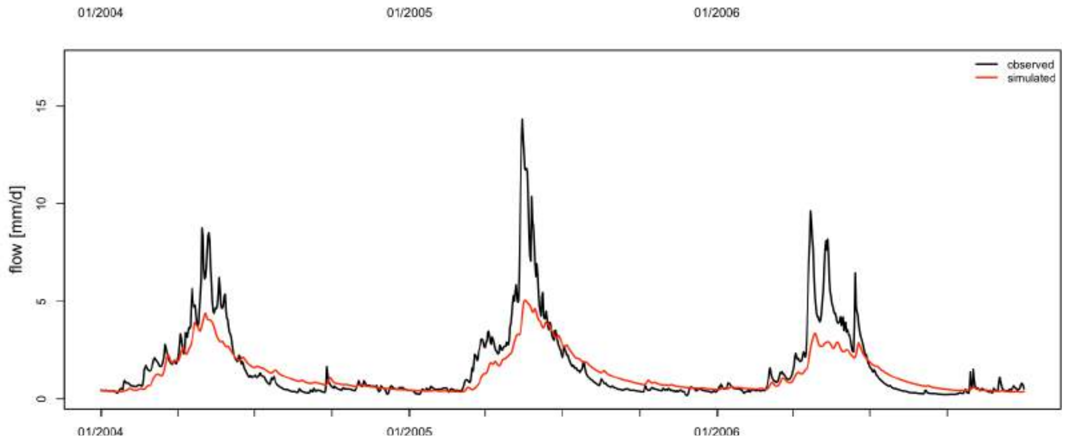

FIGURE 3.16: $c1 = 0.1[-]$
Simulated hydrograph (in orange) of a random basin through *airGR teaching* to highlight how the parameter c_1 affects the simulated discharges

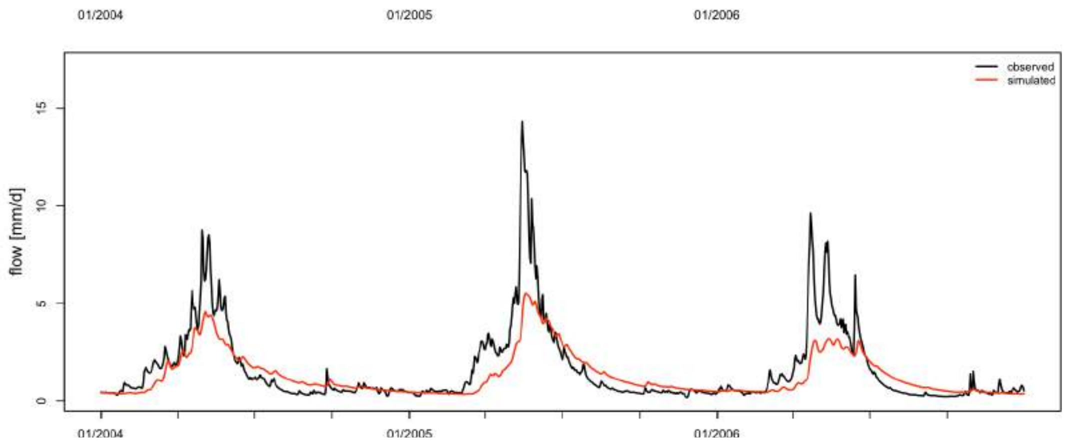

FIGURE 3.17: $c1 = 0.9[-]$
Simulated hydrograph (in orange) of a random basin through *airGR teaching* to highlight how the parameter c_1 affects the simulated discharges

The c_1 parameter does not have a clear physical meaning and impact on simulations that can be identified on a visual analysis. This reflects on its variability, which can not be explained from Fig. 3.18. According to Dr. Thirel this parameter is difficult to interpret.

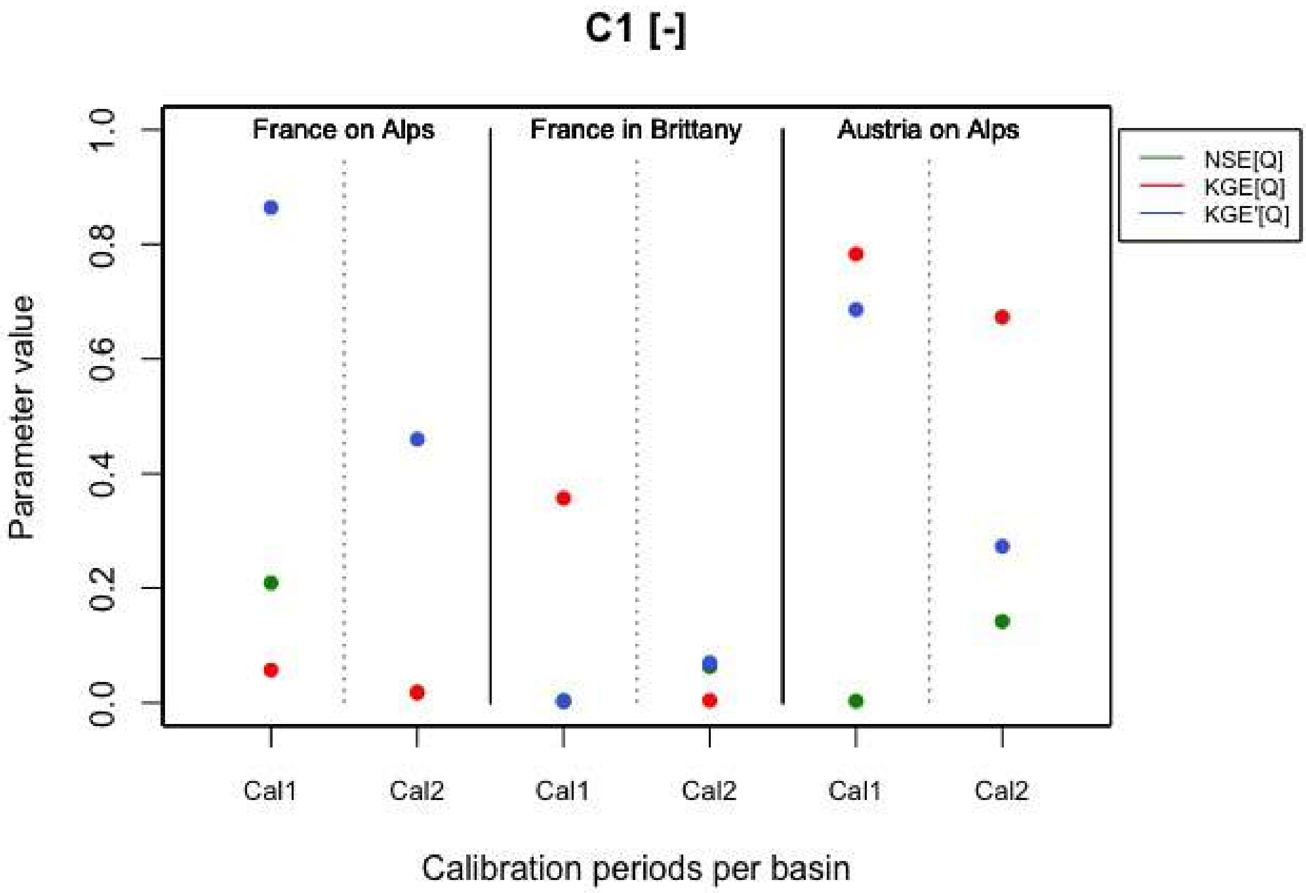

FIGURE 3.18: Parameter c_1 obtained with different objective functions and with two calibration periods for each basin ($Cal1$ and $Cal2$ are the 1^{st} and the 2^{nd} half of the sample time-series)

3.2.6 Parameter c_2

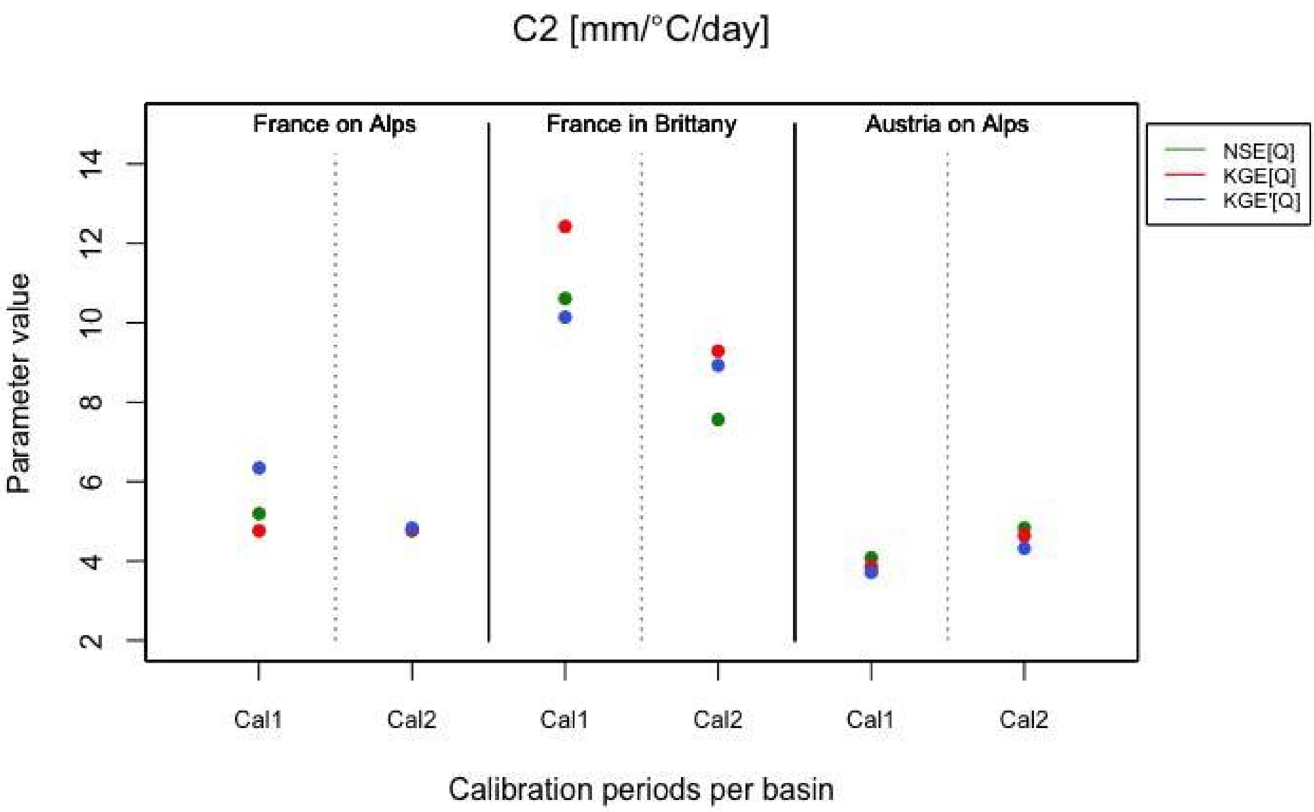

FIGURE 3.19: Parameter c_2 obtained with different objective functions and with two calibration periods for each basin ($Cal1$ and $Cal2$ are the 1^{st} and the 2^{nd} half of the sample time-series)

The c_2 value represents the degree-day melt coefficient: small values of c_2 suggest a slow melting process, therefore a slow source of water for the catchment. Viceversa happens for higher values of the parameter.

The considerable difference between the c_2 values got for the Brittany basin (J5412110) and those obtained for the other two basins is due to the lack of snow in the Breton watershed. As a result, CemaNeige responds trying to compensate the lack of water increasing the snow melt velocity and therefore the water input to the catchment. For this reason the c_2 parameter results to be higher than in other circumstances.

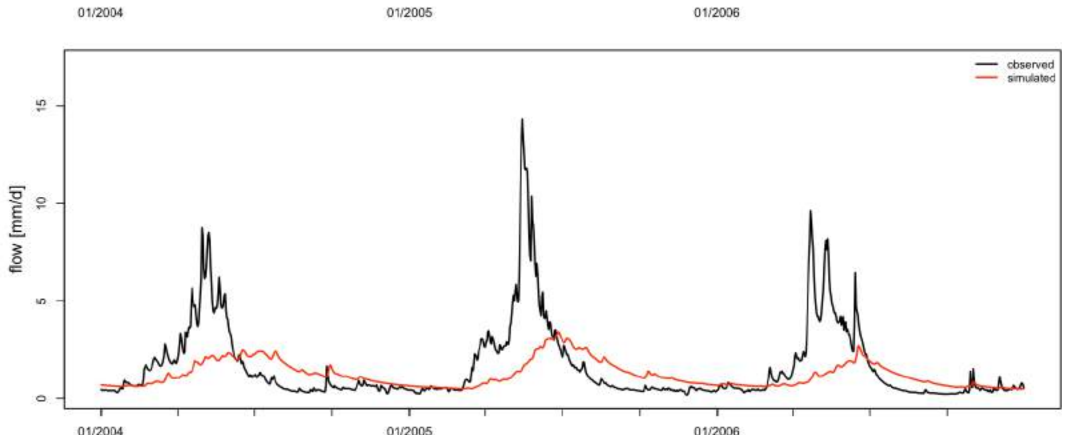

FIGURE 3.20: $c2 = 1mm/°C/day$
Simulated hydrograph (in orange) of a random basin through *airGR teaching* to highlight how the parameter c_2 affects the simulated discharges

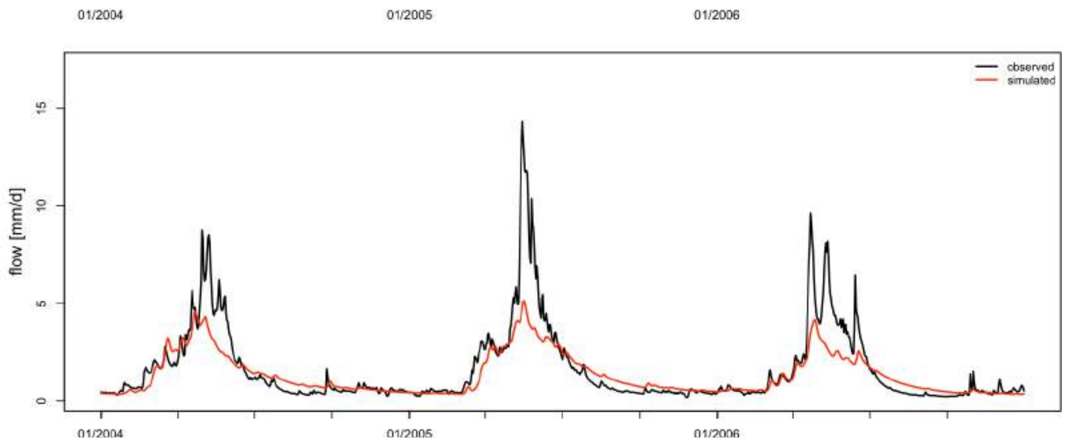

FIGURE 3.21: $c2 = 1mm/°C/day$
Simulated hydrograph (in orange) of a random basin through *airGR teaching* to highlight how the parameter c_2 affects the simulated discharges

Chapter 4

Snow analysis

In the chapter that follows, the snowfall over the three catchments will be studied, mainly focusing on two main topics: the impact that the different calibrations of CemaNeige can have on its outputs and the importance of coupling the snow model with the hydrological ones.

4.1 Snow melt simulated with different parameters

In order to show how the output of Cemaneige can change depending on the different calibrations performed, and therefore the different set of parameters used, the snow melt got for the two nival basins has been plotted in Fig. 4.1 and 4.2.

The bar-plots refer to two calibration periods (see chapter 3.1 for details) which are differentiated using orange and teal blue: the daily snow melts output from the snow model have been summed up to get monthly time series that, successively, have been averaged for each month over the years.

For each calibration period, the objective functions used are *NSE*, *KGE* and *KGE'*.

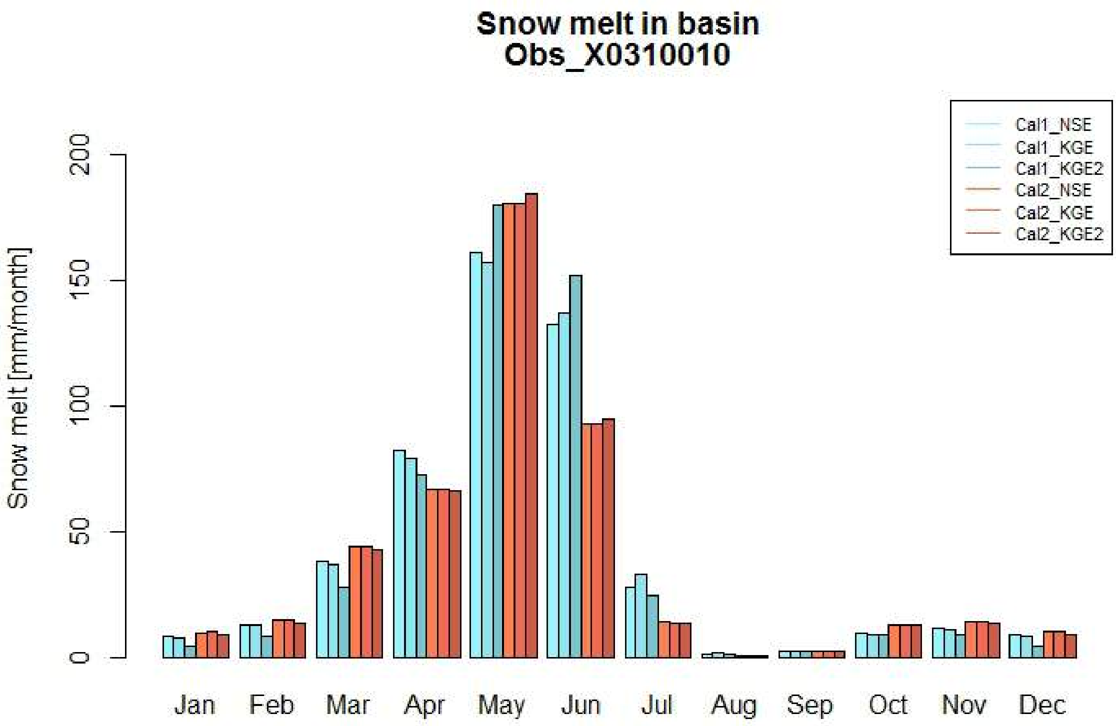

FIGURE 4.1: Snow melt output of CemaNeige for the basin X0310010 (Alps in France): each bar is related to one of the calibration processes obtained combining the two calibration periods ($Cal1$ and $Cal2$ are the 1^{st} and the 2^{nd} half of the sample time-series) and the three most meaningful objective functions (NSE, KGE, KGE')

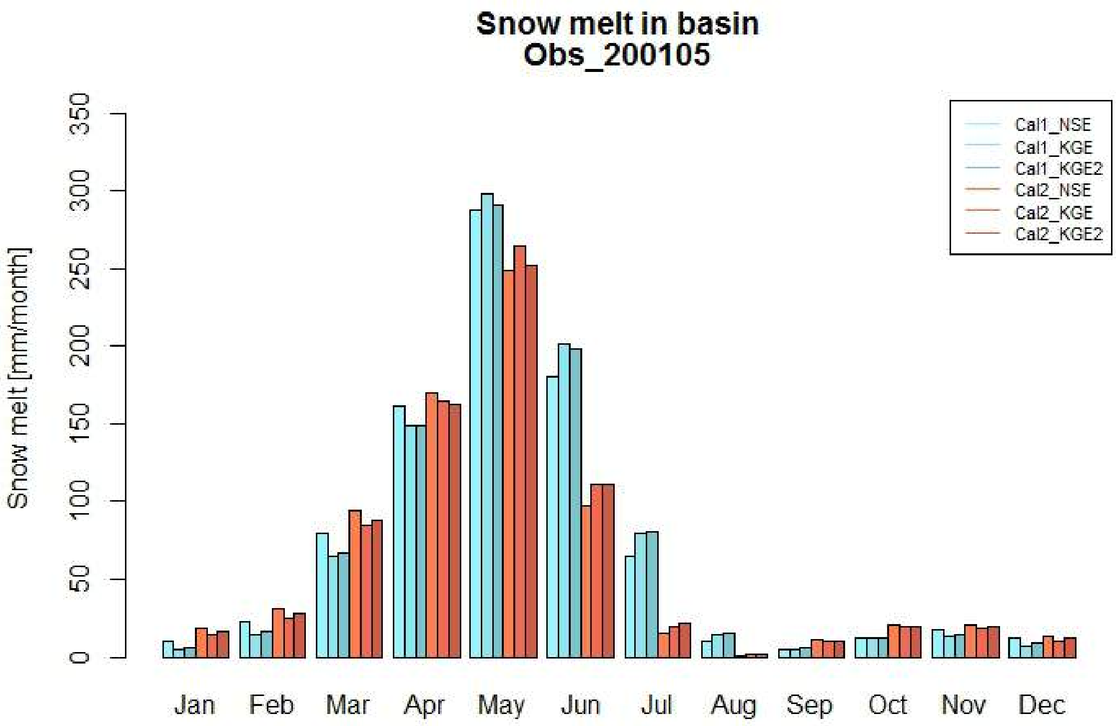

FIGURE 4.2: Snow melt output of CemaNeige for the basin J5412110 (Alps in Austria): each bar is related to one of the calibration processes obtained combining the two calibration periods ($Cal1$ and $Cal2$ are the 1^{st} and the 2^{nd} half of the sample time-series) and the three most meaningful objective functions (NSE, KGE, KGE')

From the two plots in Fig.4.1 and 4.2 it emerges that the snow melt increases during the spring, with the first warmer temperatures, and then it decreases from May to August: probably this latter trend is due to a decrease of solid precipitation during the previous season, which feeds the stored snow in smaller quantity and then the snow that can actually melt decreases.

Additionally, observing the bars in June, for both the plots, it can be noted how the calibration periods (and then the climates) affect the simulated snow melt: between the $Cal1$ and $Cal2$, in fact, there is a difference of about 80-100 mm/day, which is a variance that should be taken into account.
Furthermore, it's interesting to see how the objective functions in the month of June of Fig.4.1 already impacts the snow melt. The snow melt got calibrating CemaNeige with NSE over $Cal1$ differs indeed from the one obtained with KGE' of about 20 mm/day.

An additional observation can be done referring to the French basin on Alps (X0310010) and comparing the snow melt values obtained calibrating the CemaNeige over $Cal2$, with the related $c2$ values (plotted in Fig. 3.19): within each month indicated in Fig.4.1, the three reddish bars have more or less the same heights, meaning that the calibrations performed with the three different objective functions lead to simulate similar snow melt amounts; it's interesting to see that this fact is consistent with the $c2$ parameter, the values of which, in fact, almost coincide for the second calibration period (overlapping points in Fig. 3.19 for the French Alpine basin and $Cal2$).

4.2 Solid precipitation simulated with different parameters

Differently from the snow melt, the solid precipitation output of the CemaNeige model is not dependent on the parameters, as shown in Fig.4.3 and 4.4: in fact the solid fraction of precipitation P_{sol} is function only of the temperature and of the mean elevation of the catchment (see Fig. 2.4.b.). For both the nival basins, in fact, for each month, the bars related to the same calibration period have the same height, while they differ if related to $Cal1$ rather than $Cal2$.
Therefore, specifying the objective function used for the plots in the paragraph 2.8 was not relevant.

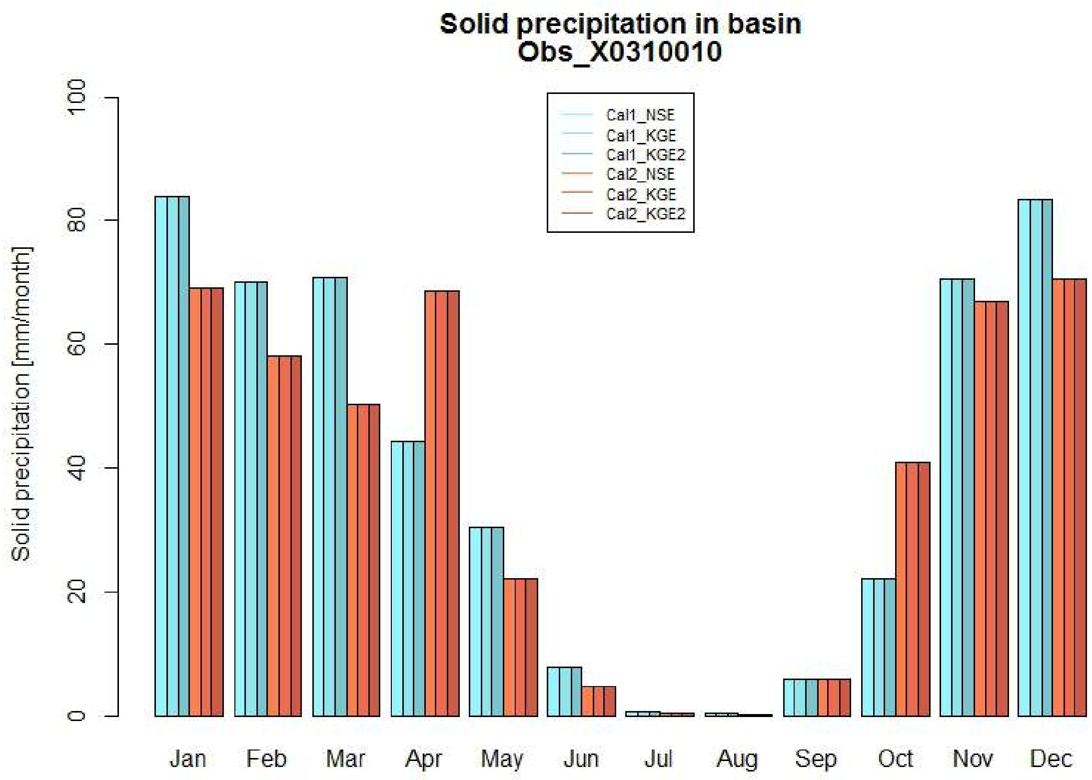

FIGURE 4.3: Solid precipitation output of CemaNeige for the basin X0310010 (Alps in France): each bar is related to one of the calibration processes obtained combining the two calibration periods ($Cal1$ and $Cal2$ are the 1^{st} and the 2^{nd} half of the sample time-series) and the three most meaningful objective functions (NSE, KGE, KGE')

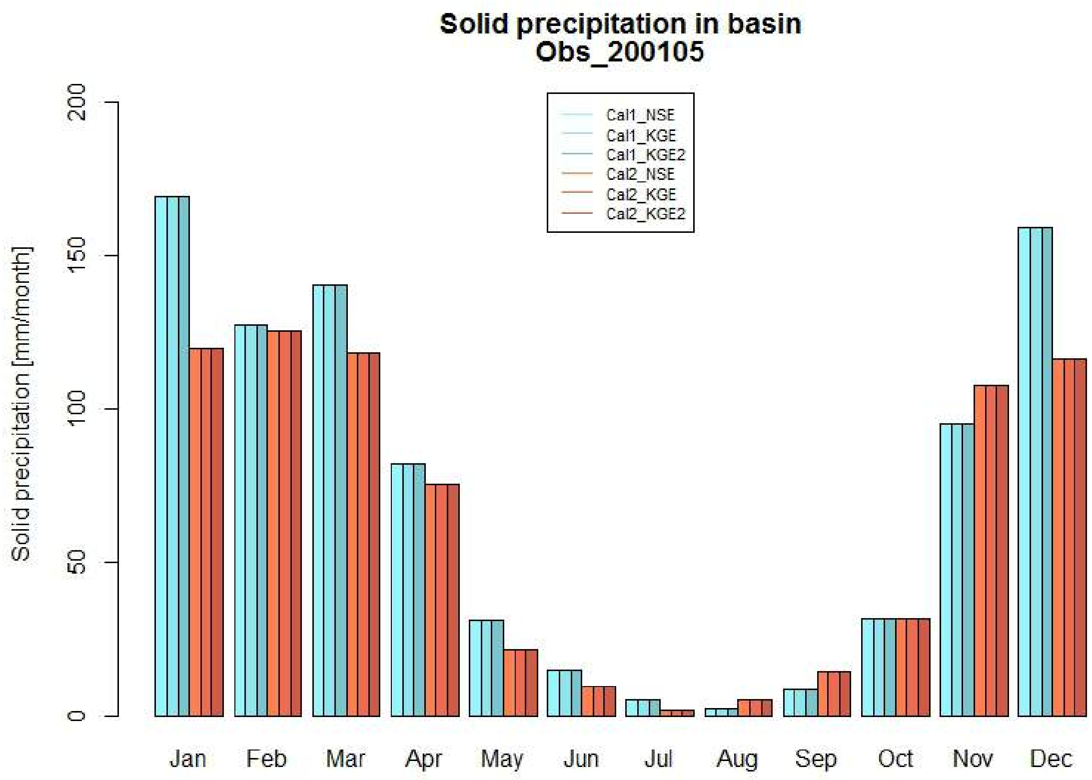

FIGURE 4.4: Solid precipitation output of CemaNeige for the basin J5412110 (Alps in Austria): each bar is related to one of the calibration processes obtained combining the two calibration periods (*Cal*1 and *Cal*2 are the 1^{st} and the 2^{nd} half of the sample time-series) and the three most meaningful objective functions (NSE, KGE, KGE')

4.3 The importance of CemaNeige for nival basins

With the goal of verifying, and understanding, which one of the three catchment flows need the snow model to be simulated, different plots have been done for each one of the simulations performed with GR6J and GR4J + CemaNeige. In particular simulated hydrographs have been compared with the observed ones. The selection of GR6J and GR4J + CemaNeige has been done in order to compare results obtained by means of models with the same overall number of parameters but respectively without and with the snow module.

The results obtained for the French basin X0310010 are similar to those got for the Austrian one (200105): GR6J applied without CemaNeige gives poor

results, with the simulated hydrographs that are far from fitting the observed ones (see the example of the basin X0310010 in Fig.4.5 and 4.6), whereas the presence of the CemaNeige model coupled with GR4J (Fig.4.7 and Fig.4.8) allows to obtain higher quality simulated discharges.
Different is the behaviour of the third catchment: the quality of the results got for the Breton watershed J5412110 using GR4J+CemaNeige is good as well. However, with the GR6J model, the goodness-of-fit between simulated and observed flows is comparable with the one obtained with the presence of the snow model (see the example in Fig.4.9 and Fig.4.11 and the relative 2-years zooms in Fig.4.10 and 4.12); therefore the presence of the CemaNeige for the Breton catchment results to be a surplus.
Overall, solid precipitations are a fundamental process that has to be taken into account by means of the CemaNeige module on the basin X0310010 (French basin on the Alps) and 200105 (Austrian basin), while it can be neglected for the basin J5412110 since it does not add any value to the results.

The plots chosen as examples and reported in the figures 4.5 - 4.12 are obtained performing the simulations with both objective function and criteria equal to "KGE - Kling Gupta Efficiency".

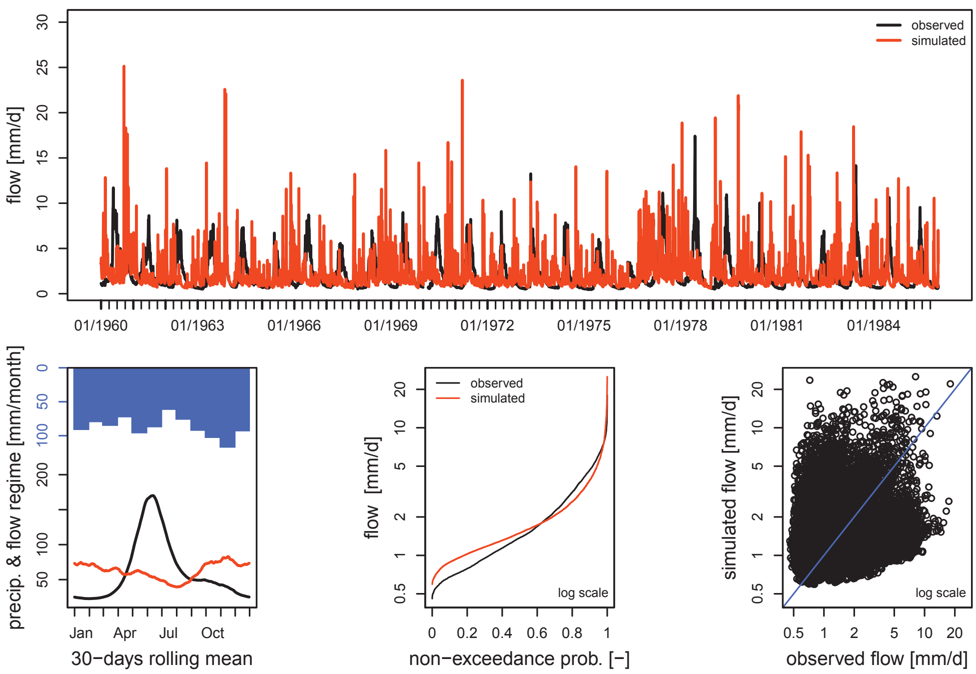

FIGURE 4.5: Example of simulation
Basin X0310010 (Alps)
Model GR6J

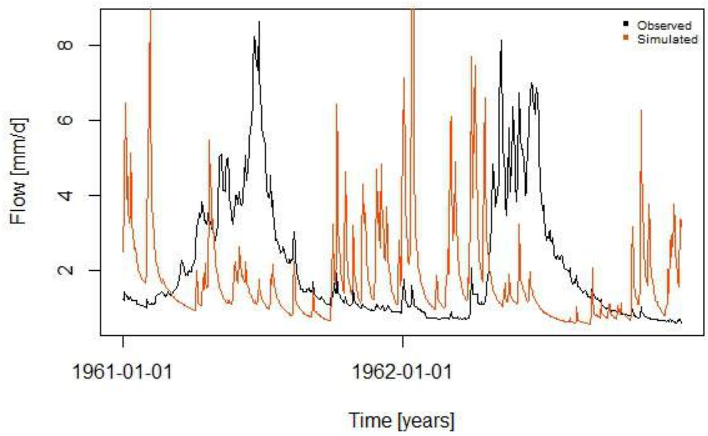

FIGURE 4.6: 1961 - 1962 zoom hydrograph
Basin X0310010 (Alps)
Model GR6J

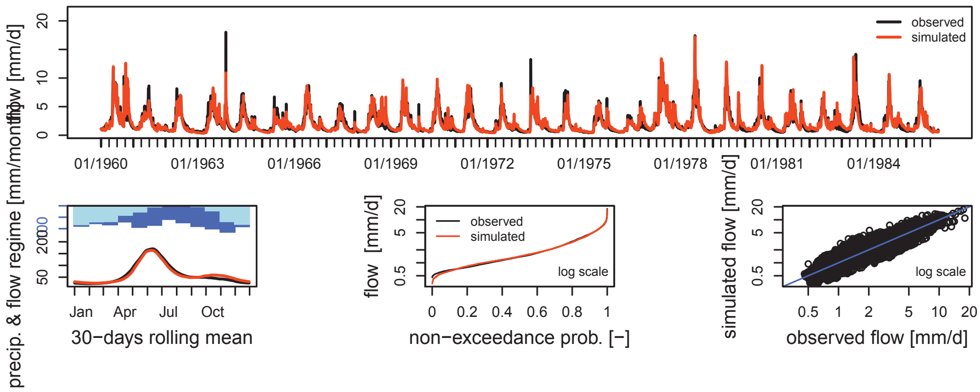

FIGURE 4.7: Example of simulation
Basin X0310010 (Alps)
Model GR4J+CemaNeige

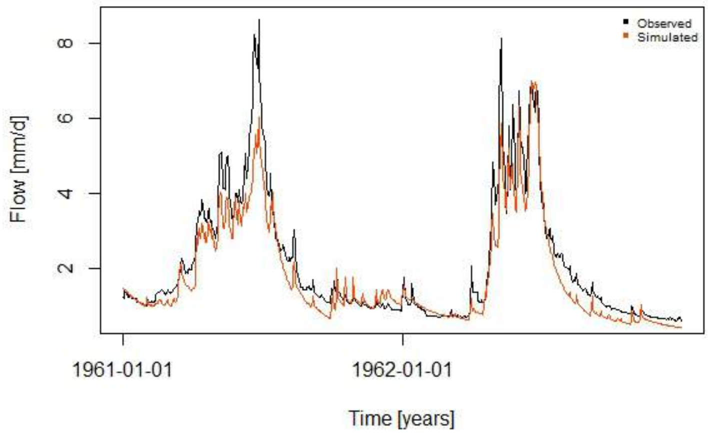

FIGURE 4.8: 1961 - 1962 zoom hydrograph
Basin X0310010 (Alps) - Model GR4J+CemaNeige

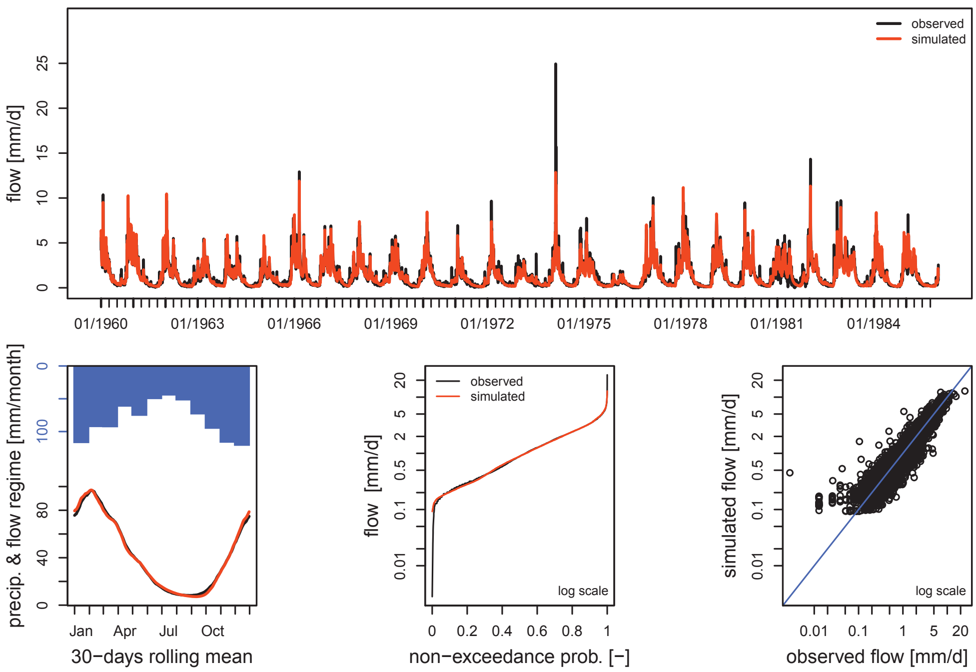

FIGURE 4.9: Example of simulation
Basin J5412110 (Brittany)
Model GR6J

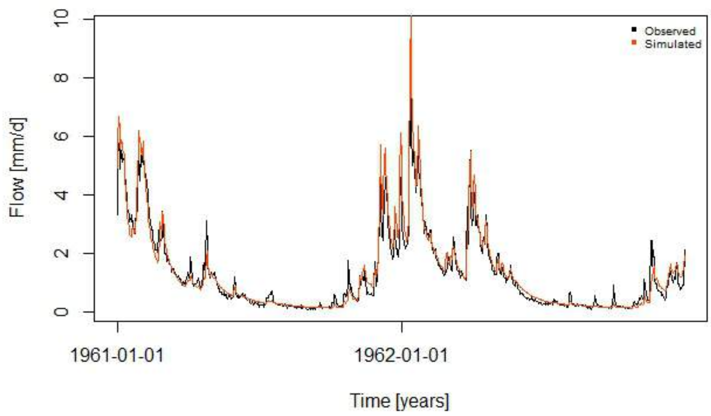

FIGURE 4.10: 1961 - 1962 zoom hydrograph
Basin J5412110 (Brittany) - Model GR6J

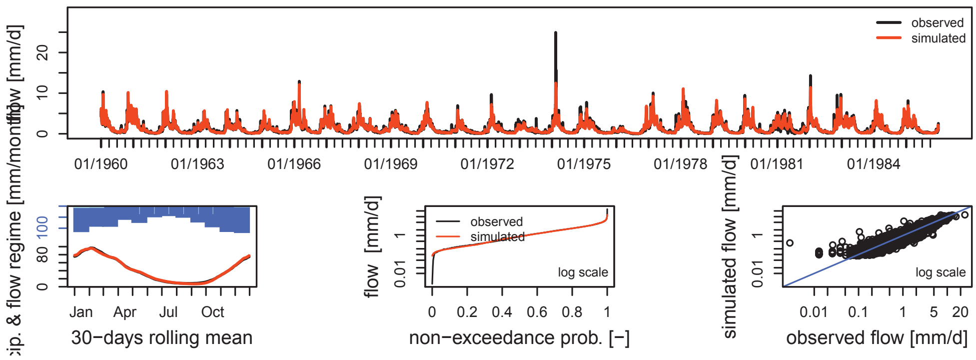

FIGURE 4.11: Example of simulation
Basin J5412110 (Brittany)
Model GR4J+CemaNeige

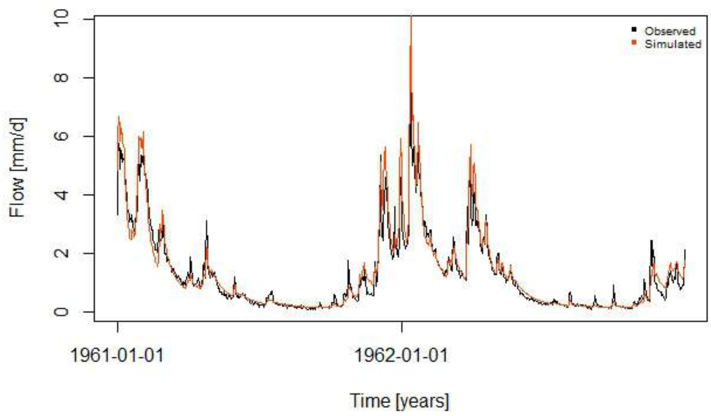

FIGURE 4.12: 1961 - 1962 zoom hydrograph
Basin J5412110 (Brittany) - Model GR4J+CemaNeige

4.4 Identification of the best model for each basin

The purpose of this section is identifying the model that simulates the best discharge over the three catchments analyzed. Due to the results got in chapter 4.3, we expect to get the model GR4J+CemaNeige as the best performing one on the two Alpine basins.

4.4.1 Analysis over the three basins

The mean of each criterion (computed over the validation periods) of the simulations performed with the parameters obtained in all the calibration experiments, has been calculated for all the simulations run over the same basin and with the same model (see Fig. 4.13).

The criteria formulas listed in paragraph 1.2.4 are rather self-explicit for the comprehension of the values in Fig.4.13. As already mentioned, the best values to aim at are:

- RMSE = 0
- NSE = 1
- KGE = 1
- KGE' = 1

Looking at the Fig. 4.13, it can be noticed that for the basins X0310010 (Alps in France) and 200105 (Alps in Austria) the results obtained for GR4J+CemaNeige are clearly better than those obtained with the other models. Criteria values relative to GR4J, GR5J and GR6J, indeed, seem to be far from the optimal values: this can be due to the lack of the snow precipitation and melting processes modelled.
Regarding the Breton basin (J5412110), instead, the criteria values do not differ too much among the models: this helps demonstrating that the use of CemaNeige can be avoided for this basin since it does not increase the quality of the results, which in fact is already high without the use of the snow model.
All these considerations are consistent with the analysis performed in the paragraph 4.3.

	Basin	Model	RMSE[Q]	NSE[Q]	KGE[Q]	KGE'[Q]
1	Obs_X0310010	RunModel_GR4J	1.83	−0.21	0.11	0.11
2	Obs_X0310010	RunModel_GR5J	1.81	−0.19	0.12	0.12
3	Obs_X0310010	RunModel_GR6J	1.83	−0.21	0.12	0.12
4	Obs_X0310010	RunModel_CemaNeigeGR4J	0.66	0.85	0.89	0.91
5	Obs_J5412110	RunModel_GR4J	0.53	0.88	0.85	0.88
6	Obs_J5412110	RunModel_GR5J	0.53	0.88	0.84	0.88
7	Obs_J5412110	RunModel_GR6J	0.52	0.88	0.85	0.88
8	Obs_J5412110	RunModel_CemaNeigeGR4J	0.52	0.88	0.85	0.88
9	Obs_200105	RunModel_GR4J	5.47	−0.06	0.23	0.23
10	Obs_200105	RunModel_GR5J	5.36	−0.02	0.24	0.24
11	Obs_200105	RunModel_GR6J	5.52	−0.09	0.20	0.21
12	Obs_200105	RunModel_CemaNeigeGR4J	2.56	0.77	0.83	0.84

FIGURE 4.13: Identification of the best model for each basin analyzing the mean (computed over all the calibration experiments) of the criteria values in validations

Once the CemaNeige model has been excluded as option for the analysis of the Breton basin (J5412110), the best model among GR4J, GR5J and GR6J had to be found (see paragraph 4.4.2).

4.4.2 Best Model over J5412110 (French basin in Brittany)

For each observed streamflow, a Q_{sim} has been obtained using each one of the three models: then for each Q_{obs} the model simulating the discharge with the minimum $\|Q_{sim} - Q_{obs}\|$ has been chosen as "the best one".
Then, the whole time-series of observed streamflows (with the related "best models") has been divided in 500 different subsets of streamflows: since the full time-series was sorted in chronological order, these subsets can be called "chronological classes". For each one of the latters, the most frequent best model has been identified and reported in the Fig.4.14 as a vertical grey segment.

In the plot of Fig. 4.15 instead, the discharges have been sorted with respect to the ascending order of the observed values, 500 classes have been determined and then, for each of them, the best model is marked with a different color. From the first plot in Fig. 4.14 it appears clear that the GR4J model is the one that results to be the best model less frequently. Between GR5J and GR6J the difference is not huge, even though the grey lines of the GR5J stripe are slightly less dense for the first part of the time series (around the period 1965 - 1985).
Regarding the second plot in Fig.4.15, it can be observed that the red color is the one that pops up less frequently, indicating that GR4J is the worst model among the three. Additionally, we see a series of just yellow segments corresponding to the lowest 2000 discharges, followed by a part in which the blue segments prevail. Then for the discharges higher than $2mm_/day$, in the right-hand side of the plot, the blue and yellow segments seem to alternate quite uniformly. Overall, if we have to choose just one model, the one with higher frequency of being the best one is GR6J. On the other hand, if we would have to perform a study on a specific range of streamflow, the best way to proceed would be to look at the corresponding portion of the plot in 4.15.

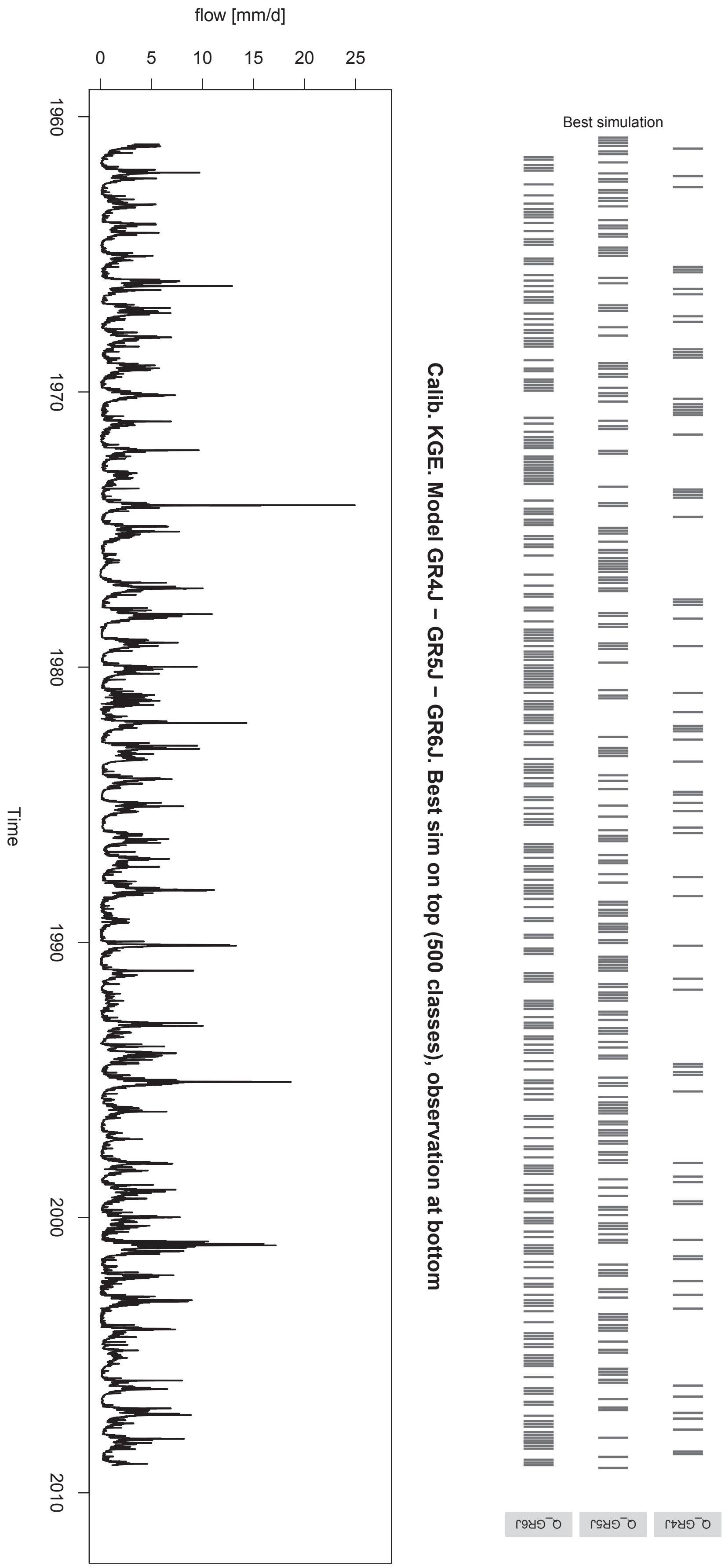

FIGURE 4.14: Identification of the best model for the basin J5412110 among GR4J, GR5J and GR6J (over each one of the 500 chronological classes of streamflows)

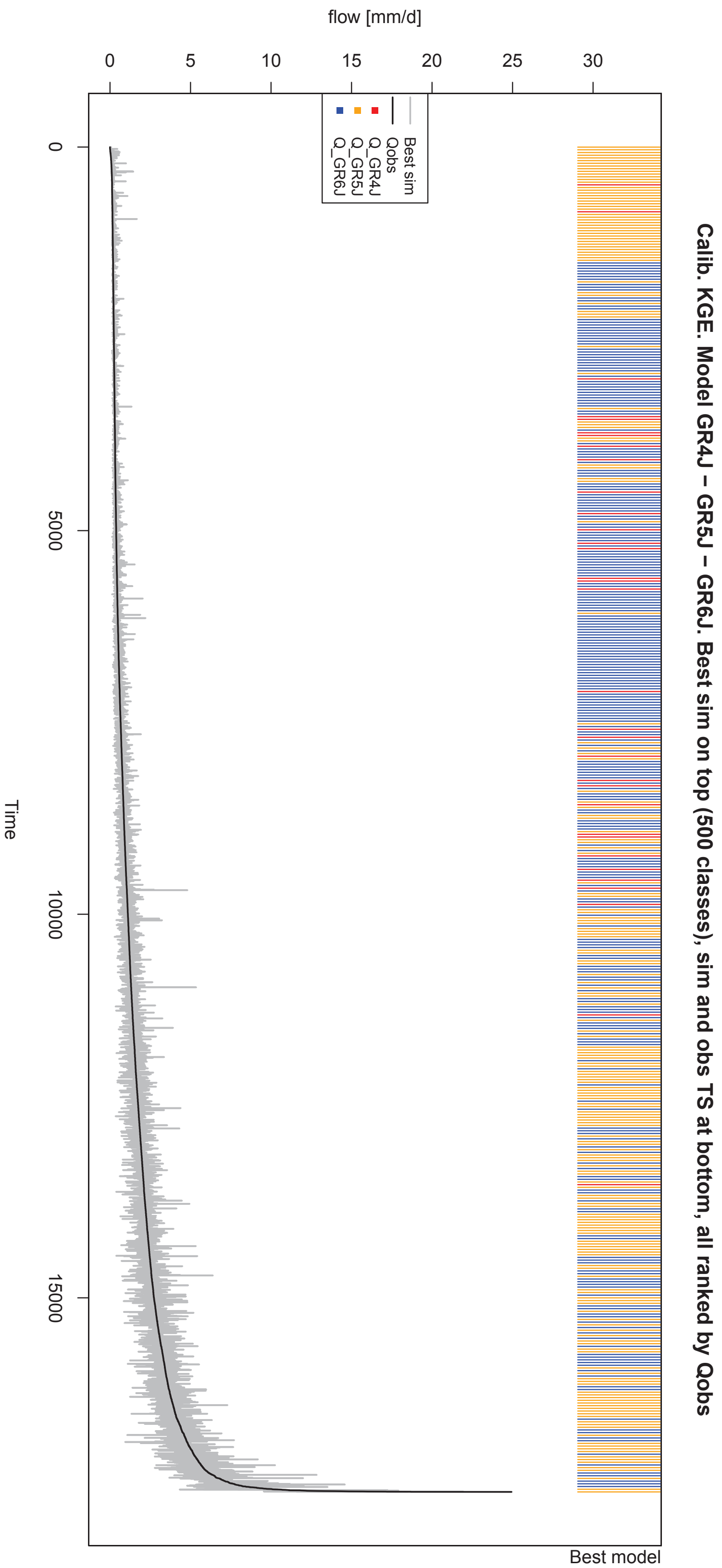

FIGURE 4.15: Identification of the best model for the basin J5412110 among GR4J, GR5J and GR6J (over each one of the 500 classes of sorted - observed - streamflows). The x-axes indicates the times that a certain discharge is not exceeded by the other ones of the sample

Chapter 5

Transformations versus extreme flows

As already introduced in chapter 1.2.5, in order to better analyse the performances of the "best" model over respectively high and low flows, transformations can be applied during the calibration process to re-scale the magnitude of the streamflow values. Some example of transformations used in hydrology are:

- Square root $\sqrt{Q}$
- Logarithm $\log(Q)$
- Inverse $1/Q$

Usually the invert and log functions are used to simulate better the low flows, whereas applying no transformation appears to be more efficient for the analysis of high flows. The square root function, instead, is well performing for both high and low flows.

In order to highlight the effects of the different transformations on the analysis of high and low flows, the model GR4J together with CemaNeige has been run over the French and Austrian nival basins (X0310010 and 200105) and calibrated over the calibration periods and objective functions already introduced. For each parametrization, a series of simulations of the extreme flows (see below for details) over different validation periods have been performed. The four goodness-of-fit criteria ($RMSE$, NSE, KGE and KGE') have been then calculated with the different four transformations Q, $\sqrt{Q}$, $1/Q$ and $\log(Q)$ for each simulation.
The "log" transformation coupled with KGE and KGE′ may lead to biased evaluation of model performance, and therefore has been avoided during the calculations [Santos et al., 2018].

Therefore the different combinations of criteria and transformations used to study the goodness of the simulations are:

- RMSE[Q], RMSE[$\sqrt{Q}$], RMSE[$\log(Q)$], RMSE[$1/Q$]
- NSE[Q], NSE[$\sqrt{Q}$], NSE[$\log(Q)$], NSE[$1/Q$]
- KGE[Q], KGE[$\sqrt{Q}$], KGE[$1/Q$]
- KGE'[Q], KGE'[$\sqrt{Q}$], KGE'[$1/Q$]

Regarding the "Low flows" I considered flow values smaller than the 20 percentile of the discharges, while I took values higher than the 80 percentile as high flows.
Since the optimal values of the criteria can be either 0 or 1, the direct comparison between criteria values would be meaningless. Hence, after each simulation I subtracted to each criteria value its optimum value, obtaining a series of "deviations": the bigger was the deviation, the worse was the simulation. Then the deviations got from the criteria values computed with the same transformation (e.g. RMSE[$\sqrt{Q}$], NSE[$\sqrt{Q}$], KGE[$\sqrt{Q}$] and KGE'[$\sqrt{Q}$]) have been grouped, and, for each group corresponding to a transformation, the mean of the related set of deviations has been computed (see Fig.5.1): the smaller is the deviation mean, the closer is, on average, the criteria to its optimal value.

Mean.Dev	Q	sqrt	log	inv
High Flows	0.245	0.432	0.699	0.97
Low Flows	1.770	1.327	1.684	1.25

FIGURE 5.1: Means of the deviations/differences of all criteria calculated with each transformation over high and low flows over the two nival basins (see Appendix C for the script used)

The results obtained over the two nival basins (Fig.5.1) are consistent with what has been written above: the mean of the deviations got for the "High Flows" grows from "no transformation" (best performing) to the "inverse transformation", and viceversa for the "Low Flows". Despite that, the result got for the "log transformation" does not perform well on the low flow, as it was instead expected: this could be due to the fact that the average, in this case, has been computed over a smaller number of values due to the incompatibility between KGE and KGE' with "log".

Moreover, the same kind of analysis that has been done to get the plot in Fig.4.15 has been performed for each basin to compare the square root and the inverse transformed simulated streamflows with the natural simulated discharges (see from Fig.5.2 to Fig.5.7): in this case the KGE has been used as objective function in calibration.
The following plots have been performed with an R-script provided by the Dr. Guillaume Thirel, who will be publish it soon with a scientific article.
In particular, in Fig.5.2, 5.4 and 5.6, the best transformation (for each one of the 500 classes defined over the observed streamflows) is shown through the different colors.
Differently, in each one of Fig. 5.3, 5.5 and 5.7 the discharge time-series have been divided in 200 subsets, namely classes of streamflows. For each one of them, the simulated and observed trasformed discharges has been compared for each transformation used: a transformation was chosen as the best one when the related transformed simulated streamflow resulted to be the closest one to the transformed observed Q. Then the three transformations have been ranked depending on their frequency of resulting the first (1), the second (2) or the third (3) best transformation among the three. In this case only 200 classes have been used because an higher level of detail of the chart would make the general trend of the transformation less clear.
The darker oranges indicate the higher frequencies of a transformation to appear with a certain rank (1st, 2nd or 3rd, depending on the row), while the lightest orange is related to the lowest number of occurrences.

5.1 Transformation over X0310010 (French basin on Alps)

From the chart in Fig.5.2 we can see immediately the clear difference between the portion on the left hand side of the chart, which is mainly blue (inverse transformation) and the right one in red (no transformation): the square root function (in yellow), as best transformation, results to be distributed within the whole range of streamflows. Hence, the behaviour of the three transformations, in this case, result to be consistent with the expectations.
The only peculiarity is that the yellow segments unexpectedly appear with a denser concentration for extremes high flows. The analysis of this behaviour would be interesting to be deepened, but despite that it will not be performed

in this circumstance: in fact, in order to study this particular trend of *sqrt* I would need a wider range of basins to be studied and that would deviate my study from its initial goal.

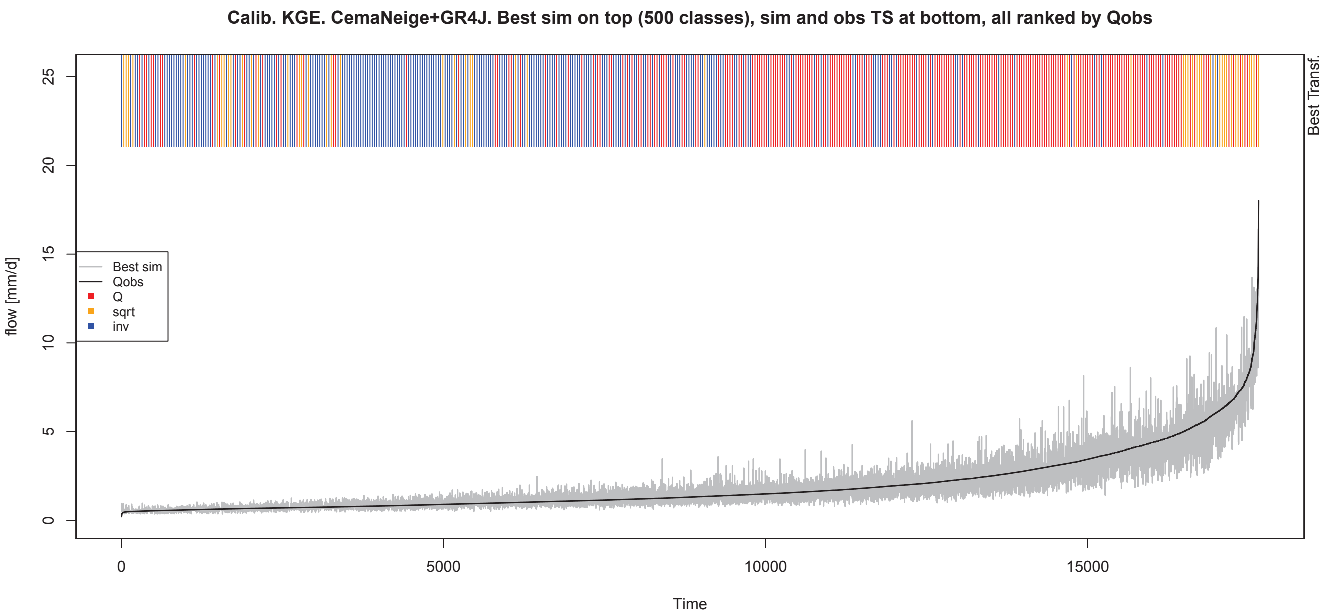

FIGURE 5.2: Identification of the best transformation for the basin X0310010 among Q (no transformation), *square root function* and *inverse function* (over each one of the 500 classes of sorted - observed - streamflows). The x-axes of the plot indicates the number of times that a certain discharge is not exceeded by the other ones of the sample.

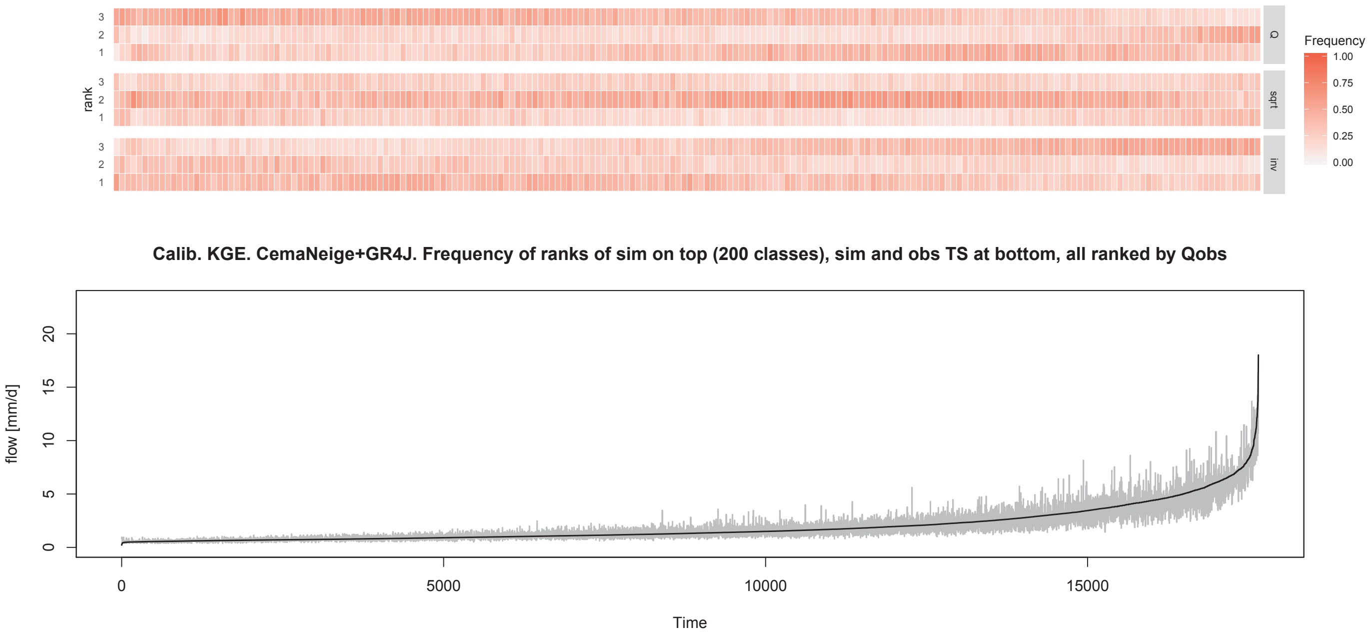

FIGURE 5.3: Rank of the transformations Q (no transformation), *square root function* and *inverse function* for the basin X0310010, over each one of the 200 classes of sorted - observed - streamflows. The x-axes of the plot indicates the number of times that a certain discharge is not exceeded by the other ones of the sample.

The trend of the transformations resulting in the chart of Fig. 5.3 confirms the one just observed in Fig.5.2: the best transformation (dark orange in line (1)) is *inv* for low flow, Q for high flow and *sqrt* for extreme high flows.
The *sqrt* portion of the chart is of the darkest orange for almost the whole line (2) and the lightest one for almost the line (3): this transformation is never the worst one, suggesting that *sqrt* is well performing for a wider range of streamflows, it seems a good compromise between high and low flows.
The shades of orange in lines (3) of Q and *inv* are opposite to the relative lines (1), showing that Q is not good to analyse low flows and vice versa for *inv*.

5.2 Transformation over J5412110 (French basin in Brittany)

An analysis similar to the one just done for the Alpine catchment, can be done for the Breton basin (Fig.5.4). The right hand side part of the chart is prevailed again by the color red, indicating the lack of transformation as the best way to approach the study of high flows. *inv* transformation results to be the best one mainly for the *extreme* low flows, while the yellow segments of the root square function appear homogeneously distributed over the whole range of stream flows: differently form the previous case, the *sqrt* seems to improve its performances for the low flows, where the *inv* used to rule for the Alpine basin (X0310010). The presence of *inv* transformation as the best one pop up just for extremes low flows: this could be due to the fact that the breton basin has a different regime with respect to the French Alpine one, without presence of snow. Again, in order to justify properly this fact one would need a larger set of catchments to be compared. However, one more time the general trend is consistent with the expectations.

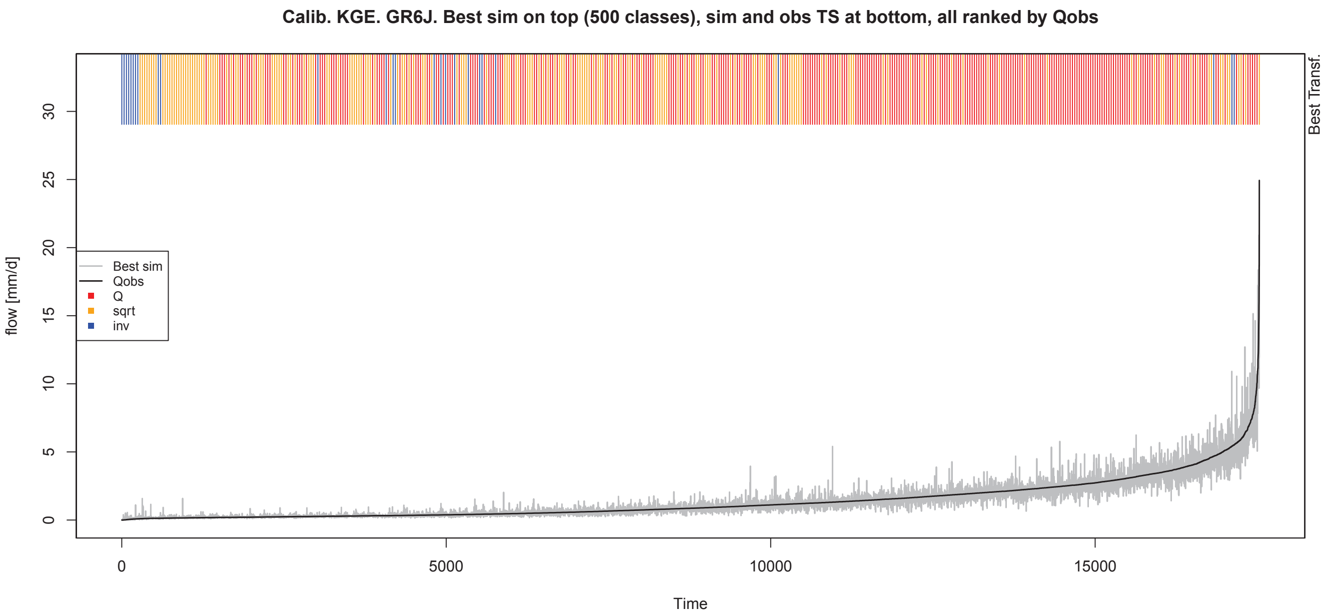

FIGURE 5.4: Identification of the best transformation for the basin J5412110 among Q (no transformation), *square root function* and *inverse function* (over each one of the 500 classes of sorted - observed - streamflows). The x-axes of the plot indicates the number of times that a certain discharge is not exceeded by the other ones of the sample.

The Fig. 5.5 confirms what just said, and helps highlighting the fact that the *sqrt* transformation is never the worst one, so it allows to perform analysis that concern a wide ranges of streamflows. The dark orange that immediately jumps out looking at the figure is the one on the line (3) of the *inv* transformation, pointing out the bad results got with it for almost all the discharges.

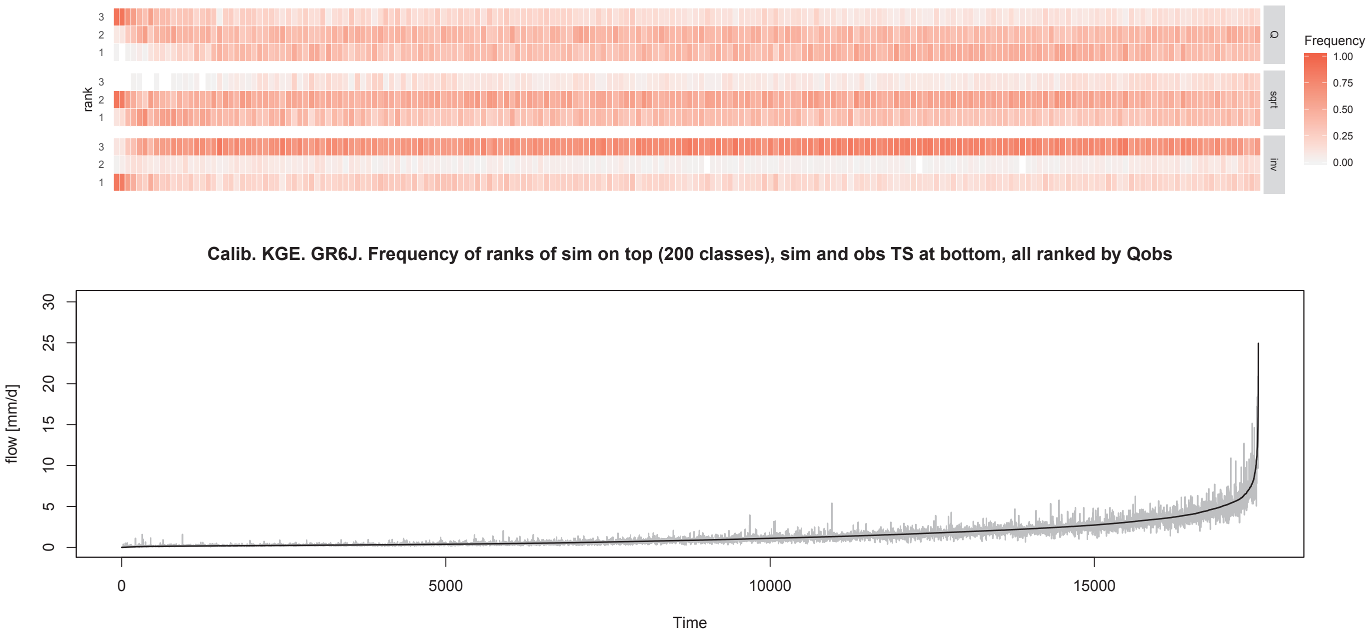

FIGURE 5.5: Rank of the transformations Q (no transformation), *square root function* and *inverse function* for the basin J5412110, over each one of the 200 classes of sorted - observed - streamflows. The x-axes of the plot indicates the number of times that a certain discharge is not exceeded by the other ones of the sample.

5.3 Transformation over 200105 (Austrian basin)

For the Austrian basin the comments that can be done are similar to those done for the previous two catchments.
However, in this case the pattern come out more clearly from the second chart of Fig.5.7 instead of Fig.5.6. From the latter it can be observed that the *inv* transformation works better on low flows, while the color red for Q and yellow for *sqrt* are mixed for intermediate and high flows. In Fig.5.7, it can be seen that for low flows, Q is the worst transformation (dark orange in line (3)), *inv* is the best one and *sqrt* is in the second place. Vice versa happens for the high flows. *sqrt* remains the second best model over almost all the stream flows.

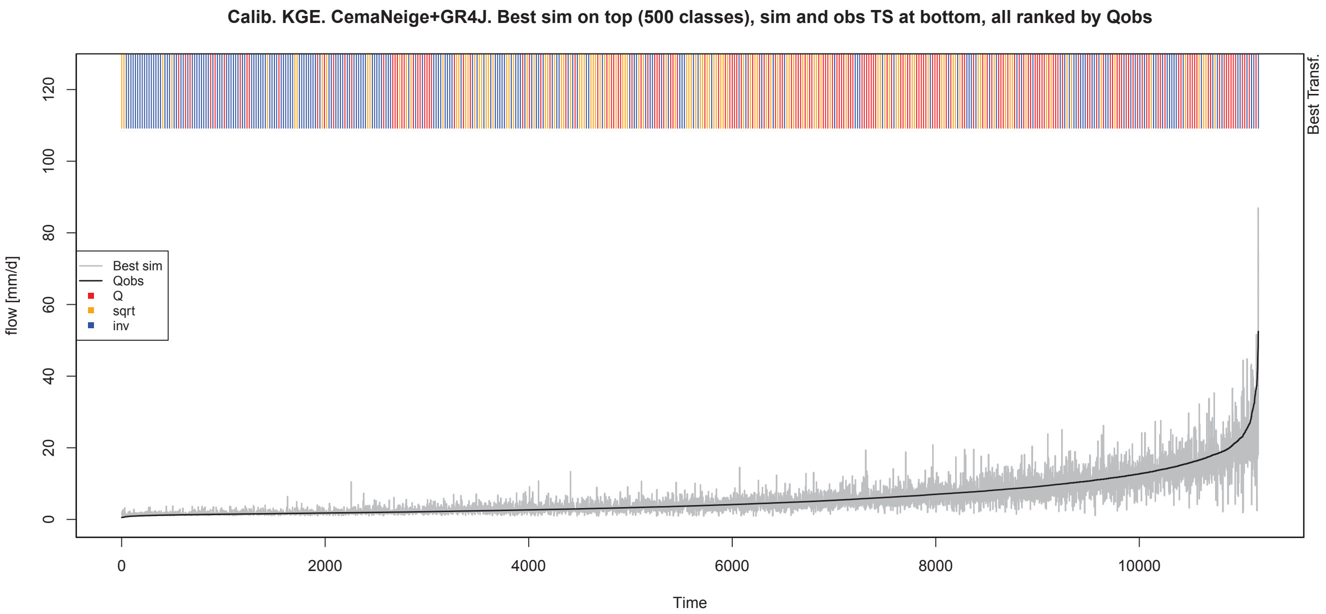

FIGURE 5.6: Identification of the best transformation for the basin 200105 among Q (no transformation), *square root function* and *inverse function* (over each one of the 500 classes of sorted - observed - streamflows). The x-axes of the plot indicates the number of times that a certain discharge is not exceeded by the other ones of the sample.

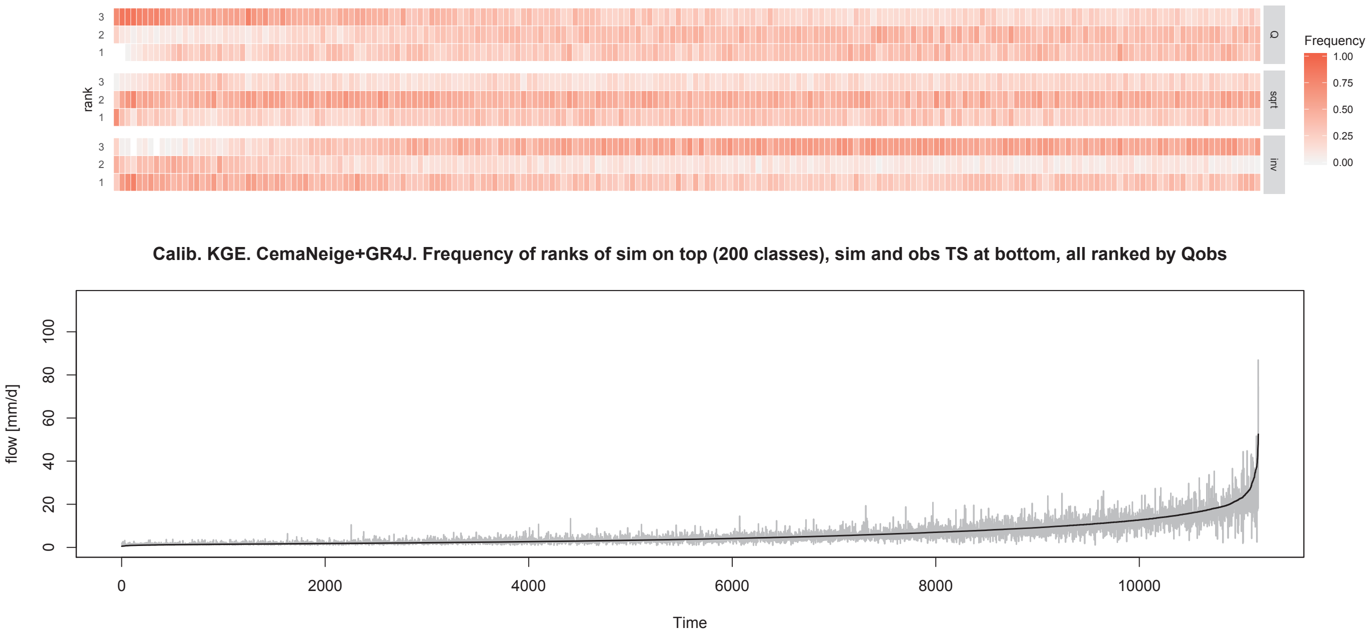

FIGURE 5.7: Rank of the transformations Q (no transformation), *square root function* and *inverse function* for the basin 200105, over each one of the 200 classes of sorted - observed - streamflows. The x-axes of the plot indicates the number of times that a certain discharge is not exceeded by the other ones of the sample.

Chapter 6

Dam module analysis

6.1 Introduction: artificial reservoirs and dams

It is well known that reservoirs, natural or not, can be used for different purposes, such as drinking water supply, irrigation, production of hydroelectric power, flood control, low-flow enhancement and recreation. If we talk about an artificial reservoir we do not have to forget that they are essentially designed to modify the natural hydrologic processes [Payan et al., 2008].
Some of the impacts that a barrage can have are:

- Fragmentation of river ecosystems
- Reservoir sedimentation
- Erosion of riverbed, river shores, coasts
- Evapotranspiration, infiltration and water transfer at the reservoir level
- Water temperature

Forecasting and quantifying all the effects of a man-made regulating structure downstream of a water course, result to be rather complex: in order to take into account the presence of a dam in a rainfall-runoff model, particular modules are needed. Due to the fact that a barrage is a punctual work in a river catchment, the most intuitive way to proceed would be using a distributed model and suitably modelling the sub-catchment corresponding to the dam in order to take into account its behaviour: running lumped models seems to be incompatible with managing the presence of a local reservoir because, as in every conceptual model, the model components do not represent directly physical processes, so it appears difficult to understand their correspondent role in the real world. However Payan et al. [2008] showed that it's possible to use lumped models if observed values of the variation of water volume stored in the reservoir are considered as additional input to the model: thus, the model structure remains unchanged and no new parameter or function has to be considered. This solution does not take into account the precise location of the dam within the catchment: nevertheless, it ends up improving the simulation of the downstream regulated discharge, and additionally it results to be a robust solution, namely applicable to different basins conditions [Payan et al., 2008].

6.2 Description of the module

The dam module is taken into account in the GR models considering an additional storage between the *Production store* and the *Routing store* (see Fig. 6.1). It has been originally designed for GR4J [Payan et al., 2008], but since the structure of the latter is similar to the one of GR5J then the presence of the barrage could be simulated also with this last model. Differently happens for GR6J, which has a structure different from GR4J and therefore its use together with the dam module has to be avoided.
The content of the additional store corresponds to the total stored volume observed in the reservoir and, as for the other stores, is expressed in *mm* (volume of the content divided by the watershed area, exactly as all the other volumetric measures in the model).

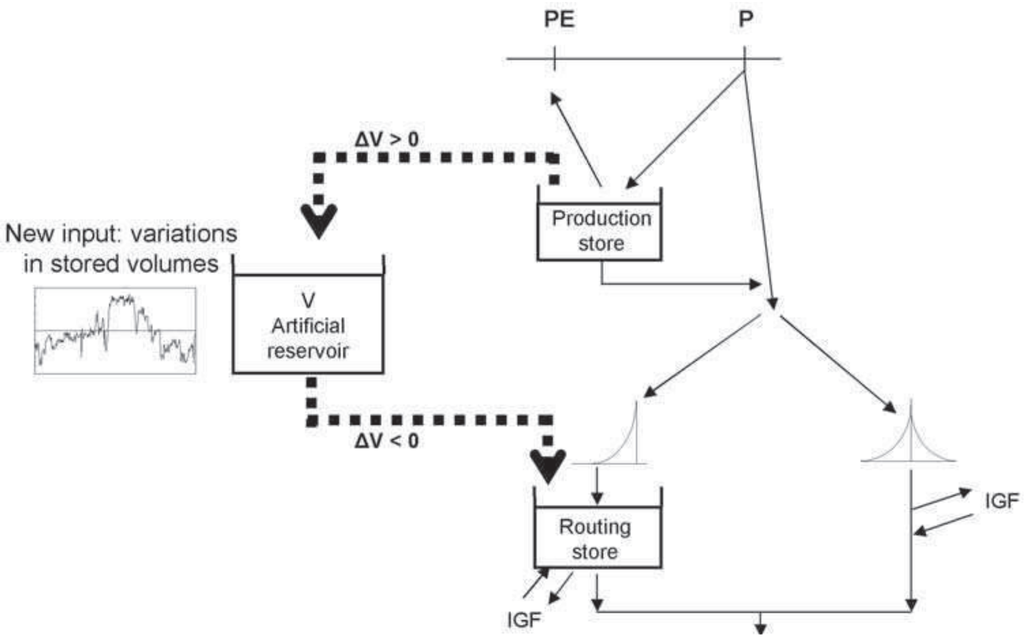

FIGURE 6.1: How to account for an artificial reservoir volume variation (V) - structure of a generic GR model combined with the "dam module" [Payan et al., 2008]

All the processes occurring within the reservoir (e.g. infiltration, evaporation, operations...) are not taken into account singularly in the model, but they are summed up dealing with the volume variation only. The behaviour of the store is totally described through ΔV observations, without any need of parametrization: this is due to the fact that the changes in the water height in the storage are mainly results of management actions and are not directly related to natural processes. Therefore no additional parameters has to be calibrated to take into account the presence of the dam: just the volume variations have to be taken into account as new input for the model.

The part that follows describes more in detail how the model works.
Assuming that the volume in our store is V_i at the *i*-time step and V_{i+1} at the *(i+1)*-time step, it results clear that $\Delta V = |V_{i+1} - V_i|$ is the quantity of water exchanged between the lumped representation of the artificial reservoir and the lumped representation of the basin (rainfall-runoff structure) [Payan et al., 2008]. It is not obvious a priori which part of the rainfall-runoff model is affected by this volume change: in [Payan, 2007] and [Payan et al., 2008] different model options are tested to finally show that, on average, the presence of the dam impacts mainly the *Routing and production stores*, in the way that is explained below.
If $\Delta V > 0$, then the water level in the reservoir has increased and it means that the water has flown from the natural system towards the reservoir: in this case the ΔV is subtracted from the *Routing store*. Vice versa, when the stored volume in the reservoir decreases, the $\Delta V < 0$ is added to the *Production store*. This justifies the location in Fig. 6.1 of the storage representing the artificial reservoir between the two stores just mentioned.

The following chapter will analyse how the the presence of a barrage could affect the streamflow in a watershed.
Due to the fact that among the three main basins available for my study there was not any catchment with an actual reservoir, the results got testing the module on one of them resulted to be pointless, and therefore the outcomes of this latter analyses will not be included in this report. Despite that, in order to highlight the role and the impact that a barrage can have on a watershed, I picked a fourth basin with a real barrage (Ill river basin - A1320310): I compared few years of observed streamflows at its closure section with the relative water levels measured in its reservoir.

6.3 Analysis of input data: observed discharges and volume data in a real case of a basin with a barrage (A1320310)

The following paragraph will not concern the use of the "dam module" but it will just focus on the study of its main input, namely the observed discharges at the closure section and the observed volume variation in the reservoir of the catchment FR_A1320310.
In particular, the comparison of these quantities will allow to underline the mitigation role that the reservoir has on high and low flows: as explained below, these two different effects emerge from the analysis performed on data available at two different time-scales.

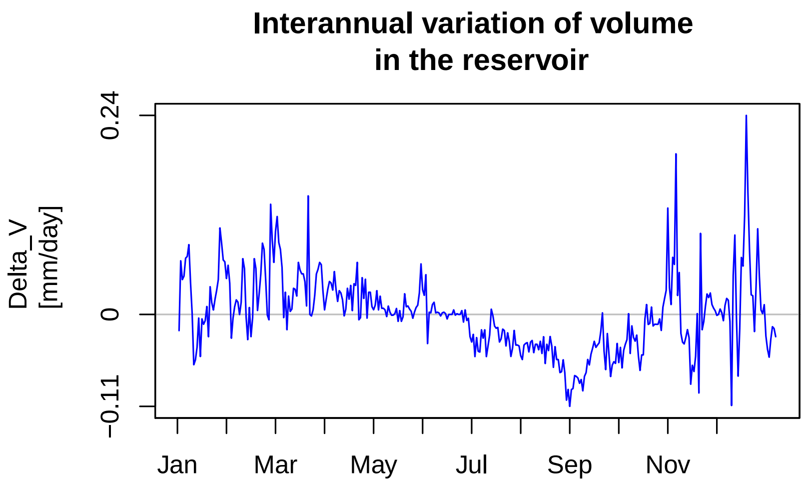

FIGURE 6.2: Interannual variation of volume in the reservoir of basin FR_A1320310

First of all, the periods during which the reservoir collects or releases water can be easily distinguished looking at Fig.6.2. From March to June the ΔV results to be mainly positive, indicating that the water is, on average, withdrawn for the natural environment and gathered in the reservoir; vice versa happens in the period between the beginning of July and the end of October, in which the water balance of the reservoir is outgoing, hence with the water flowing toward the natural system. For the months November-February instead, the trend is not clearly definable, with the interannual ΔV oscillating between values above and below zero.

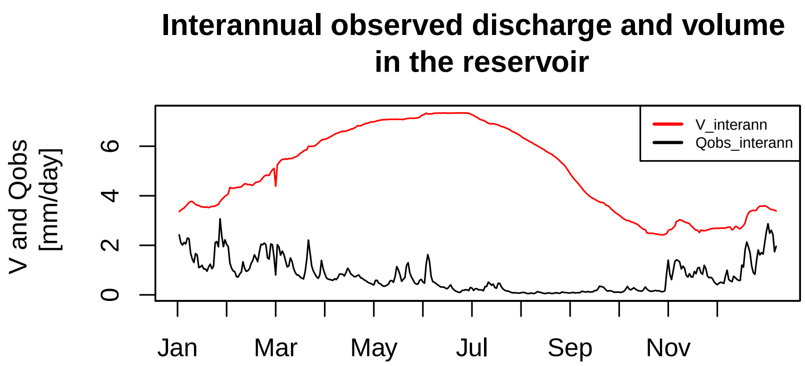

FIGURE 6.3: Interannual observed volume in the reservoir (graph in red) and streamflow at the closure section (graph in black on the bottom) of basin FR_A1320310

Obviously the pattern of Fig.6.2 reflects directly on the volume values plotted in Fig.6.3, with V decreasing in case of negative ΔV and and increasing otherwise.

Comparing the streamflow and the volume regime in Fig.6.3, some considerations can be made: the volume results to increase (reservoir collects water) when the Q decreases during the winter period, suggesting that the inflow to the reservoir is diverted to create a storage. Moreover, V decreases while very low and almost constant discharges occur during the summer-autumn months. The speed with which the volume increases is a bit lower than the one with which it decreases.

This pattern makes clear the mitigation target of the dam, which indeed allows to collect water in case of higher flows and to release it in case of lower flows.

However, the faster dropping of the reservoir volume in correspondence of an almost constant out-going discharge indicates a period in which the in-going flow of the reservoir decreases drastically.

Thus, with the analysis of the seasonal regime of volumes and discharges, the droughts counteracting role of the reservoir emerges more clearly: between June and October it allows to have at least a low flow, mitigating therefore the dry season.

On the other hand, analysing the full daily time-series of volumes and discharges instead of their regimes, will allow to highlight other aspects of the storage behaviour: the higher variability of data present in this type of study

allows the peaks of high flows to appear distinctly and therefore the delineation of the effects of the reservoir on high flows results to be straightforward.
Thus, the plots in Fig.6.4 and 6.5 will target to show the impact of the reservoir on floods: in particular Fig.6.5 shows the relation between the variation of observed streamflows and the change in the reservoir volume during specific events.
In particular, by way of example, an event is circled in red in Fig.6.4 and reported zoomed in Fig.6.5: in the latter plot, at the beginning of November, the variation of volume in the reservoir increases sharply in correspondence of the rapid increase of *Qobs*. This suggests that the probable flood inflow coming to the reservoir is still released as high flow but it is also mitigated by the presence of the dam, which, in fact allows part of the water to be stored, and then released just after the *Qobs* peak, decreasing the hypothetical dangerous character of the discharge. Going on with the research and producing new zoomed charts, several similar events can be located over the full time series.

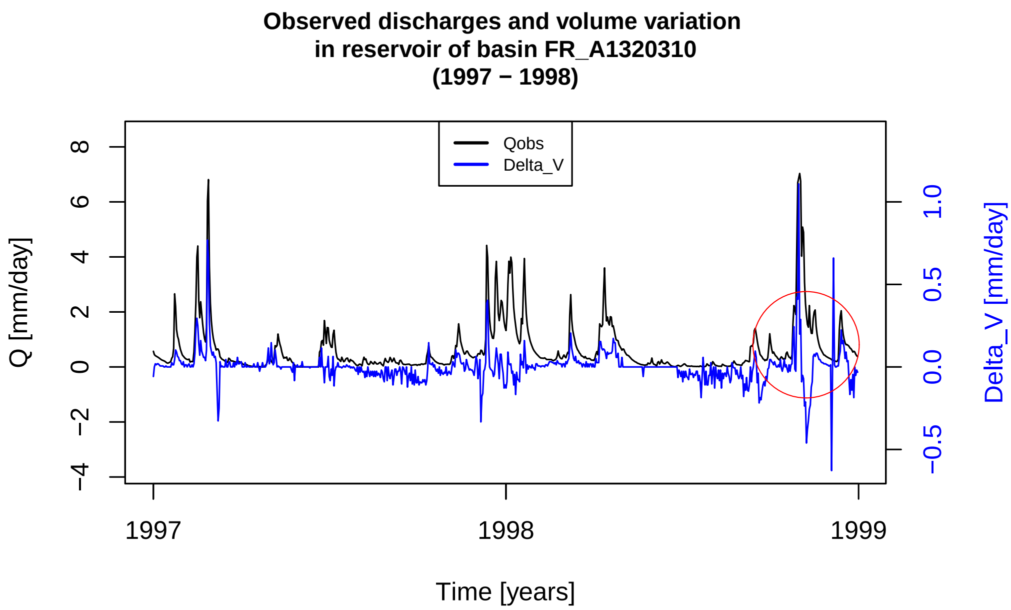

FIGURE 6.4: 2-years zoom of observed discharges affected by the dam and variation of volumes in the reservoir present in basin FR_A1320310

Overall, the mitigating behaviour of the reservoir on low and high flows appears clear. As expected, the smoothed effects of low flows come out more clearly observing the long and seasonal time-series, while the high-flow ones can be easily observed focusing more on the single events.

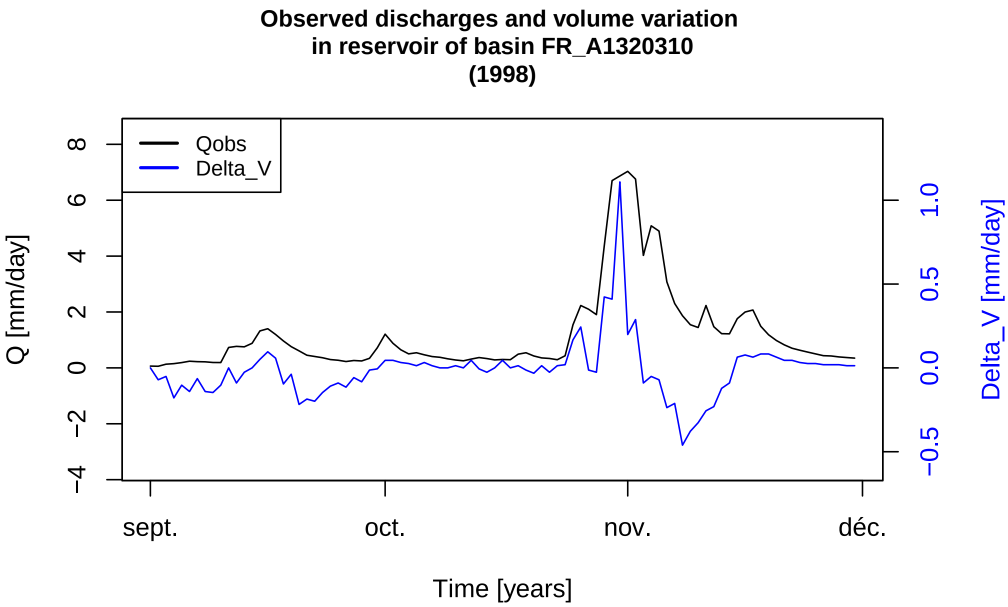

FIGURE 6.5: 3-months zoom of observed discharges affected by the dam and variation of volumes in the reservoir present in basin FR_A1320310

Chapter 7

Conclusions

The analysis, subject of this report, targets my attention on different aspects of hydrological modelling, which, overall, result to be connected all together.

As first step, before running the model, examining the hydro-climatic regime of the watershed is a fundamental step, together with the inspection of nival characteristic of the basin: indeed, it needs to be done in order to decide whether the snow accounting model CemaNeige has to be applied or not.

How to compare the different basins, models, objective functions and calibration periods, has been the core of the study: in particular, the way all these aspects influence the model parameters, and therefore the simulated hydrograph, was an important part of the study that helped me to understand better the model behaviour.
A secondary analysis has been done on the different components of the KGE goodness-of-fit criteria: indeed, they can be studied in validation to obtain information on different aspects of the simulated hydrographs.

Moreover, this report studies the importance of the snow for a river catchment, and therefore the necessity of combining an hydrological model with a snow accounting model as CemaNeige: I performed some tests in order to understand how the lack of the snow module, and of the simulated snow melt, would impact the simulated discharges both on nival and pluvial basins. Particular attention has been put also to find out how the different calibration periods and objective functions affect the outputs of the just mentioned snow accounting model. Overall, CemaNeige resulted to be essential for the nival basins while it does not improve the simulation for the pluvial ones.

Furthermore, the improvement that using a transformation of the streamflow values can bring to the analysis of extremes flows have been tackled:

the natural *Q*, the *sqrt* and the *inv* transformations have been tested over the three basins; the trend that emerges in general is the good performance of *Q* for high flows, of *inv(Q)* for low flows and of *sqrt(Q)* for a wider range of discharges. However, there are slight differences from a basin to another, and this could be due to the different climates of the areas.

The last topic analysed was the effect that the presence of a dam can have on the catchment behaviour: a *dam module* has been developed by the IRSTEA research center to simulated the presence of a barrage in a watershed. Unfortunately, among the cases study available for my study, there were not catchments including reservoirs so I did not have the chance to test the module; although, in order to understand how the mitigating role of a barrage can be studied, few years of observed streamflows and water levels in the reservoir of a fourth basin have been compared in order to highlight the influence that an actual dam can have.

All these analyses allowed me to gain confidence with different aspects of hydrological modelling (additionally to improve my coding skills). The sets of the models tested and of the basins studied were limited, but they gave me the chance to get acquainted with a branch of water engineering that was unknown to me: thanks to this thesis work I will be able to face similar topics and issues with more flexibility and an higher level of critical thinking.

Appendix A

Script used for the calibration experiments

```r
rm(list=ls())
options(digits = 3)

library(airGR)
library(airGRteaching)

#setwd("C:/Data/chiara/03_CODES")
setwd("~/Desktop/chiara/03_CODES")

## DATA.FRAME OF OBSERVED DATA
# X0310010_BV
Input_X0310010 <- read.table(file =
"~/Desktop/chiara/02_DATA/French_Data/X0310010_BV.txt",
                             header = TRUE, sep = ";", skip = 50)
Input_X0310010$Q[Input_X0310010$Q < 0] <- NA;
Input_X0310010$Ptot[Input_X0310010$Ptot < 0] <- NA
A_X0310010 <- 2282.76
Input_X0310010$Qmm <- 0.0864*(Input_X0310010$Q)/A_X0310010
Input_X0310010$Time <- as.POSIXct(as.character(Input_X0310010$Date),
format = "%Y%m%d", tz = "UTC")
Obs_X0310010 <- Input_X0310010[, c("Time", "Ptot", "ETP_O", "Qmm",
"Temp")]

# J5412110_BV
Input_J5412110 <- read.table(file =
"~/Desktop/chiara/02_DATA/French_Data/J5412110_BV.txt",
                             header = TRUE, sep = ";", skip = 50)
Input_J5412110$Q[Input_J5412110$Q < 0] <- NA;
Input_J5412110$Ptot[Input_J5412110$Ptot < 0] <- NA
A_J5412110 <- 675.64
Input_J5412110$Qmm <- 0.0864*(Input_J5412110$Q)/A_J5412110
Input_J5412110$Time <- as.POSIXct(as.character(Input_J5412110$Date),
format = "%Y%m%d", tz = "UTC")
Obs_J5412110 <- Input_J5412110[, c("Time", "Ptot", "ETP_O", "Qmm",
"Temp")]
# 200105
Input_200105_1 <- read.table(file =
"~/Desktop/chiara/02_DATA/Austrian_Data/BaciniTest/obs_200105.txt",
                             header = FALSE, sep = "\t")
Input_200105_2 <- read.table(file =
"~/Desktop/chiara/02_DATA/Austrian_Data/BaciniTest/input_200105.txt",
                             header = TRUE, sep = "\t")
#Input_200105_2 <- Input_200105_2[[1]] %in% seq(from = as.POSIXct("1976-
01-01"), to = as.POSIXct("2008-12-31"), by = 1)
Input_200105_2 <- Input_200105_2[1:12054,]
Input_200105_3 <- read.table(file =
"~/Desktop/chiara/02_DATA/Austrian_Data/BaciniTest/AreaDist_200105.txt",
                             header = TRUE, sep = " ")

Input_200105_1$V2[Input_200105_1$V2 < 0] <- NA
Input_200105_2[Input_200105_2 < -99] <- NA #Warning
A_200105 <- 95.50   #Input_200105$Qmm <- 0.0864*(Input_200105$Q)/A_200105
Input_200105_1$Time <- as.POSIXct(as.character(Input_200105_1[,1]),
format = "%Y-%m-%d", tz = "UTC")
T <- apply(Input_200105_2[,21:39], 1, function(x, w) weighted.mean(x,
Input_200105_3[,1]))
P <- apply(Input_200105_2[,2:20], 1, function(x, w) weighted.mean(x,
Input_200105_3[,1]))
EP <- apply(Input_200105_2[,40:58], 1, function(x, w) weighted.mean(x,
Input_200105_3[,1]))
# function(x, w) weighted.mean(x, Input_200105_3[,1])
# function(x) sum(x, na.rm = TRUE)

Obs_200105 <- data.frame(col1 = Input_200105_1$Time,
                         col2 = P,
                         col3 = EP,
                         col4 = Input_200105_1$V2,
                         col5 = T)
colnames(Obs_200105) <-  c("Time", "Ptot", "ETP_O", "Qmm", "Temp")


# Input CemaNeige
Input_Neige_Fr <- read.table(file =
"~/Desktop/chiara/02_DATA/French_Data/_ListeBV_Quantiles_altitude.txt",
                             header = TRUE, sep = "")
Input_Hypso_Au <- read.table(file =
"~/Desktop/chiara/02_DATA/Austrian_Data/HypsoZ.txt",
                             header = TRUE, sep = "\t")
Input_Zin_Au <- read.table(file =
"~/Desktop/chiara/02_DATA/Austrian_Data/Zin.txt",
                           header = TRUE, sep = "\t")
Input_CN <- data.frame(matrix(nrow = 102, ncol = 3))
rownames(Input_CN) <- c(colnames(Input_Neige_Fr[,5:106]))
colnames(Input_CN) <- c("X0310010", "J5412110", "200105")
Input_CN["Zmean", ] <- c(Input_Neige_Fr$Zmean, Input_Zin_Au[1,1])
Input_CN[2:102, "X0310010"] <- t(Input_Neige_Fr[1, 6:106])
Input_CN[2:102, "J5412110"] <- t(Input_Neige_Fr[2, 6:106])
Input_CN[2:102, "200105"] <- t(Input_Hypso_Au["200105", 1:101])

# Vectors for the loop
BASIN <- list(Obs_X0310010, Obs_J5412110, Obs_200105)
BASIN_char <- c("Obs_X0310010", "Obs_J5412110", "Obs_200105")
MOD <- c(RunModel_GR4J, RunModel_GR5J, RunModel_GR6J,
RunModel_CemaNeigeGR4J)
MOD_char <- c("RunModel_GR4J", "RunModel_GR5J", "RunModel_GR6J",
"RunModel_CemaNeigeGR4J")
CRIT <- c(ErrorCrit_RMSE, ErrorCrit_NSE, ErrorCrit_KGE, ErrorCrit_KGE2)
CRIT_char <- c("RMSE[Q]", "NSE[Q]", "KGE[Q]", "KGE'[Q]")

Ind_Run_AU <- list((seq(which(format(Obs_200105$Time, format = "%Y-%m-
%d", "UTC") == "1978-01-01"),
                        which(format(Obs_200105$Time, format = "%Y-%m-
%d", "UTC") == "1993-12-31"))),
                   (seq(which(format(Obs_200105$Time, format = "%Y-%m-
%d", "UTC") == "1994-01-01"),
                        which(format(Obs_200105$Time, format = "%Y-%m-
%d", "UTC") == "2008-12-31"))))
Ind_Run_FR <- list((seq(which(format(Obs_X0310010$Time, format = "%Y-%m-
%d", "UTC") == "1961-01-01"),
                        which(format(Obs_X0310010$Time, format = "%Y-%m-
%d", "UTC") == "1985-12-31"))),
                   (seq(which(format(Obs_X0310010$Time, format = "%Y-%m-
%d", "UTC") == "1986-01-01"),
                        which(format(Obs_X0310010$Time, format = "%Y-%m-
%d", "UTC") == "2010-07-31"))))
```

```
Per_Cal_FR <- c("1961-1985", "1986-2010")
Per_Cal_AU <- c("1978-1993", "1994-2008")
Tran <- c("", "sqrt", "log", "inv", "sort")
#Forc_m <- c(mean(Obs_X0310010[seq(from =  as.Date("1961-01-01"), to =
as.Date("1985-12-31"), by = "1 day"),
#                  "Ptot"]),
#            mean(Obs_X0310010[seq(from =  as.Date("1961-01-01"), to =
as.Date("1985-12-31"), by = "1 day"),
#                  "Temp"]))


# Dataframe results
results_df <-  as.data.frame(matrix(ncol = 18))
colnames(results_df) <- c("Basin", "Model", "Per_Cal", "Per_Val",
"OF_Name", "OF_Value",
                          "X1", "X2", "X3", "X4", "X5", "X6", "C1", "C2",
                          "RMSE", "NSE", "KGE", "KGE2")
Crit_values <- t(vector(length = 4))

SNOW <- data.frame(matrix(ncol = 1, nrow = 12))
SNOW[,1] <- c("01", "02","03", "04", "05", "06", "07", "08", "09", "10",
"11", "12")
colnames(SNOW) <- "Month"
La <- as.data.frame(matrix(ncol = 6))

coeff_df <-  as.data.frame(matrix(ncol = 9))
colnames(coeff_df) <- c("Per_Cal", "OF_Name", "Per_Val",
                        "KGE_Value", "KGE2_Value", "r", "alpha", "beta",
"gamma")

# #pdf("Results_X0310010.pdf", width = , height = , pointsize = 12)
# pdf("CemaNeige_Test_GR6J-CN.pdf", pointsize = 10)
# par(oma = c(0, 0, 7, 0))
# xx <- c(3, 4)

#results_df <- read.csv(file =
"~/Desktop/chiara/03_CODES/1.2.3_basins_analysis/Results_3basins_modified")


## LOOPS
for (b in seq_along(BASIN)[1]){
  for (x in seq_along(MOD)[4]){
    for (p in seq_along(Per_Cal_FR)){
      for (i in seq_along(CRIT)[3:4]) {
        ff = 1 #for (ff in seq_along(Tran)) {

        # define calibration period
        if (b == 3) {
          Ind_Run_Cal <- Ind_Run_AU[[p]]
          Per_Cal <- Per_Cal_AU[[p]]
          WuP <- 1:730
        } else  {
          Ind_Run_Cal <- Ind_Run_FR[[p]]
          Per_Cal <- Per_Cal_FR[[p]]
          WuP <- 1:(as.numeric(difftime(strptime("1960-12-31", format =
"%Y-%m-%d"),
                                        strptime(Obs_X0310010[1, "Time"],
format = "%Y-%m-%d"), units = "days")))
        }

        # define inputs model and run options
        InputsModel <- CreateInputsModel(FUN_MOD = MOD[[x]], DatesR =
BASIN[[b]]$Time,
                                         Precip = BASIN[[b]]$Ptot,
PotEvap = BASIN[[b]]$ETP_O,
                                         TempMean = BASIN[[b]]$Temp,
ZInputs = Input_CN["Zmean", b],
                                         HypsoData = Input_CN[2:102, b],
NLayers = 5)

        RunOptions_Cal <- CreateRunOptions(FUN_MOD = MOD[[x]],
InputsModel = InputsModel,
                                           IndPeriod_Run = Ind_Run_Cal,
IndPeriod_WarmUp = WuP) #warning solid precipitation

        ## CALIBRATION
        InputsCrit <- CreateInputsCrit(FUN_CRIT = CRIT[[i]], InputsModel
= InputsModel,
                                       RunOptions = RunOptions_Cal, Qobs
= BASIN[[b]]$Qmm[Ind_Run_Cal],  transfo = Tran[[ff]])
        CalibOptions <- CreateCalibOptions(FUN_MOD = MOD[[x]], FUN_CALIB
= Calibration_Michel)
        OutputsCalib <- Calibration_Michel(InputsModel = InputsModel,
RunOptions = RunOptions_Cal,
                                           InputsCrit = InputsCrit,
CalibOptions = CalibOptions,
                                           FUN_MOD = MOD[[x]], FUN_CRIT =
CRIT[[i]])
        # Store param from OutputsCalib$
        if (length(OutputsCalib$`ParamFinalR`) == 4) {
          PAR <- c(OutputsCalib$`ParamFinalR`[[1]],
OutputsCalib$`ParamFinalR`[[2]], OutputsCalib$`ParamFinalR`[[3]],
                   OutputsCalib$`ParamFinalR`[[4]], NA, NA, NA, NA)
        } else {
          if (length(OutputsCalib$`ParamFinalR`) == 5) {
            PAR <- c(OutputsCalib$`ParamFinalR`[[1]],
OutputsCalib$`ParamFinalR`[[2]], OutputsCalib$`ParamFinalR`[[3]],
                     OutputsCalib$`ParamFinalR`[[4]],
OutputsCalib$`ParamFinalR`[[5]], NA, NA, NA)
          } else {
            if (x == 4) {   #GR4J+CemaNeige
              PAR <- c(OutputsCalib$`ParamFinalR`[[1]],
OutputsCalib$`ParamFinalR`[[2]], OutputsCalib$`ParamFinalR`[[3]],
                       OutputsCalib$`ParamFinalR`[[4]], NA, NA,
OutputsCalib$`ParamFinalR`[[5]], OutputsCalib$`ParamFinalR`[[6]])
            } else {        #GR6J
              PAR <- c(OutputsCalib$`ParamFinalR`[[1]],
OutputsCalib$`ParamFinalR`[[2]], OutputsCalib$`ParamFinalR`[[3]],
                       OutputsCalib$`ParamFinalR`[[4]],
OutputsCalib$`ParamFinalR`[[5]], OutputsCalib$`ParamFinalR`[[6]], NA, NA)
            }}
        }



        ## SIMULATION
```

```
        for (pp in seq_along(Per_Cal_FR))  {
          for (ii in seq_along(CRIT)[3:4]) {

            if (b == 3) {
              Ind_Run_Val <- Ind_Run_AU[[pp]]
              Per_Val <- Per_Cal_AU[[pp]]
            } else  {
              Ind_Run_Val <- Ind_Run_FR[[pp]]
              Per_Val <- Per_Cal_FR[[pp]]
            }

            RunOptions_Val <- CreateRunOptions(FUN_MOD = MOD[[x]],
InputsModel = InputsModel,
                                               IndPeriod_WarmUp = WuP,
IndPeriod_Run = Ind_Run_Val)
            Param <- OutputsCalib$`ParamFinalR`
            OutputsModel <- MOD[[x]](InputsModel = InputsModel,
                                     RunOptions = RunOptions_Val, Param =
Param)
            InputsCrit <- CreateInputsCrit(FUN_CRIT = CRIT[[ii]],
InputsModel = InputsModel,
                                           RunOptions = RunOptions_Val,
Qobs = BASIN[[b]]$Qmm[Ind_Run_Val])
            OutputsCrit <- CRIT[[ii]](InputsCrit = InputsCrit,
OutputsModel = OutputsModel)

            Crit_values[1, ii] <- OutputsCrit$CritValue

            #Plot snow from CemaNeige
            if (b == 1 & x == 4){
              La <- data.frame(col1 = as.Date(OutputsModel$DatesR),
                               col2 =
OutputsModel$CemaNeigeLayers$Layer01$Melt,
                               col3 =
OutputsModel$CemaNeigeLayers$Layer02$Melt,
                               col4 =
OutputsModel$CemaNeigeLayers$Layer03$Melt,
                               col5 =
OutputsModel$CemaNeigeLayers$Layer04$Melt,
                               col6 =
OutputsModel$CemaNeigeLayers$Layer05$Melt)
              colnames(La) <-  c("Time", "Layer01", "Layer02", "Layer03",
"Layer04", "Layer05")
              La$sum <- (1/5)*(apply(La[,2:6], MARGIN = 1, FUN = sum))
            } else {
              if (b == 3 & x == 4){
                La <- data.frame(col1 = as.Date(OutputsModel$DatesR),
                                 col2 =
OutputsModel$CemaNeigeLayers$Layer01$Melt,
                                 col3 =
OutputsModel$CemaNeigeLayers$Layer02$Melt,
                                 col4 =
OutputsModel$CemaNeigeLayers$Layer03$Melt,
                                 col5 =
OutputsModel$CemaNeigeLayers$Layer04$Melt,
                                 col6 =
OutputsModel$CemaNeigeLayers$Layer05$Melt)
                colnames(La) <-  c("Time", "Layer01", "Layer02",
"Layer03", "Layer04", "Layer05")
                La$sum <- (1/5)*(apply(La[,2:6], MARGIN = 1, FUN = sum))
              }
            }

            #Aggregate data over each month (daily to monthly data)
            La$my <- format(as.Date(La$Time), "%Y-%m")
            len2 <- length(aggregate(La$sum, by = list(La$my),
                                     function (x) sum(x, na.rm =
TRUE))$Group.1)
            SnowM <- as.data.frame(matrix(nrow = len2, ncol = 3))
            colnames(SnowM) <-  c("my", "month","Agg_S")
            SnowM$Agg_S <- aggregate(La$sum, by = list(La$my),
                                     function (x) sum(x, na.rm = TRUE))$x
            SnowM$my <- aggregate(La$sum, by = list(La$my),
                                  function (x) sum(x, na.rm =
TRUE))$Group.1
            SnowM$month <- substring(SnowM$my, 6, 7)

            # Mean over the years per each month
            Agg_S <- (aggregate(SnowM$Agg_S,
                                by = list(SnowM$month),
                                function(x) mean(x, na.rm = TRUE)))$x
            SNOW <- cbind(SNOW, Agg_S)


            #Plot
            plot(OutputsModel, Qobs = BASIN[[b]]$Qmm[Ind_Run_Val])
            mtext(text = paste(BASIN_char[[b]], "   ", MOD_char[[x]]),
line = 4, outer = TRUE)
            mtext2 <- sprintf("Calib. Period = %s - Valid. Period = %s -
O. F. = %s - Criteria = %s",
                              Per_Cal, Per_Val,
                              OutputsCalib$CritName,
OutputsCrit$CritName)
            mtext(text = mtext2, line = 3, outer = TRUE)

            #Store crit value from OutputsCrit$
            result_row <- c(BASIN_char[[b]], MOD_char[[x]], Per_Cal,
Per_Val, OutputsCalib$CritName,
                            as.numeric(OutputsCalib$CritFinal), PAR,
Crit_values)
            results_df <- rbind(results_df, result_row)


            #Plot of the zooms
            #Method 1: not working
            Obs_X0310010 <- Obs_X0310010[Obs_X0310010$Time >=
as.POSIXlt("1961-01-01", format = "%Y-%m-%d", tz = "UTC"),]
            Period_Plot <- seq(which(format(BASIN[[b]]$Time, format =
"%Y-%m-%d", "UTC") == "1978-01-01"),
                               which(format(BASIN[[b]]$Time, format =
"%Y-%m-%d", "UTC") == "1979-12-31"))
            plot(OutputsModel, Qobs = BASIN[[b]]$Qmm[Period_Plot],
IndPeriod_Plot = Period_Plot, which = "Flows")
            plot(OutputsModel$Qsim[Period_Plot])
            lines(BASIN[[b]]$Qmm[Period_Plot])
            length(OutputsModel$Qsim[Period_Plot])
```

```
            length(BASIN[[b]]$Qmm[Period_Plot])

            #Method 2: working
            dates <- seq(from = as.POSIXlt("1961-01-01", format = "%Y-%m-
%d", "UTC"),
                       to = as.POSIXlt("1962-12-31", format = "%Y-%m-%d",
"UTC"), by = "1 day")
            plot(dates, y = BASIN[[b]][(BASIN[[b]]$Time %in% dates),
colnames(BASIN[[b]]) %in% "Qmm"],
               type = "l", col = "black", xlab = "Time [years]", ylab =
"Flow [mm/d]", lwd = 1.8, xaxt = "n")

            lines(OutputsModel$DatesR[1:730], OutputsModel$Qsim[1:730],
col = "orangered", lwd = 1.8)
            legend ("topright", legend = c("Observed", "Simulated"),
                  col = c("black", "orangered"), cex = 0.6,
                  pch = 15, bty = "n")
            axis(side = 1, labels = dates[format(as.Date(dates), "%m-%d")
%in% "01-01"], at = dates[format(as.Date(dates), "%m-%d") %in% "01-01"])

            if (ii == 3 & b == 1 & x == 4){    #KGE
              coeff_row <- c(Per_Cal, OutputsCalib$CritName, Per_Val,
                             OutputsCrit$CritValue, NA,
                             OutputsCrit$SubCritValues[[1]],
OutputsCrit$SubCritValues[[2]], OutputsCrit$SubCritValues[[3]], NA)
            } else {
              if (ii == 4 & b == 1 & x == 4) {   #KGE2
                coeff_row[9] <- OutputsCrit$SubCritValues[[2]]
                coeff_row[5] <- OutputsCrit$CritValue
                coeff_df <- rbind(coeff_df, coeff_row)
              }
            }
          }
        }
      }
    }
  }
}

#################################################################
#Print the tables of coefficient KGE and KGE'
coeff_df[, 4:9] <- apply(coeff_df[, 4:9], 2, as.numeric)
coeff_df[, 4:9] <- round(coeff_df[, 4:9], digits = 3)
coeff_df <- coeff_df[!is.na(coeff_df$gamma),]

dev_coeff_df <- coeff_df
colnames(dev_coeff_df) <- c("Per_Cal", "OF_Name", "Per_Val",
                            "Dev_KGE_Value", "Dev_KGE2_Value", "Dev_r",
"Dev_alpha", "Dev_beta", "Dev_gamma")
dev_coeff_df[4:9] <- abs((coeff_df[4:9])-1)
dev_coeff_df[, 4:9] <- apply(dev_coeff_df[, 4:9], 2, as.numeric)
dev_coeff_df <- rbind(dev_coeff_df, c("Abs_dev_min", NA, NA,
min(abs(dev_coeff_df[4])), min(abs(dev_coeff_df[5])),
                                      min(abs(dev_coeff_df[6])),
min(abs(dev_coeff_df[7])),
                                      min(abs(dev_coeff_df[8])),
min(abs(dev_coeff_df[9]))))
dev_coeff_df[, 4:9] <- apply(dev_coeff_df[, 4:9], 2, as.numeric)
dev_coeff_df <- rbind(dev_coeff_df, c("Abs_dev_max", NA, NA,
max(abs(dev_coeff_df[4])), max(abs(dev_coeff_df[5])),
                                      max(abs(dev_coeff_df[6])),
max(abs(dev_coeff_df[7])),
                                      max(abs(dev_coeff_df[8])),
max(abs(dev_coeff_df[9]))))
dev_coeff_df[, 4:9] <- apply(dev_coeff_df[, 4:9], 2, as.numeric)
dev_coeff_df[, 4:9] <- round(dev_coeff_df[, 4:9], digits = 3)
dev_coeff_df[, 4:9] <- round(dev_coeff_df[, 4:9], digits = 3)

mean_coeff <- c(mean(dev_coeff_df[, 6]), mean(dev_coeff_df[, 7]),
                mean(dev_coeff_df[, 8]), mean(dev_coeff_df[, 9]))

write.csv(coeff_df, "Coeff_criteria", col.names = TRUE)
write.csv(dev_coeff_df, "Dev_Coeff_criteria", col.names = TRUE)

library(gridExtra)
pdf("Coeff_criteria.pdf", height = 15, width = 15)
grid.table(coeff_df)
pdf("Dev_Coeff_criteria.pdf")
grid.table(dev_coeff_df)
dev.off()

#################################################################
#Store and print results in the data.frame

results_df[, 6:18] <- apply(results_df[, 6:18], 2, as.numeric)
results_df[, 6:18] <- round(results_df[, 6:18], digits = 3)

results_df <- results_df[!is.na(results_df$Model),]
write.csv(results_df, "Results_3basins_modified")

results_df$X <- NULL
library(gridExtra)
pdf("Results_3basins_Obs_X0310010.pdf", height = 20, width = 20)
grid.table(results_df[results_df$Basin %in% "Obs_X0310010", ])
pdf("Results_3basins_Obs_J5412110.pdf", height = 20, width = 20)
grid.table(results_df[results_df$Basin %in% "Obs_J5412110",])
pdf("Results_3basins_Obs_200105.pdf", height = 20, width = 20)
grid.table(results_df[results_df$Basin %in% "Obs_200105", ])
dev.off()

#################################################################
#Plot monthly "Melt" or "Psol" (depending on what is in the loops)
#over different calibrations

colnames(SNOW) <-  c("Month", "Cal1_NSE", "Cal1_KGE", "Cal1_KGE2",
"Cal2_NSE", "Cal2_KGE", "Cal2_KGE2")
colo_melt <- c("cadetblue1","cadetblue2", "cadetblue3", "coral",
"coral2", "coral3")
barplot(t(as.matrix(SNOW[2:7])), ylim = range(SNOW[2:7])*c(0,1.2),
        names.arg = month.abb, col = colo_melt, ylab = "Snow melt
[mm/month]",
        beside = T, main = c("Snow melt in basin", BASIN_char[[b]]))
legend("topright", legend = colnames(SNOW)[2:7], col = colo_melt, lwd =
1, cex = 0.7)
```

```
#################################################################
## Hydroclimatic regime of the three basins
# Aggregate data over each month (daily to monthly data)

Obs_X0310010$Basin <- c("Obs_X0310010")
Obs_J5412110$Basin <- c("Obs_J5412110")
Obs_200105$Basin <- c("Obs_200105")
Ob <- rbind(rbind(Obs_X0310010, Obs_J5412110), Obs_200105)

Ob_vec <- c("Obs_X0310010", "Obs_J5412110", "Obs_200105")
# La <- c(Layers_b1, Layers_b3)
Int <- data.frame(matrix(ncol = 5))
colnames(Int) <-  c("Basin","Month", "Agg_P", "Agg_Q", "Agg_S")

for (b in seq_along(Ob_vec)) {
  b = 1
  Obs <-  Ob[Ob$Basin == Ob_vec[b], ]
  #P and Q
  Obs$mY <- format(as.Date(Obs$Time), "%Y-%m")
  Obs <-  Obs[!is.na(Obs$Qmm),]
  len1 <- length(aggregate(Obs$Ptot, by = list(Obs$mY),
                           function (x) sum(x, na.rm = TRUE))$x)

  Month_Obs <- as.data.frame(matrix(nrow = len1, ncol = 4))
  colnames(Month_Obs) <-  c("mY", "month","Agg_P", "Agg_Q")

  Month_Obs$Agg_P <- aggregate(Obs$Ptot, by = list(Obs$mY),
                               function (x) sum(x, na.rm = TRUE))$x
  Month_Obs$Agg_Q <- aggregate(Obs$Qmm, by = list(Obs$mY),
                               function (x) sum(x, na.rm = TRUE))$x
  Month_Obs$mY <- aggregate(Obs$Ptot, by = list(Obs$mY),
                            function (x) sum(x, na.rm = TRUE))$Group.1
  Month_Obs$month <- substring(Month_Obs$mY, 6, 7)


  if (b !=2){
    La$my <- format(as.Date(La$Time), "%Y-%m")
    len2 <- length(aggregate(La$sum, by = list(La$my),
                             function (x) sum(x, na.rm = TRUE))$Group.1)
    SnowM <- as.data.frame(matrix(nrow = len2, ncol = 3))
    colnames(SnowM) <-  c("my", "month","Agg_S")
    SnowM$Agg_S <- aggregate(La$sum, by = list(La$my),
                             function (x) sum(x, na.rm = TRUE))$x
    SnowM$my <- aggregate(La$sum, by = list(La$my),
                          function (x) sum(x, na.rm = TRUE))$Group.1
    SnowM$month <- substring(SnowM$my, 6, 7)
  }
}

# Mean over the years per each month
Interann <- as.data.frame(matrix(ncol = 5, nrow = 12))
colnames(Interann) <-  c("Basin","Month", "Agg_P", "Agg_Q", "Agg_S")
Interann$Basin <- Ob_vec[b]
Interann$Agg_P <- (aggregate(Month_Obs$Agg_P,
                             by = list(Month_Obs$month),
                             function(x) mean(x, na.rm = TRUE)))$x
Interann$Agg_Q <- (aggregate(Month_Obs$Agg_Q,
                             by = list(Month_Obs$month),
                             function(x) mean(x, na.rm = TRUE)))$x
Interann$Month <- (aggregate(Month_Obs$Agg_Q,
                             by = list(Month_Obs$month),
                             function(x) mean(x, na.rm = TRUE)))$Group.1
if (b != 2){
  Interann$Agg_S <- (aggregate(SnowM$Agg_S,
                               by = list(SnowM$month),
                               function(x) mean(x, na.rm = TRUE)))$x
} else {
  Interann$Agg_S <- NA
}
Int <- rbind(Int, Interann)
Int <- Int[!is.na(Int$Basin),]


#Plot regime (interannual trend)
for (bas in seq_along(bas_char)){
  bas = b
  dev.new()
  Int_bas <- Int[Int$Basin ==BASIN_char[bas],]
  #pdf(paste("Hydroclimats.pdf"), width = , height = , pointsize = 12)
  par(oma = c(1,1,1,3), xpd=TRUE, mar = c(6, 4, 4, 4))
  plot(Int_bas$Agg_Q,
       ylim = range(Int_bas$Agg_Q, na.rm = TRUE)*c(0.1, 1.7),
       type = "l", lwd = 2, col = "red", main =
c("Basin",BASIN_char[[bas]]),
       xlab = "", ylab = "Qobs [mm/month]", xaxt = "n")
  axis(side = 2, col.axis = "red")
  axis(side = 1, labels = c(month.abb), at = Int_bas$Month)
  legend (x = 5, y = -(0.3*max(Int_bas$Agg_Q, na.rm = TRUE)) , legend =
c("Precipitation", "Snowfall"),
          col = c("navy", "cadetblue3"), cex = 1,
          pch = 15, bty = "n")

  if (b != 2) {
    par(new = TRUE)
    test <- rbind(Int_bas$Agg_P, Int_bas$Agg_S)
    barplot(test, ylim = rev(range(Int_bas$Agg_S))*c(3.5,0), ylab = "",
xaxt = "n", yaxt = "n", col = c("navy","cadetblue3"), beside = T)
    axis(side = 4, col.axis = "navy")
    mtext("Precipitation [mm/month]", side = 4, line = 3)
  } else {
    par(new = TRUE)
    barplot(Int_bas$Agg_P, ylim = rev(range(Int_bas$Agg_P))*c(3.5,0),
            col = "navy", ylab = "", xaxt = "n", yaxt = "n")
    axis(side = 4, col.axis = "navy")
    mtext("Precipitation [mm/month]", side = 4, line = 3)
  }
}


#################################################################
## Model that simulate better the discharge

#Analysis of all basins together
best_mod <- as.data.frame(matrix(nrow = 4, ncol = 4))
colnames(best_mod) <- CRIT_char
rownames(best_mod) <- MOD_char
```

```
for (mm in seq_along(MOD)) {
  for (ii in seq_along(CRIT)) {
    res_df <- results_df[results_df$Model == MOD_char[[mm]],]
    me <- mean(as.numeric(res_df[res_df$Crit_Name == CRIT_char[[ii]],
colnames(res_df) %in% "Crit_Value"]), na.rm = TRUE)
    best_mod[mm, ii] <- me
  }
}
best_mod <- round(best_mod, digits = 2)
# Optimum values: RMSE[Q] = 0, NSE[Q] = 1, KGE[Q] = 1, KGE'[Q] = 1

#Basin by basin analysis
best_mod_bas <- as.data.frame(matrix(nrow = 4, ncol = 6))
best_mod_tot <- as.data.frame(matrix(nrow = 12, ncol = 6))
colnames(best_mod_bas) <- c("Basin","Model", CRIT_char)
colnames(best_mod_tot) <- c("Basin","Model", CRIT_char)
best_mod_row <- vector(length = 5)
for (b in seq_along(BASIN)) {
  for (mm in seq_along(MOD)) {
    for (ii in seq_along(CRIT)) {
      res_bas <- results_df[results_df$Basin == BASIN_char[[b]],]
      res_df <- res_bas[res_bas$Model == MOD_char[[mm]],]
      best_mod_row[ii] <- mean(as.numeric(res_df[res_df$Crit_Name ==
CRIT_char[[ii]],
                                                  colnames(res_df) %in%
"Crit_Value"]), na.rm = TRUE)
    }
    ro <- c(BASIN_char[[b]], MOD_char[[mm]], best_mod_row)
    best_mod_bas <- rbind(best_mod_bas, ro)
  }
}

best_mod_bas <- as.data.frame(best_mod_bas[!is.na(best_mod_bas$Model),])
best_mod_bas[, 3:6] <-
round(as.matrix(as.data.frame(lapply(best_mod_bas[, 3:6], as.numeric))),
digits = 2)
rownames(best_mod_bas) <- NULL
# Chop off rows with NA and round numbers to 2 digits

library(gridExtra)
pdf("best_mod_bas.pdf", height=5, width=8)
grid.table(best_mod_bas)
dev.off()

pdf("best_mod.pdf", height=2, width=6)
grid.table(best_mod)
dev.off()

## Parameter values of GR4J+CemaNeige changing OF and Cal_Per (1)
results_df <- results_df[!is.na(results_df$Basin),]
res_CemaNeigeGR4J <- results_df[results_df$Model ==
"RunModel_CemaNeigeGR4J",]
OF <- c("NSE[Q]", "KGE[Q]", "KGE'[Q]")
par_CemaNeigeGR4J <- c("X1", "X2", "X3", "X4", "C1", "C2")
par_CemaNeigeGR4J_title <- c("X1 [mm]", "X2 [mm/d]", "X3 [mm]", "X4
[day]", "C1 [-]", "")
sym <- 19 #sym <- c(19, 19, 1, 1, 15, 15)
```

```
colo <- c("green4", "red", "royalblue")
Ylim_1 <- list(c(0, 1000), c(-2, 4), c(0, 500), c(1.1, 2.3), c(0, 1),
c(2, 15))

pdf("Param.pdf", height = 6, width = 9)
for (p in seq_along(par_CemaNeigeGR4J)) {
  p = 6
  res_CemaNeigeGR4J_par <-
as.numeric(res_CemaNeigeGR4J[res_CemaNeigeGR4J$OF_Name == OF[[1]] &

((res_CemaNeigeGR4J$Per_Cal == "1961-1985" & res_CemaNeigeGR4J$Per_Val ==
"1961-1985") |

(res_CemaNeigeGR4J$Per_Cal == "1986-2010" & res_CemaNeigeGR4J$Per_Val ==
"1961-1985") |

(res_CemaNeigeGR4J$Per_Cal == "1994-2008" & res_CemaNeigeGR4J$Per_Val ==
"1994-2008") |

(res_CemaNeigeGR4J$Per_Cal == "1978-1993" & res_CemaNeigeGR4J$Per_Val ==
"1994-2008")),

par_CemaNeigeGR4J[[p]]])

  bbb <- c(paste("Cal1"), paste("Cal2"),
           paste("Cal1"), paste("Cal2"),
           paste("Cal1"), paste("Cal2"))
  par(oma = c(1,1,1,4), xpd=TRUE, mar = c(5, 4, 4, 8))
  # xpd enable things to be drawn outside the plot region; oma boarderds
of the window; mar boarders of the printed thing

  if (p == 6) {
    plot(res_CemaNeigeGR4J_par,   type = "n",
         xlim = c(0.5, 6.5), ylim = Ylim_1[[p]], cex = 0.5, xaxt = "n",
         xlab = "Calibration periods per basin", ylab = "Parameter
value", main = expression(paste("C2 [mm/", degree, "C/day]")))
    } else {
      plot(res_CemaNeigeGR4J_par,   type = "n",
             xlim = c(0.5, 6.5), ylim = Ylim_1[[p]], cex = 0.5, xaxt =
"n",
             xlab = "Calibration periods per basin", ylab = "Parameter
value", main = paste(par_CemaNeigeGR4J_title[[p]]))
    }

  axis(side = 1, labels = bbb , at = 1:6, cex.axis = 0.8)
  for (of in seq_along(OF)) {
    # res_CemaNeigeGR4J_par <-
res_CemaNeigeGR4J[res_CemaNeigeGR4J$OF_Name == OF[[of]],
par_CemaNeigeGR4J[[p]]]
    res_CemaNeigeGR4J_par <-
as.numeric(res_CemaNeigeGR4J[res_CemaNeigeGR4J$OF_Name == OF[[of]] &

((res_CemaNeigeGR4J$Per_Cal == "1961-1985" & res_CemaNeigeGR4J$Per_Val ==
"1961-1985") |

(res_CemaNeigeGR4J$Per_Cal == "1986-2010" & res_CemaNeigeGR4J$Per_Val ==
"1961-1985") |
```

```
(res_CemaNeigeGR4J$Per_Cal == "1994-2008" & res_CemaNeigeGR4J$Per_Val ==
"1994-2008") |

(res_CemaNeigeGR4J$Per_Cal == "1978-1993" & res_CemaNeigeGR4J$Per_Val ==
"1994-2008")),

par_CemaNeigeGR4J[[p]]])
    #barplot(res_CemaNeigeGR4J_par)
    points(res_CemaNeigeGR4J_par,   type = "b",
           pch = sym, cex = 1, col = colo[[of]], lwd = 0)
    legend(6.8, Ylim_1[[p]][2], legend = OF, col = colo, lwd = 1, cex =
0.7)
    segments(x0 = 2.5, y0 = Ylim_1[[p]][1], x1 = 2.5, y1 = Ylim_1[[p]]
[2], col = "black", lty = 1)
    segments(x0 = 4.5, y0 = Ylim_1[[p]][1], x1 = 4.5, y1 = Ylim_1[[p]]
[2], col = "black", lty = 1)
    segments(x0 = 1.5, y0 = Ylim_1[[p]][1], x1 = 1.5, y1 = Ylim_1[[p]]
[2]*0.95, col = "gray60", lty = 3)
    segments(x0 = 3.5, y0 = Ylim_1[[p]][1], x1 = 3.5, y1 = Ylim_1[[p]]
[2]*0.95, col = "gray60", lty = 3)
    segments(x0 = 5.5, y0 = Ylim_1[[p]][1], x1 = 5.5, y1 = Ylim_1[[p]]
[2]*0.95, col = "gray60", lty = 3)
    box(which = "plot")
    text("France on Alps", x = 1.5, y = Ylim_1[[p]][2], cex = 0.8)
    text("France in Brittany", x = 3.5, y = Ylim_1[[p]][2], cex = 0.8)
    text("Austria on Alps", x = 5.5, y = Ylim_1[[p]][2], cex = 0.8)
    }
}
dev.off()


## Parameter values of GR4J+CemaNeige changing OF and Cal_Per (2)

results_df <- results_df[!is.na(results_df$Basin),]
res_CemaNeigeGR4J <- results_df[results_df$Model ==
"RunModel_CemaNeigeGR4J",]
OF <- c("NSE[Q]", "KGE[Q]", "KGE'[Q]")
par_CemaNeigeGR4J <- c("X1", "X2", "X3", "X4", "C1", "C2")
par_CemaNeigeGR4J_title <- c("X1 [mm]", "X2 [mm/d]", "X3 [mm]", "X4
[day]", "C1 [-]", "C2 [mm/?C/day]")
sym <- 19 #sym <- c(19, 19, 1, 1, 15, 15)
colo <- c("green4", "red", "royalblue")
Ylim_1 <- list(c(0, 1000), c(-2, 4), c(0, 500), c(1.1, 2.3), c(0, 1),
c(2, 15))

pdf("Param.pdf", height = 6, width = 9)
for (p in seq_along(par_CemaNeigeGR4J)) {
  res_CemaNeigeGR4J_par <-
as.numeric(res_CemaNeigeGR4J[res_CemaNeigeGR4J$OF_Name == OF[[1]] &

((res_CemaNeigeGR4J$Per_Cal == "1961-1985" & res_CemaNeigeGR4J$Per_Val ==
"1961-1985") |

(res_CemaNeigeGR4J$Per_Cal == "1986-2010" & res_CemaNeigeGR4J$Per_Val ==
"1961-1985") |

(res_CemaNeigeGR4J$Per_Cal == "1994-2008" & res_CemaNeigeGR4J$Per_Val ==
"1994-2008") |

(res_CemaNeigeGR4J$Per_Cal == "1978-1993" & res_CemaNeigeGR4J$Per_Val ==
"1994-2008")),

par_CemaNeigeGR4J[[p]]])

  bbb <- c(paste("Basin1", "\n Cal1"), paste("Basin1", "\n Cal2"),
           paste("Basin2", "\n Cal1"), paste("Basin2", "\n Cal2"),
           paste("Basin3", "\n Cal1"), paste("Basin3", "\n Cal2"))
  par(oma = c(1,1,1,4), xpd=TRUE, mar = c(5, 4, 4, 8))
  # xpd enable things to be drawn outside the plot region; oma boarderds
of the window; mar boarders of the printed thing
  plot(res_CemaNeigeGR4J_par,   type = "n",
       xlim = c(0.5, 6.5), ylim = Ylim_1[[p]], cex = 0.5, xaxt = "n",
       xlab = "Calibration periods per basin", ylab = "Parameter value",
main = paste(par_CemaNeigeGR4J_title[[p]]))
  axis(side = 1, labels = bbb , at = 1:6, cex.axis = 0.7)
  for (of in seq_along(OF)) {
    # res_CemaNeigeGR4J_par <-
res_CemaNeigeGR4J[res_CemaNeigeGR4J$OF_Name == OF[[of]],
par_CemaNeigeGR4J[[p]]]
    res_CemaNeigeGR4J_par <-
as.numeric(res_CemaNeigeGR4J[res_CemaNeigeGR4J$OF_Name == OF[[of]] &

((res_CemaNeigeGR4J$Per_Cal == "1961-1985" & res_CemaNeigeGR4J$Per_Val ==
"1961-1985") |

(res_CemaNeigeGR4J$Per_Cal == "1986-2010" & res_CemaNeigeGR4J$Per_Val ==
"1961-1985") |

(res_CemaNeigeGR4J$Per_Cal == "1994-2008" & res_CemaNeigeGR4J$Per_Val ==
"1994-2008") |

(res_CemaNeigeGR4J$Per_Cal == "1978-1993" & res_CemaNeigeGR4J$Per_Val ==
"1994-2008")),

par_CemaNeigeGR4J[[p]]])
    points(res_CemaNeigeGR4J_par,   type = "b",
           pch = sym, cex = 1, col = colo[[of]])
    legend(6.8, Ylim_1[[p]][2], legend = OF, col = colo, lwd = 1, cex =
0.7)
    legend(6.8, Ylim_1[[p]][2]-0.2*(Ylim_1[[p]][2]-Ylim_1[[p]][1]),
           legend = c("Basin1: Obs_X0310010", "Basin2: Obs_J5412110",
"Basin3: Obs_200105"),
           cex = 0.7)
    segments(x0 = 2.5, y0 = Ylim_1[[p]][1], x1 = 2.5, y1 = Ylim_1[[p]]
[2], col = "gray60", lty = 3)
    segments(x0 = 4.5, y0 = Ylim_1[[p]][1], x1 = 4.5, y1 = Ylim_1[[p]]
[2], col = "gray60", lty = 3)
  }
}
dev.off()

#################################################################
#Mean of temperature and precipitation
# Per_Cal_FR <- c("1961-1985", "1986-2010")
# Per_Cal_AU <- c("1978-1993", "1994-2008")
# Obs_X0310010$Basin <- c("Obs_X0310010")
```

```
# Obs_J5412110$Basin <- c("Obs_J5412110")
# Obs_200105$Basin <- c("Obs_200105")
# Ob <- rbind(rbind(Obs_X0310010, Obs_J5412110), Obs_200105)

Mean_df <- as.data.frame(matrix(ncol = 7, nrow = 3))
Mean_df[,1] <- c("Mean_X0310010","Mean_J5412110", "Mean_200105")
Mean_X0310010 <- c(mean(Obs_X0310010[Obs_X0310010$Time %in% seq(from =
as.POSIXlt("1961-01-01", "%Y-%m-%d"), to = as.POSIXlt("1985-12-31", "%Y-
%m-%d"), by = "1 day"),
                                      colnames(Obs_X0310010) %in% "Ptot"],
na.rm = TRUE),
                   mean(Obs_X0310010[Obs_X0310010$Time %in% seq(from =
as.POSIXlt("1986-01-01", "%Y-%m-%d"), to = as.POSIXlt("2010-07-31", "%Y-
%m-%d"), by = "1 day"),
                                      colnames(Obs_X0310010) %in% "Ptot"],
na.rm = TRUE),
                   mean(Obs_X0310010[Obs_X0310010$Time %in% seq(from =
as.POSIXlt("1961-01-01", "%Y-%m-%d"), to = as.POSIXlt("1985-12-31", "%Y-
%m-%d"), by = "1 day"),
                                      colnames(Obs_X0310010) %in% "Temp"],
na.rm = TRUE),
                   mean(Obs_X0310010[Obs_X0310010$Time %in% seq(from =
as.POSIXlt("1986-01-01", "%Y-%m-%d"), to = as.POSIXlt("2010-07-31", "%Y-
%m-%d"), by = "1 day"),
                                      colnames(Obs_X0310010) %in% "Temp"],
na.rm = TRUE),
                   mean(Obs_X0310010[Obs_X0310010$Time %in% seq(from =
as.POSIXlt("1961-01-01", "%Y-%m-%d"), to = as.POSIXlt("1985-12-31", "%Y-
%m-%d"), by = "1 day"),
                                      colnames(Obs_X0310010) %in% "Qmm"],
na.rm = TRUE),
                   mean(Obs_X0310010[Obs_X0310010$Time %in% seq(from =
as.POSIXlt("1986-01-01", "%Y-%m-%d"), to = as.POSIXlt("2010-07-31", "%Y-
%m-%d"), by = "1 day"),
                                      colnames(Obs_X0310010) %in% "Qmm"],
na.rm = TRUE))

Mean_J5412110 <- c(mean(Obs_J5412110[Obs_J5412110$Time %in% seq(from =
as.POSIXlt("1961-01-01", "%Y-%m-%d"), to = as.POSIXlt("1985-12-31", "%Y-
%m-%d"), by = "1 day"),
                                      colnames(Obs_J5412110) %in% "Ptot"],
na.rm = TRUE),
                   mean(Obs_J5412110[Obs_J5412110$Time %in% seq(from =
as.POSIXlt("1986-01-01", "%Y-%m-%d"), to = as.POSIXlt("2010-07-31", "%Y-
%m-%d"), by = "1 day"),
                                      colnames(Obs_J5412110) %in% "Ptot"],
na.rm = TRUE),
                   mean(Obs_J5412110[Obs_J5412110$Time %in% seq(from =
as.POSIXlt("1961-01-01", "%Y-%m-%d"), to = as.POSIXlt("1985-12-31", "%Y-
%m-%d"), by = "1 day"),
                                      colnames(Obs_J5412110) %in% "Temp"],
na.rm = TRUE),
                   mean(Obs_J5412110[Obs_J5412110$Time %in% seq(from =
as.POSIXlt("1986-01-01", "%Y-%m-%d"), to = as.POSIXlt("2010-07-31", "%Y-
%m-%d"), by = "1 day"),
                                      colnames(Obs_J5412110) %in% "Temp"],
na.rm = TRUE),
                   mean(Obs_J5412110[Obs_J5412110$Time %in% seq(from =
as.POSIXlt("1961-01-01", "%Y-%m-%d"), to = as.POSIXlt("1985-12-31", "%Y-
%m-%d"), by = "1 day"),
                                      colnames(Obs_J5412110) %in% "Qmm"],
na.rm = TRUE),
                   mean(Obs_J5412110[Obs_J5412110$Time %in% seq(from =
as.POSIXlt("1986-01-01", "%Y-%m-%d"), to = as.POSIXlt("2010-07-31", "%Y-
%m-%d"), by = "1 day"),
                                      colnames(Obs_J5412110) %in% "Qmm"],
na.rm = TRUE))

Mean_200105 <- c(mean(Obs_200105[Obs_200105$Time %in% seq(from =
as.POSIXlt("1978-01-01", "%Y-%m-%d"), to = as.POSIXlt("1993-12-31", "%Y-
%m-%d"), by = "1 day"),
                                  colnames(Obs_200105) %in% "Ptot"], na.rm
= TRUE),
                 mean(Obs_200105[Obs_200105$Time %in% seq(from =
as.POSIXlt("1994-01-01", "%Y-%m-%d"), to = as.POSIXlt("2008-12-31", "%Y-
%m-%d"), by = "1 day"),
                                  colnames(Obs_200105) %in% "Ptot"], na.rm
= TRUE),
                 mean(Obs_200105[Obs_200105$Time %in% seq(from =
as.POSIXlt("1978-01-01", "%Y-%m-%d"), to = as.POSIXlt("1993-12-31", "%Y-
%m-%d"), by = "1 day"),
                                  colnames(Obs_200105) %in% "Temp"], na.rm
= TRUE),
                 mean(Obs_200105[Obs_200105$Time %in% seq(from =
as.POSIXlt("1994-01-01", "%Y-%m-%d"), to = as.POSIXlt("2008-12-31", "%Y-
%m-%d"), by = "1 day"),
                                  colnames(Obs_200105) %in% "Temp"], na.rm
= TRUE),
                 mean(Obs_200105[Obs_200105$Time %in% seq(from =
as.POSIXlt("1978-01-01", "%Y-%m-%d"), to = as.POSIXlt("1993-12-31", "%Y-
%m-%d"), by = "1 day"),
                                  colnames(Obs_200105) %in% "Qmm"], na.rm
= TRUE),
                 mean(Obs_200105[Obs_200105$Time %in% seq(from =
as.POSIXlt("1994-01-01", "%Y-%m-%d"), to = as.POSIXlt("2008-12-31", "%Y-
%m-%d"), by = "1 day"),
                                  colnames(Obs_200105) %in% "Qmm"], na.rm
= TRUE))
Mean_df[,2:7] <- rbind(rbind(Mean_X0310010, Mean_J5412110), Mean_200105)
colnames(Mean_df) <- c("Basin",
                       paste("Ptot_Cal1", "\n", "[mm/day]"),
paste("Ptot_Cal2", "\n", "[mm/day]"),
                       paste("Temp_Cal1", "\n", "[?C]"),
paste("Temp_Cal2", "\n", "[?C]"),
                       paste("Qobs_Cal1", "\n", "[mm/day]"),
paste("Qobs_Cal2", "\n", "[mm/day]"))
rownames(Mean_df) <- NULL

Mean_df <- round(Mean_df, digits = 2)

library(gridExtra)
pdf("mean_PandT.pdf", height=2, width=8)
grid.table(Mean_df)
dev.off()

big.df <- results_df[1:20, 2:17]
```

```
library(gridExtra)
pdf("big.df.pdf", height=6, width=13)
grid.table(big.df)
dev.off()

# Water yield
Cal1 <- seq(which(format(Obs_J5412110$Time, format = "%Y-%m-%d")=="1961-
01-01"),
            which(format(Obs_J5412110$Time, format = "%Y-%m-%d")=="1985-
12-31"))
Cal2 <- seq(which(format(Obs_J5412110$Time, format = "%Y-%m-%d")=="1986-
01-01"),
            which(format(Obs_J5412110$Time, format = "%Y-%m-%d")=="2010-
07-31"))
mean(Obs_J5412110$Qmm[Cal1], na.rm = TRUE)
mean(Obs_J5412110$Ptot[Cal1], na.rm = TRUE)
mean(Obs_J5412110$Qmm[Cal1], na.rm = TRUE)/mean(Obs_J5412110$Ptot[Cal1],
na.rm = TRUE)
mean(Obs_J5412110$Qmm[Cal2], na.rm = TRUE)
mean(Obs_J5412110$Ptot[Cal2], na.rm = TRUE)
mean(Obs_J5412110$Qmm[Cal2], na.rm = TRUE)/mean(Obs_J5412110$Ptot[Cal2],
na.rm = TRUE)
```

Appendix B

Dataframe of results

	Basin	Model	Per_Cal	Per_Val	OF_Name	OF_Value	X1	X2	X3	X4	X5	X6	C1	C2	RMSE	NSE	KGE	KGE2
1	Obs_X0310010	RunModel_GR4J	1961−1985	1961−1985	RMSE[Q]	1.704	1671	4.141	4501.000	1.66	NA	NA	NA	NA	1.704	0.111	0.035	0.033
2	Obs_X0310010	RunModel_GR4J	1961−1985	1986−2010	RMSE[Q]	1.704	1671	4.141	4501.000	1.66	NA	NA	NA	NA	1.474	0.096	0.106	0.088
3	Obs_X0310010	RunModel_GR4J	1961−1985	1961−1985	NSE[Q]	0.111	1671	4.141	4501.000	1.66	NA	NA	NA	NA	1.704	0.111	0.035	0.033
4	Obs_X0310010	RunModel_GR4J	1961−1985	1986−2010	NSE[Q]	0.111	1671	4.141	4501.000	1.66	NA	NA	NA	NA	1.474	0.096	0.106	0.088
5	Obs_X0310010	RunModel_GR4J	1961−1985	1961−1985	KGE[Q]	0.132	2595	0.110	8.399	2.00	NA	NA	NA	NA	2.292	−0.609	0.132	0.130
6	Obs_X0310010	RunModel_GR4J	1961−1985	1986−2010	KGE[Q]	0.132	2595	0.110	8.399	2.00	NA	NA	NA	NA	2.037	−0.726	0.199	0.202
7	Obs_X0310010	RunModel_GR4J	1961−1985	1961−1985	KGE'[Q]	0.133	2723	0.000	8.313	2.00	NA	NA	NA	NA	2.262	−0.566	0.130	0.133
8	Obs_X0310010	RunModel_GR4J	1961−1985	1986−2010	KGE'[Q]	0.133	2723	0.000	8.313	2.00	NA	NA	NA	NA	1.998	−0.660	0.206	0.207
9	Obs_X0310010	RunModel_GR4J	1986−2010	1961−1985	RMSE[Q]	1.462	2702	−3.552	4772.815	1.63	NA	NA	NA	NA	1.716	0.099	−0.007	0.004
10	Obs_X0310010	RunModel_GR4J	1986−2010	1986−2010	RMSE[Q]	1.462	2702	−3.552	4772.815	1.63	NA	NA	NA	NA	1.462	0.110	0.069	0.070
11	Obs_X0310010	RunModel_GR4J	1986−2010	1961−1985	NSE[Q]	0.110	2702	−3.552	4772.815	1.63	NA	NA	NA	NA	1.716	0.099	−0.007	0.004
12	Obs_X0310010	RunModel_GR4J	1986−2010	1986−2010	NSE[Q]	0.110	2702	−3.552	4772.815	1.63	NA	NA	NA	NA	1.462	0.110	0.069	0.070
13	Obs_X0310010	RunModel_GR4J	1986−2010	1961−1985	KGE[Q]	0.219	2352	0.071	31.050	2.00	NA	NA	NA	NA	2.125	−0.383	0.113	0.117
14	Obs_X0310010	RunModel_GR4J	1986−2010	1986−2010	KGE[Q]	0.219	2352	0.071	31.050	2.00	NA	NA	NA	NA	1.849	−0.422	0.219	0.214
15	Obs_X0310010	RunModel_GR4J	1986−2010	1961−1985	KGE'[Q]	0.219	2368	−0.181	30.569	1.99	NA	NA	NA	NA	2.111	−0.365	0.102	0.120
16	Obs_X0310010	RunModel_GR4J	1986−2010	1986−2010	KGE'[Q]	0.219	2368	−0.181	30.569	1.99	NA	NA	NA	NA	1.825	−0.386	0.215	0.219
17	Obs_X0310010	RunModel_GR5J	1961−1985	1961−1985	RMSE[Q]	1.705	1657	−0.088	4523.440	1.67	0.697	NA	NA	NA	1.705	0.110	0.029	0.028
18	Obs_X0310010	RunModel_GR5J	1961−1985	1986−2010	RMSE[Q]	1.705	1657	−0.088	4523.440	1.67	0.697	NA	NA	NA	1.471	0.099	0.099	0.083
19	Obs_X0310010	RunModel_GR5J	1961−1985	1961−1985	NSE[Q]	0.110	1657	−0.088	4523.440	1.67	0.697	NA	NA	NA	1.705	0.110	0.029	0.028
20	Obs_X0310010	RunModel_GR5J	1961−1985	1986−2010	NSE[Q]	0.110	1657	−0.088	4523.440	1.67	0.697	NA	NA	NA	1.471	0.099	0.099	0.083
21	Obs_X0310010	RunModel_GR5J	1961−1985	1961−1985	KGE[Q]	0.211	3744	5.180	22.911	1.33	0.717	NA	NA	NA	2.270	−0.578	0.211	0.210
22	Obs_X0310010	RunModel_GR5J	1961−1985	1986−2010	KGE[Q]	0.211	3744	5.180	22.911	1.33	0.717	NA	NA	NA	2.117	−0.865	0.225	0.206
23	Obs_X0310010	RunModel_GR5J	1961−1985	1961−1985	KGE'[Q]	0.203	3593	4.373	18.715	1.42	0.752	NA	NA	NA	2.223	−0.513	0.202	0.203
24	Obs_X0310010	RunModel_GR5J	1961−1985	1986−2010	KGE'[Q]	0.203	3593	4.373	18.715	1.42	0.752	NA	NA	NA	2.047	−0.743	0.241	0.206
25	Obs_X0310010	RunModel_GR5J	1986−2010	1961−1985	RMSE[Q]	1.461	2175	−2.859	5328.824	1.64	0.253	NA	NA	NA	1.719	0.096	−0.018	−0.006
26	Obs_X0310010	RunModel_GR5J	1986−2010	1986−2010	RMSE[Q]	1.461	2175	−2.859	5328.824	1.64	0.253	NA	NA	NA	1.461	0.112	0.053	0.053
27	Obs_X0310010	RunModel_GR5J	1986−2010	1961−1985	NSE[Q]	0.112	2175	−2.859	5328.824	1.64	0.253	NA	NA	NA	1.719	0.096	−0.018	−0.006
28	Obs_X0310010	RunModel_GR5J	1986−2010	1986−2010	NSE[Q]	0.112	2175	−2.859	5328.824	1.64	0.253	NA	NA	NA	1.461	0.112	0.053	0.053
29	Obs_X0310010	RunModel_GR5J	1986−2010	1961−1985	KGE[Q]	0.286	3041	3.899	83.931	1.19	0.581	NA	NA	NA	2.023	−0.253	0.133	0.129
30	Obs_X0310010	RunModel_GR5J	1986−2010	1986−2010	KGE[Q]	0.286	3041	3.899	83.931	1.19	0.581	NA	NA	NA	1.753	−0.279	0.286	0.281
31	Obs_X0310010	RunModel_GR5J	1986−2010	1961−1985	KGE'[Q]	0.220	2451	−0.110	30.036	1.40	0.000	NA	NA	NA	2.091	−0.339	0.093	0.116
32	Obs_X0310010	RunModel_GR5J	1986−2010	1986−2010	KGE'[Q]	0.220	2451	−0.110	30.036	1.40	0.000	NA	NA	NA	1.799	−0.347	0.213	0.220
33	Obs_X0310010	RunModel_GR6J	1961−1985	1961−1985	RMSE[Q]	1.701	1620	3.265	2677.185	1.65	0.267	109.08	NA	NA	1.701	0.114	0.069	0.068
34	Obs_X0310010	RunModel_GR6J	1961−1985	1986−2010	RMSE[Q]	1.701	1620	3.265	2677.185	1.65	0.267	109.08	NA	NA	1.491	0.075	0.154	0.140
35	Obs_X0310010	RunModel_GR6J	1961−1985	1961−1985	NSE[Q]	0.114	1620	3.265	2677.185	1.65	0.267	109.08	NA	NA	1.701	0.114	0.069	0.068
36	Obs_X0310010	RunModel_GR6J	1961−1985	1986−2010	NSE[Q]	0.114	1620	3.265	2677.185	1.65	0.267	109.08	NA	NA	1.491	0.075	0.154	0.140
37	Obs_X0310010	RunModel_GR6J	1961−1985	1961−1985	KGE[Q]	0.091	0	−0.340	84.783	1.67	−0.044	145.92	NA	NA	2.441	−0.825	0.091	0.091
38	Obs_X0310010	RunModel_GR6J	1961−1985	1986−2010	KGE[Q]	0.091	0	−0.340	84.783	1.67	−0.044	145.92	NA	NA	2.194	−1.002	0.167	0.185
39	Obs_X0310010	RunModel_GR6J	1961−1985	1961−1985	KGE'[Q]	0.250	1882	2.820	11.246	1.92	0.724	44.26	NA	NA	2.087	−0.333	0.240	0.250
40	Obs_X0310010	RunModel_GR6J	1961−1985	1986−2010	KGE'[Q]	0.250	1882	2.820	11.246	1.92	0.724	44.26	NA	NA	1.909	−0.516	0.281	0.242
41	Obs_X0310010	RunModel_GR6J	1986−2010	1961−1985	RMSE[Q]	1.461	2084	−2.196	3595.144	1.66	0.248	121.06	NA	NA	1.720	0.094	−0.021	−0.010
42	Obs_X0310010	RunModel_GR6J	1986−2010	1986−2010	RMSE[Q]	1.461	2084	−2.196	3595.144	1.66	0.248	121.06	NA	NA	1.461	0.111	0.050	0.050
43	Obs_X0310010	RunModel_GR6J	1986−2010	1961−1985	NSE[Q]	0.111	2084	−2.196	3595.144	1.66	0.248	121.06	NA	NA	1.720	0.094	−0.021	−0.010
44	Obs_X0310010	RunModel_GR6J	1986−2010	1986−2010	NSE[Q]	0.111	2084	−2.196	3595.144	1.66	0.248	121.06	NA	NA	1.461	0.111	0.050	0.050
45	Obs_X0310010	RunModel_GR6J	1986−2010	1961−1985	KGE[Q]	0.217	0	−0.263	181.506	1.53	−0.493	151.46	NA	NA	2.167	−0.438	0.058	0.071
46	Obs_X0310010	RunModel_GR6J	1986−2010	1986−2010	KGE[Q]	0.217	0	−0.263	181.506	1.53	−0.493	151.46	NA	NA	1.882	−0.474	0.217	0.217
47	Obs_X0310010	RunModel_GR6J	1986−2010	1961−1985	KGE'[Q]	0.227	2339	−0.128	0.007	2.14	0.478	9.12	NA	NA	2.060	−0.299	0.091	0.122
48	Obs_X0310010	RunModel_GR6J	1986−2010	1986−2010	KGE'[Q]	0.227	2339	−0.128	0.007	2.14	0.478	9.12	NA	NA	1.769	−0.302	0.213	0.227
49	Obs_X0310010	RunModel_CemaNeigeGR4J	1961−1985	1961−1985	RMSE[Q]	0.673	493	1.404	390.820	1.38	NA	NA	0.209	5.19	0.673	0.861	0.908	0.912
50	Obs_X0310010	RunModel_CemaNeigeGR4J	1961−1985	1986−2010	RMSE[Q]	0.673	493	1.404	390.820	1.38	NA	NA	0.209	5.19	0.636	0.832	0.889	0.909
51	Obs_X0310010	RunModel_CemaNeigeGR4J	1961−1985	1961−1985	NSE[Q]	0.861	493	1.404	390.820	1.38	NA	NA	0.209	5.19	0.673	0.861	0.908	0.912
52	Obs_X0310010	RunModel_CemaNeigeGR4J	1961−1985	1986−2010	NSE[Q]	0.861	493	1.404	390.820	1.38	NA	NA	0.209	5.19	0.636	0.832	0.889	0.909
53	Obs_X0310010	RunModel_CemaNeigeGR4J	1961−1985	1961−1985	KGE[Q]	0.928	419	1.466	347.617	1.38	NA	NA	0.057	4.76	0.684	0.857	0.928	0.928
54	Obs_X0310010	RunModel_CemaNeigeGR4J	1961−1985	1986−2010	KGE[Q]	0.928	419	1.466	347.617	1.38	NA	NA	0.057	4.76	0.656	0.821	0.843	0.888
55	Obs_X0310010	RunModel_CemaNeigeGR4J	1961−1985	1961−1985	KGE'[Q]	0.930	735	1.100	257.822	1.35	NA	NA	0.864	6.34	0.677	0.860	0.930	0.930
56	Obs_X0310010	RunModel_CemaNeigeGR4J	1961−1985	1986−2010	KGE'[Q]	0.930	735	1.100	257.822	1.35	NA	NA	0.864	6.34	0.693	0.800	0.827	0.874
57	Obs_X0310010	RunModel_CemaNeigeGR4J	1986−2010	1961−1985	RMSE[Q]	0.595	665	0.120	249.635	1.36	NA	NA	0.018	4.77	0.708	0.846	0.840	0.883
58	Obs_X0310010	RunModel_CemaNeigeGR4J	1986−2010	1986−2010	RMSE[Q]	0.595	665	0.120	249.635	1.36	NA	NA	0.018	4.77	0.595	0.853	0.922	0.921
59	Obs_X0310010	RunModel_CemaNeigeGR4J	1986−2010	1961−1985	NSE[Q]	0.853	665	0.120	249.635	1.36	NA	NA	0.018	4.77	0.708	0.846	0.840	0.883
60	Obs_X0310010	RunModel_CemaNeigeGR4J	1986−2010	1986−2010	NSE[Q]	0.853	665	0.120	249.635	1.36	NA	NA	0.018	4.77	0.595	0.853	0.922	0.921
61	Obs_X0310010	RunModel_CemaNeigeGR4J	1986−2010	1961−1985	KGE[Q]	0.926	656	0.377	254.177	1.35	NA	NA	0.018	4.78	0.697	0.851	0.857	0.891
62	Obs_X0310010	RunModel_CemaNeigeGR4J	1986−2010	1986−2010	KGE[Q]	0.926	656	0.377	254.177	1.35	NA	NA	0.018	4.78	0.597	0.852	0.926	0.926
63	Obs_X0310010	RunModel_CemaNeigeGR4J	1986−2010	1961−1985	KGE'[Q]	0.926	738	0.397	223.401	1.36	NA	NA	0.460	4.83	0.699	0.851	0.855	0.888
64	Obs_X0310010	RunModel_CemaNeigeGR4J	1986−2010	1986−2010	KGE'[Q]	0.926	738	0.397	223.401	1.36	NA	NA	0.460	4.83	0.598	0.851	0.926	0.926

	Basin	Model	Per_Cal	Per_Val	OF_Name	OF_Value	X1	X2	X3	X4	X5	X6	C1	C2	RMSE	NSE	KGE	KGE2
65	Obs_J5412110	RunModel_GR4J	1961−1985	1961−1985	RMSE[Q]	0.420	347.695	0.429	188.807	2.012	NA	NA	NA	NA	0.420	0.915	0.938	0.941
66	Obs_J5412110	RunModel_GR4J	1961−1985	1986−2010	RMSE[Q]	0.420	347.695	0.429	188.807	2.012	NA	NA	NA	NA	0.631	0.840	0.773	0.812
67	Obs_J5412110	RunModel_GR4J	1961−1985	1961−1985	NSE[Q]	0.915	347.695	0.429	188.807	2.012	NA	NA	NA	NA	0.420	0.915	0.938	0.941
68	Obs_J5412110	RunModel_GR4J	1961−1985	1986−2010	NSE[Q]	0.915	347.695	0.429	188.807	2.012	NA	NA	NA	NA	0.631	0.840	0.773	0.812
69	Obs_J5412110	RunModel_GR4J	1961−1985	1961−1985	KGE[Q]	0.955	337.348	0.410	154.336	2.063	NA	NA	NA	NA	0.432	0.910	0.955	0.955
70	Obs_J5412110	RunModel_GR4J	1961−1985	1986−2010	KGE[Q]	0.955	337.348	0.410	154.336	2.063	NA	NA	NA	NA	0.693	0.807	0.740	0.808
71	Obs_J5412110	RunModel_GR4J	1961−1985	1961−1985	KGE'[Q]	0.955	309.899	0.345	176.690	2.038	NA	NA	NA	NA	0.430	0.911	0.955	0.955
72	Obs_J5412110	RunModel_GR4J	1961−1985	1986−2010	KGE'[Q]	0.955	309.899	0.345	176.690	2.038	NA	NA	NA	NA	0.680	0.814	0.747	0.814
73	Obs_J5412110	RunModel_GR4J	1986−2010	1961−1985	RMSE[Q]	0.489	328.963	−1.940	278.148	1.779	NA	NA	NA	NA	0.543	0.859	0.735	0.825
74	Obs_J5412110	RunModel_GR4J	1986−2010	1986−2010	RMSE[Q]	0.489	328.963	−1.940	278.148	1.779	NA	NA	NA	NA	0.489	0.904	0.935	0.935
75	Obs_J5412110	RunModel_GR4J	1986−2010	1961−1985	NSE[Q]	0.904	328.963	−1.940	278.148	1.779	NA	NA	NA	NA	0.543	0.859	0.735	0.825
76	Obs_J5412110	RunModel_GR4J	1986−2010	1986−2010	NSE[Q]	0.904	328.963	−1.940	278.148	1.779	NA	NA	NA	NA	0.489	0.904	0.935	0.935
77	Obs_J5412110	RunModel_GR4J	1986−2010	1961−1985	KGE[Q]	0.949	323.540	−1.665	227.737	2.053	NA	NA	NA	NA	0.510	0.875	0.763	0.836
78	Obs_J5412110	RunModel_GR4J	1986−2010	1986−2010	KGE[Q]	0.949	323.540	−1.665	227.737	2.053	NA	NA	NA	NA	0.501	0.899	0.949	0.949
79	Obs_J5412110	RunModel_GR4J	1986−2010	1961−1985	KGE'[Q]	0.949	313.086	−1.791	233.090	2.044	NA	NA	NA	NA	0.512	0.874	0.761	0.834
80	Obs_J5412110	RunModel_GR4J	1986−2010	1986−2010	KGE'[Q]	0.949	313.086	−1.791	233.090	2.044	NA	NA	NA	NA	0.502	0.899	0.949	0.949
81	Obs_J5412110	RunModel_GR5J	1961−1985	1961−1985	RMSE[Q]	0.422	347.373	0.094	174.107	1.623	0.084	NA	NA	NA	0.422	0.915	0.931	0.929
82	Obs_J5412110	RunModel_GR5J	1961−1985	1986−2010	RMSE[Q]	0.422	347.373	0.094	174.107	1.623	0.084	NA	NA	NA	0.623	0.844	0.777	0.805
83	Obs_J5412110	RunModel_GR5J	1961−1985	1961−1985	NSE[Q]	0.915	347.373	0.094	174.107	1.623	0.084	NA	NA	NA	0.422	0.915	0.931	0.929
84	Obs_J5412110	RunModel_GR5J	1961−1985	1986−2010	NSE[Q]	0.915	347.373	0.094	174.107	1.623	0.084	NA	NA	NA	0.623	0.844	0.777	0.805
85	Obs_J5412110	RunModel_GR5J	1961−1985	1961−1985	KGE[Q]	0.954	386.926	0.649	143.148	1.610	0.410	NA	NA	NA	0.435	0.909	0.954	0.954
86	Obs_J5412110	RunModel_GR5J	1961−1985	1986−2010	KGE[Q]	0.954	386.926	0.649	143.148	1.610	0.410	NA	NA	NA	0.678	0.815	0.738	0.800
87	Obs_J5412110	RunModel_GR5J	1961−1985	1961−1985	KGE'[Q]	0.955	367.308	0.557	150.312	1.613	0.402	NA	NA	NA	0.433	0.910	0.955	0.955
88	Obs_J5412110	RunModel_GR5J	1961−1985	1986−2010	KGE'[Q]	0.955	367.308	0.557	150.312	1.613	0.402	NA	NA	NA	0.674	0.818	0.741	0.804
89	Obs_J5412110	RunModel_GR5J	1986−2010	1961−1985	RMSE[Q]	0.486	357.964	−1.189	242.615	1.555	0.359	NA	NA	NA	0.539	0.860	0.729	0.821
90	Obs_J5412110	RunModel_GR5J	1986−2010	1986−2010	RMSE[Q]	0.486	357.964	−1.189	242.615	1.555	0.359	NA	NA	NA	0.486	0.905	0.926	0.914
91	Obs_J5412110	RunModel_GR5J	1986−2010	1961−1985	NSE[Q]	0.905	357.964	−1.189	242.615	1.555	0.359	NA	NA	NA	0.539	0.860	0.729	0.821
92	Obs_J5412110	RunModel_GR5J	1986−2010	1986−2010	NSE[Q]	0.905	357.964	−1.189	242.615	1.555	0.359	NA	NA	NA	0.486	0.905	0.926	0.914
93	Obs_J5412110	RunModel_GR5J	1986−2010	1961−1985	KGE[Q]	0.950	383.814	−0.472	203.124	1.567	0.234	NA	NA	NA	0.526	0.867	0.752	0.828
94	Obs_J5412110	RunModel_GR5J	1986−2010	1986−2010	KGE[Q]	0.950	383.814	−0.472	203.124	1.567	0.234	NA	NA	NA	0.498	0.901	0.950	0.950
95	Obs_J5412110	RunModel_GR5J	1986−2010	1961−1985	KGE'[Q]	0.950	369.968	−0.654	200.937	1.577	0.288	NA	NA	NA	0.522	0.869	0.751	0.829
96	Obs_J5412110	RunModel_GR5J	1986−2010	1986−2010	KGE'[Q]	0.950	369.968	−0.654	200.937	1.577	0.288	NA	NA	NA	0.498	0.900	0.950	0.950
97	Obs_J5412110	RunModel_GR6J	1961−1985	1961−1985	RMSE[Q]	0.416	278.183	0.089	82.264	2.081	0.311	17.491	NA	NA	0.416	0.917	0.931	0.928
98	Obs_J5412110	RunModel_GR6J	1961−1985	1986−2010	RMSE[Q]	0.416	278.183	0.089	82.264	2.081	0.311	17.491	NA	NA	0.627	0.842	0.776	0.807
99	Obs_J5412110	RunModel_GR6J	1961−1985	1961−1985	NSE[Q]	0.917	278.183	0.089	82.264	2.081	0.311	17.491	NA	NA	0.416	0.917	0.931	0.928
100	Obs_J5412110	RunModel_GR6J	1961−1985	1986−2010	NSE[Q]	0.917	278.183	0.089	82.264	2.081	0.311	17.491	NA	NA	0.627	0.842	0.776	0.807
101	Obs_J5412110	RunModel_GR6J	1961−1985	1961−1985	KGE[Q]	0.956	270.449	0.299	209.848	2.211	0.362	4.182	NA	NA	0.430	0.911	0.956	0.956
102	Obs_J5412110	RunModel_GR6J	1961−1985	1986−2010	KGE[Q]	0.956	270.449	0.299	209.848	2.211	0.362	4.182	NA	NA	0.674	0.818	0.746	0.810
103	Obs_J5412110	RunModel_GR6J	1961−1985	1961−1985	KGE'[Q]	0.955	304.133	0.321	173.839	2.182	0.364	4.283	NA	NA	0.430	0.911	0.955	0.955
104	Obs_J5412110	RunModel_GR6J	1961−1985	1986−2010	KGE'[Q]	0.955	304.133	0.321	173.839	2.182	0.364	4.283	NA	NA	0.677	0.816	0.744	0.809
105	Obs_J5412110	RunModel_GR6J	1986−2010	1961−1985	RMSE[Q]	0.486	229.939	−0.532	141.564	1.831	0.288	26.931	NA	NA	0.539	0.861	0.727	0.820
106	Obs_J5412110	RunModel_GR6J	1986−2010	1986−2010	RMSE[Q]	0.486	229.939	−0.532	141.564	1.831	0.288	26.931	NA	NA	0.486	0.905	0.924	0.912
107	Obs_J5412110	RunModel_GR6J	1986−2010	1961−1985	NSE[Q]	0.905	229.939	−0.532	141.564	1.831	0.288	26.931	NA	NA	0.539	0.861	0.727	0.820
108	Obs_J5412110	RunModel_GR6J	1986−2010	1986−2010	NSE[Q]	0.905	229.939	−0.532	141.564	1.831	0.288	26.931	NA	NA	0.486	0.905	0.924	0.912
109	Obs_J5412110	RunModel_GR6J	1986−2010	1961−1985	KGE[Q]	0.949	218.426	−0.521	144.629	1.828	0.258	19.610	NA	NA	0.518	0.871	0.755	0.835
110	Obs_J5412110	RunModel_GR6J	1986−2010	1986−2010	KGE[Q]	0.949	218.426	−0.521	144.629	1.828	0.258	19.610	NA	NA	0.501	0.899	0.949	0.949
111	Obs_J5412110	RunModel_GR6J	1986−2010	1961−1985	KGE'[Q]	0.949	196.370	−0.510	165.670	1.798	0.226	20.086	NA	NA	0.524	0.868	0.753	0.833
112	Obs_J5412110	RunModel_GR6J	1986−2010	1986−2010	KGE'[Q]	0.949	196.370	−0.510	165.670	1.798	0.226	20.086	NA	NA	0.501	0.899	0.949	0.949
113	Obs_J5412110	RunModel_CemaNeigeGR4J	1961−1985	1961−1985	RMSE[Q]	0.414	340.359	0.422	194.416	1.954	NA	NA	0.004	10.609	0.414	0.918	0.940	0.943
114	Obs_J5412110	RunModel_CemaNeigeGR4J	1961−1985	1986−2010	RMSE[Q]	0.414	340.359	0.422	194.416	1.954	NA	NA	0.004	10.609	0.630	0.841	0.773	0.814
115	Obs_J5412110	RunModel_CemaNeigeGR4J	1961−1985	1961−1985	NSE[Q]	0.918	340.359	0.422	194.416	1.954	NA	NA	0.004	10.609	0.414	0.918	0.940	0.943
116	Obs_J5412110	RunModel_CemaNeigeGR4J	1961−1985	1986−2010	NSE[Q]	0.918	340.359	0.422	194.416	1.954	NA	NA	0.004	10.609	0.630	0.841	0.773	0.814
117	Obs_J5412110	RunModel_CemaNeigeGR4J	1961−1985	1961−1985	KGE[Q]	0.956	337.827	0.440	157.788	2.051	NA	NA	0.357	12.427	0.425	0.913	0.956	0.956
118	Obs_J5412110	RunModel_CemaNeigeGR4J	1961−1985	1986−2010	KGE[Q]	0.956	337.827	0.440	157.788	2.051	NA	NA	0.357	12.427	0.690	0.809	0.739	0.806
119	Obs_J5412110	RunModel_CemaNeigeGR4J	1961−1985	1961−1985	KGE'[Q]	0.957	305.970	0.348	180.911	2.027	NA	NA	0.002	10.140	0.423	0.914	0.957	0.957
120	Obs_J5412110	RunModel_CemaNeigeGR4J	1961−1985	1986−2010	KGE'[Q]	0.957	305.970	0.348	180.911	2.027	NA	NA	0.002	10.140	0.678	0.815	0.746	0.814
121	Obs_J5412110	RunModel_CemaNeigeGR4J	1986−2010	1961−1985	RMSE[Q]	0.487	322.494	−1.973	284.529	1.780	NA	NA	0.063	7.561	0.538	0.861	0.737	0.826
122	Obs_J5412110	RunModel_CemaNeigeGR4J	1986−2010	1986−2010	RMSE[Q]	0.487	322.494	−1.973	284.529	1.780	NA	NA	0.063	7.561	0.487	0.905	0.936	0.935
123	Obs_J5412110	RunModel_CemaNeigeGR4J	1986−2010	1961−1985	NSE[Q]	0.905	322.494	−1.973	284.529	1.780	NA	NA	0.063	7.561	0.538	0.861	0.737	0.826
124	Obs_J5412110	RunModel_CemaNeigeGR4J	1986−2010	1986−2010	NSE[Q]	0.905	322.494	−1.973	284.529	1.780	NA	NA	0.063	7.561	0.487	0.905	0.936	0.935
125	Obs_J5412110	RunModel_CemaNeigeGR4J	1986−2010	1961−1985	KGE[Q]	0.949	337.942	−1.547	220.161	2.054	NA	NA	0.004	9.288	0.507	0.877	0.762	0.836
126	Obs_J5412110	RunModel_CemaNeigeGR4J	1986−2010	1986−2010	KGE[Q]	0.949	337.942	−1.547	220.161	2.054	NA	NA	0.004	9.288	0.499	0.900	0.949	0.949
127	Obs_J5412110	RunModel_CemaNeigeGR4J	1986−2010	1961−1985	KGE'[Q]	0.950	297.865	−1.935	247.735	2.027	NA	NA	0.070	8.928	0.508	0.876	0.762	0.834
128	Obs_J5412110	RunModel_CemaNeigeGR4J	1986−2010	1986−2010	KGE'[Q]	0.950	297.865	−1.935	247.735	2.027	NA	NA	0.070	8.928	0.499	0.900	0.950	0.950

	Basin	Model	Per_Cal	Per_Val	OF_Name	OF_Value	X1	X2	X3	X4	X5	X6	C1	C2	RMSE	NSE	KGE	KGE2
129	Obs_200105	RunModel_GR4J	1978−1993	1978−1993	RMSE[Q]	5.665	11257.721	1.175	19.041	1.666	NA	NA	NA	NA	5.665	0.115	0.072	0.072
130	Obs_200105	RunModel_GR4J	1978−1993	1994−2008	RMSE[Q]	5.665	11257.721	1.175	19.041	1.666	NA	NA	NA	NA	4.197	0.186	0.194	0.187
131	Obs_200105	RunModel_GR4J	1978−1993	1978−1993	NSE[Q]	0.115	11257.721	1.175	19.041	1.666	NA	NA	NA	NA	5.665	0.115	0.072	0.072
132	Obs_200105	RunModel_GR4J	1978−1993	1994−2008	NSE[Q]	0.115	11257.721	1.175	19.041	1.666	NA	NA	NA	NA	4.197	0.186	0.194	0.187
133	Obs_200105	RunModel_GR4J	1978−1993	1978−1993	KGE[Q]	0.280	451.551	1.016	7.126	1.999	NA	NA	NA	NA	7.115	−0.396	0.280	0.280
134	Obs_200105	RunModel_GR4J	1978−1993	1994−2008	KGE[Q]	0.280	451.551	1.016	7.126	1.999	NA	NA	NA	NA	5.801	−0.556	0.337	0.345
135	Obs_200105	RunModel_GR4J	1978−1993	1978−1993	KGE'[Q]	0.281	519.081	0.889	7.086	1.999	NA	NA	NA	NA	6.997	−0.350	0.279	0.281
136	Obs_200105	RunModel_GR4J	1978−1993	1994−2008	KGE'[Q]	0.281	519.081	0.889	7.086	1.999	NA	NA	NA	NA	5.663	−0.483	0.351	0.347
137	Obs_200105	RunModel_GR4J	1994−2008	1978−1993	RMSE[Q]	4.181	7631.197	1.621	45.150	1.378	NA	NA	NA	NA	5.680	0.110	0.077	0.081
138	Obs_200105	RunModel_GR4J	1994−2008	1994−2008	RMSE[Q]	4.181	7631.197	1.621	45.150	1.378	NA	NA	NA	NA	4.181	0.192	0.205	0.205
139	Obs_200105	RunModel_GR4J	1994−2008	1978−1993	NSE[Q]	0.192	7631.197	1.621	45.150	1.378	NA	NA	NA	NA	5.680	0.110	0.077	0.081
140	Obs_200105	RunModel_GR4J	1994−2008	1994−2008	NSE[Q]	0.192	7631.197	1.621	45.150	1.378	NA	NA	NA	NA	4.181	0.192	0.205	0.205
141	Obs_200105	RunModel_GR4J	1994−2008	1978−1993	KGE[Q]	0.387	1144.829	1.042	11.469	1.814	NA	NA	NA	NA	6.381	−0.123	0.255	0.257
142	Obs_200105	RunModel_GR4J	1994−2008	1994−2008	KGE[Q]	0.387	1144.829	1.042	11.469	1.814	NA	NA	NA	NA	4.960	−0.137	0.387	0.386
143	Obs_200105	RunModel_GR4J	1994−2008	1978−1993	KGE'[Q]	0.390	1373.265	0.768	10.937	1.825	NA	NA	NA	NA	6.267	−0.083	0.240	0.259
144	Obs_200105	RunModel_GR4J	1994−2008	1994−2008	KGE'[Q]	0.390	1373.265	0.768	10.937	1.825	NA	NA	NA	NA	4.827	−0.077	0.381	0.390
145	Obs_200105	RunModel_GR5J	1978−1993	1978−1993	RMSE[Q]	5.653	11653.227	6.695	20.773	1.192	0.756	NA	NA	NA	5.653	0.119	0.096	0.095
146	Obs_200105	RunModel_GR5J	1978−1993	1994−2008	RMSE[Q]	5.653	11653.227	6.695	20.773	1.192	0.756	NA	NA	NA	4.201	0.184	0.212	0.206
147	Obs_200105	RunModel_GR5J	1978−1993	1978−1993	NSE[Q]	0.119	11653.227	6.695	20.773	1.192	0.756	NA	NA	NA	5.653	0.119	0.096	0.095
148	Obs_200105	RunModel_GR5J	1978−1993	1994−2008	NSE[Q]	0.119	11653.227	6.695	20.773	1.192	0.756	NA	NA	NA	4.201	0.184	0.212	0.206
149	Obs_200105	RunModel_GR5J	1978−1993	1978−1993	KGE[Q]	0.290	707.253	7.460	15.566	1.212	0.749	NA	NA	NA	6.836	−0.289	0.290	0.286
150	Obs_200105	RunModel_GR5J	1978−1993	1994−2008	KGE[Q]	0.290	707.253	7.460	15.566	1.212	0.749	NA	NA	NA	5.524	−0.411	0.361	0.368
151	Obs_200105	RunModel_GR5J	1978−1993	1978−1993	KGE'[Q]	0.292	912.079	6.796	13.456	1.227	0.788	NA	NA	NA	6.613	−0.206	0.282	0.292
152	Obs_200105	RunModel_GR5J	1978−1993	1994−2008	KGE'[Q]	0.292	912.079	6.796	13.456	1.227	0.788	NA	NA	NA	5.269	−0.284	0.378	0.373
153	Obs_200105	RunModel_GR5J	1994−2008	1978−1993	RMSE[Q]	4.179	7331.974	2.507	44.256	1.095	0.509	NA	NA	NA	5.678	0.111	0.076	0.080
154	Obs_200105	RunModel_GR5J	1994−2008	1994−2008	RMSE[Q]	4.179	7331.974	2.507	44.256	1.095	0.509	NA	NA	NA	4.179	0.192	0.204	0.204
155	Obs_200105	RunModel_GR5J	1994−2008	1978−1993	NSE[Q]	0.192	7331.974	2.507	44.256	1.095	0.509	NA	NA	NA	5.678	0.111	0.076	0.080
156	Obs_200105	RunModel_GR5J	1994−2008	1994−2008	NSE[Q]	0.192	7331.974	2.507	44.256	1.095	0.509	NA	NA	NA	4.179	0.192	0.204	0.204
157	Obs_200105	RunModel_GR5J	1994−2008	1978−1993	KGE[Q]	0.396	1592.373	8.478	17.583	1.171	0.777	NA	NA	NA	6.264	−0.082	0.268	0.267
158	Obs_200105	RunModel_GR5J	1994−2008	1994−2008	KGE[Q]	0.396	1592.373	8.478	17.583	1.171	0.777	NA	NA	NA	4.872	−0.097	0.396	0.395
159	Obs_200105	RunModel_GR5J	1994−2008	1978−1993	KGE'[Q]	0.400	1907.918	8.099	17.066	1.169	0.798	NA	NA	NA	6.155	−0.045	0.255	0.273
160	Obs_200105	RunModel_GR5J	1994−2008	1994−2008	KGE'[Q]	0.400	1907.918	8.099	17.066	1.169	0.798	NA	NA	NA	4.753	−0.044	0.389	0.400
161	Obs_200105	RunModel_GR6J	1978−1993	1978−1993	RMSE[Q]	5.768	0.003	0.521	10017.487	1.305	−0.271	1289.996	NA	NA	5.768	0.082	−0.068	−0.068
162	Obs_200105	RunModel_GR6J	1978−1993	1994−2008	RMSE[Q]	5.768	0.003	0.521	10017.487	1.305	−0.271	1289.996	NA	NA	4.327	0.134	0.024	0.019
163	Obs_200105	RunModel_GR6J	1978−1993	1978−1993	NSE[Q]	0.082	0.003	0.521	10017.487	1.305	−0.271	1289.996	NA	NA	5.768	0.082	−0.068	−0.068
164	Obs_200105	RunModel_GR6J	1978−1993	1994−2008	NSE[Q]	0.082	0.003	0.521	10017.487	1.305	−0.271	1289.996	NA	NA	4.327	0.134	0.024	0.019
165	Obs_200105	RunModel_GR6J	1978−1993	1978−1993	KGE[Q]	0.356	0.000	4.627	8.428	1.583	0.710	42.475	NA	NA	6.989	−0.347	0.356	0.357
166	Obs_200105	RunModel_GR6J	1978−1993	1994−2008	KGE[Q]	0.356	0.000	4.627	8.428	1.583	0.710	42.475	NA	NA	5.884	−0.601	0.348	0.371
167	Obs_200105	RunModel_GR6J	1978−1993	1978−1993	KGE'[Q]	0.355	0.000	3.898	15.472	1.574	0.663	32.571	NA	NA	6.962	−0.337	0.354	0.355
168	Obs_200105	RunModel_GR6J	1978−1993	1994−2008	KGE'[Q]	0.355	0.000	3.898	15.472	1.574	0.663	32.571	NA	NA	5.816	−0.564	0.359	0.381
169	Obs_200105	RunModel_GR6J	1994−2008	1978−1993	RMSE[Q]	4.225	0.001	0.375	467.002	1.149	−0.180	525.501	NA	NA	5.763	0.084	0.054	0.061
170	Obs_200105	RunModel_GR6J	1994−2008	1994−2008	RMSE[Q]	4.225	0.001	0.375	467.002	1.149	−0.180	525.501	NA	NA	4.225	0.175	0.191	0.191
171	Obs_200105	RunModel_GR6J	1994−2008	1978−1993	NSE[Q]	0.175	0.001	0.375	467.002	1.149	−0.180	525.501	NA	NA	5.763	0.084	0.054	0.061
172	Obs_200105	RunModel_GR6J	1994−2008	1994−2008	NSE[Q]	0.175	0.001	0.375	467.002	1.149	−0.180	525.501	NA	NA	4.225	0.175	0.191	0.191
173	Obs_200105	RunModel_GR6J	1994−2008	1978−1993	KGE[Q]	0.431	0.009	0.521	39.646	1.427	0.234	54.598	NA	NA	6.364	−0.117	0.294	0.300
174	Obs_200105	RunModel_GR6J	1994−2008	1994−2008	KGE[Q]	0.431	0.009	0.521	39.646	1.427	0.234	54.598	NA	NA	4.941	−0.129	0.431	0.431
175	Obs_200105	RunModel_GR6J	1994−2008	1978−1993	KGE'[Q]	0.431	0.003	0.534	40.904	1.420	0.266	57.158	NA	NA	6.346	−0.111	0.291	0.299
176	Obs_200105	RunModel_GR6J	1994−2008	1994−2008	KGE'[Q]	0.431	0.003	0.534	40.904	1.420	0.266	57.158	NA	NA	4.917	−0.118	0.431	0.431
177	Obs_200105	RunModel_CemaNeigeGR4J	1978−1993	1978−1993	RMSE[Q]	2.845	510.232	2.683	70.441	1.325	NA	NA	0.003	4.076	2.845	0.777	0.827	0.822
178	Obs_200105	RunModel_CemaNeigeGR4J	1978−1993	1994−2008	RMSE[Q]	2.845	510.232	2.683	70.441	1.325	NA	NA	0.003	4.076	2.166	0.783	0.889	0.893
179	Obs_200105	RunModel_CemaNeigeGR4J	1978−1993	1978−1993	NSE[Q]	0.777	510.232	2.683	70.441	1.325	NA	NA	0.003	4.076	2.845	0.777	0.827	0.822
180	Obs_200105	RunModel_CemaNeigeGR4J	1978−1993	1994−2008	NSE[Q]	0.777	510.232	2.683	70.441	1.325	NA	NA	0.003	4.076	2.166	0.783	0.889	0.893
181	Obs_200105	RunModel_CemaNeigeGR4J	1978−1993	1978−1993	KGE[Q]	0.878	278.063	2.448	57.002	1.359	NA	NA	0.783	3.848	2.971	0.757	0.878	0.877
182	Obs_200105	RunModel_CemaNeigeGR4J	1978−1993	1994−2008	KGE[Q]	0.878	278.063	2.448	57.002	1.359	NA	NA	0.783	3.848	2.565	0.696	0.773	0.792
183	Obs_200105	RunModel_CemaNeigeGR4J	1978−1993	1978−1993	KGE'[Q]	0.878	239.455	2.480	60.841	1.351	NA	NA	0.686	3.713	2.960	0.758	0.878	0.878
184	Obs_200105	RunModel_CemaNeigeGR4J	1978−1993	1994−2008	KGE'[Q]	0.878	239.455	2.480	60.841	1.351	NA	NA	0.686	3.713	2.559	0.697	0.776	0.788
185	Obs_200105	RunModel_CemaNeigeGR4J	1994−2008	1978−1993	RMSE[Q]	2.034	838.194	3.006	94.962	1.296	NA	NA	0.142	4.831	2.945	0.761	0.742	0.743
186	Obs_200105	RunModel_CemaNeigeGR4J	1994−2008	1994−2008	RMSE[Q]	2.034	838.194	3.006	94.962	1.296	NA	NA	0.142	4.831	2.034	0.809	0.853	0.846
187	Obs_200105	RunModel_CemaNeigeGR4J	1994−2008	1978−1993	NSE[Q]	0.809	838.194	3.006	94.962	1.296	NA	NA	0.142	4.831	2.945	0.761	0.742	0.743
188	Obs_200105	RunModel_CemaNeigeGR4J	1994−2008	1994−2008	NSE[Q]	0.809	838.194	3.006	94.962	1.296	NA	NA	0.142	4.831	2.034	0.809	0.853	0.846
189	Obs_200105	RunModel_CemaNeigeGR4J	1994−2008	1978−1993	KGE[Q]	0.897	548.475	3.180	104.858	1.286	NA	NA	0.673	4.639	2.898	0.768	0.811	0.817
190	Obs_200105	RunModel_CemaNeigeGR4J	1994−2008	1994−2008	KGE[Q]	0.897	548.475	3.180	104.858	1.286	NA	NA	0.673	4.639	2.111	0.794	0.897	0.897
191	Obs_200105	RunModel_CemaNeigeGR4J	1994−2008	1978−1993	KGE'[Q]	0.897	468.717	3.180	105.778	1.289	NA	NA	0.273	4.317	2.877	0.772	0.811	0.819
192	Obs_200105	RunModel_CemaNeigeGR4J	1994−2008	1994−2008	KGE'[Q]	0.897	468.717	3.180	105.778	1.289	NA	NA	0.273	4.317	2.113	0.794	0.897	0.897

Appendix C

Script used for the transformations analysis

```
library(airGR)
library(airGRteaching)

## DATA.FRAME OF OBSERVED DATA
# X0310010_BV
    Input_X0310010 <- read.table(file = "C:/Data/chiara/02_DATA/French_Data/X0310010_BV.txt",
                                 header = TRUE, sep = ";", skip = 50)
    Input_X0310010$Q[Input_X0310010$Q < 0] <- NA; Input_X0310010$Ptot[Input_X0310010$Ptot <
    0] <- NA
    A_X0310010 <- 2282.76
    Input_X0310010$Qmm <- 0.0864*(Input_X0310010$Q)/A_X0310010
    Input_X0310010$Time <- as.POSIXct(as.character(Input_X0310010$Date), format = "%Y%m%d",
    tz = "UTC")
    Obs_X0310010 <- Input_X0310010[, c("Time", "Ptot", "ETP_O", "Qmm", "Temp")]
# J5412110_BV
    Input_J5412110 <- read.table(file = "C:/Data/chiara/02_DATA/French_Data/J5412110_BV.txt",
                                 header = TRUE, sep = ";", skip = 50)
    Input_J5412110$Q[Input_J5412110$Q < 0] <- NA; Input_J5412110$Ptot[Input_J5412110$Ptot <
    0] <- NA
    A_J5412110 <- 675.64
    Input_J5412110$Qmm <- 0.0864*(Input_J5412110$Q)/A_J5412110
    Input_J5412110$Time <- as.POSIXct(as.character(Input_J5412110$Date), format = "%Y%m%d",
    tz = "UTC")
    Obs_J5412110 <- Input_J5412110[, c("Time", "Ptot", "ETP_O", "Qmm", "Temp")]
# 200105
    Input_200105_1 <- read.table(file =
    "C:/Data/chiara/02_DATA/Austrian_Data/BaciniTest/obs_200105.txt",
                                 header = FALSE, sep = "\t")
    Input_200105_2 <- read.table(file =
    "C:/Data/chiara/02_DATA/Austrian_Data/BaciniTest/input_200105.txt",
                                 header = TRUE, sep = "\t")
    #Input_200105_2 <- Input_200105_2[[1]] %in% seq(from = as.POSIXct("19764101-01"), to =
    as.POSIXct("2008-12-31"), by = 1)
    Input_200105_2 <- Input_200105_2[1:12054,]
    Input_200105_3 <- read.table(file =
    "C:/Data/chiara/02_DATA/Austrian_Data/BaciniTest/AreaDist_200105.txt",
                                 header = TRUE, sep = " ")

    Input_200105_1$V2[Input_200105_1$V2 < 0] <- NA
    Input_200105_2[Input_200105_2 < -99] <- NA #Warning
    A_200105 <- 95.50   #Input_200105$Qmm <- 0.0864*(Input_200105$Q)/A_200105
    Input_200105_1$Time <- as.POSIXct(as.character(Input_200105_1[,1]), format = "%Y-%m-%d",
    tz = "UTC")
    T <- apply(Input_200105_2[,21:39], 1, function(x, w) weighted.mean(x, Input_200105_3[,1]))
    P <- apply(Input_200105_2[,2:20], 1, function(x, w) weighted.mean(x, Input_200105_3[,1]))
    EP <- apply(Input_200105_2[,40:58], 1, function(x, w) weighted.mean(x,
    Input_200105_3[,1]))

    Obs_200105 <- data.frame(col1 = Input_200105_1$Time,
                             col2 = P,
                             col3 = EP,
                             col4 = Input_200105_1$V2,
                             col5 = T)
    colnames(Obs_200105) <-  c("Time", "Ptot", "ETP_O", "Qmm", "Temp")


# Input CemaNeige
    Input_Neige_Fr <- read.table(file =
    "C:/Data/chiara/02_DATA/French_Data/_ListeBV_Quantiles_altitude.txt",
               header = TRUE, sep = "")
    Input_Hypso_Au <- read.table(file = "C:/Data/chiara/02_DATA/Austrian_Data/HypsoZ.txt",
                                 header = TRUE, sep = "\t")
    Input_Zin_Au <- read.table(file = "C:/Data/chiara/02_DATA/Austrian_Data/Zin.txt",
                                 header = TRUE, sep = "\t")
    Input_CN <- data.frame(matrix(nrow = 102, ncol = 3))
    rownames(Input_CN) <- c(colnames(Input_Neige_Fr[,5:106]))
    colnames(Input_CN) <- c("X0310010", "J5412110", "200105")
    Input_CN["Zmean", ] <- c(Input_Neige_Fr$Zmean, Input_Zin_Au[1,1])
    Input_CN[2:102, "X0310010"] <- t(Input_Neige_Fr[1, 6:106])
    Input_CN[2:102, "J5412110"] <- t(Input_Neige_Fr[2, 6:106])
    Input_CN[2:102, "200105"] <- t(Input_Hypso_Au["200105", 1:101])
```

```
# Vectors for the loop
    BASIN <- list(Obs_X0310010, Obs_J5412110, Obs_200105)
    BASIN_char <- c("Obs_X0310010", "Obs_J5412110", "Obs_200105")
    MOD <- c(RunModel_GR4J, RunModel_GR5J, RunModel_GR6J, RunModel_CemaNeigeGR4J)
    MOD_char <- c("RunModel_GR4J", "RunModel_GR5J", "RunModel_GR6J", "RunModel_CemaNeigeGR4J")
    CRIT <- c(ErrorCrit_RMSE, ErrorCrit_NSE, ErrorCrit_KGE, ErrorCrit_KGE2)
    CRIT_char <- c("ErrorCrit_RMSE", "ErrorCrit_NSE", "ErrorCrit_KGE", "ErrorCrit_KGE2")

    Ind_Run_AU <- list((seq(which(format(Obs_200105$Time, format = "%Y-%m-%d") ==
    "1978-01-01"),
                  which(format(Obs_200105$Time, format = "%Y-%m-%d") == "1993-12-31"))),
                 (seq(which(format(Obs_200105$Time, format = "%Y-%m-%d") == "1994-01-01"),
                  which(format(Obs_200105$Time, format = "%Y-%m-%d") == "2008-12-31"))))
    Ind_Run_FR <- list((seq(which(format(Obs_X0310010$Time, format = "%Y-%m-%d") ==
    "1961-01-01"),
                        which(format(Obs_X0310010$Time, format = "%Y-%m-%d") ==
                        "1985-12-31"))),
                  (seq(which(format(Obs_X0310010$Time, format = "%Y-%m-%d") ==
                  "1986-01-01"),
                      which(format(Obs_X0310010$Time, format = "%Y-%m-%d") ==
                      "2010-07-31"))))
    Per_Cal_FR <- c("1961-1985", "1986-2010")
    Per_Cal_AU <- c("1978-1993", "1994-2008")

# To store results
result_row_hl <- vector(length = 8)
results_df_hl <- as.data.frame(matrix(ncol = 8))
colnames(results_df_hl) = c("Basin", "Model", "Per_Cal", "OF_Name", "Crit_Name", "Transf",
"Crit_Val", "Deviation")

## LOOPS
for (b in 1:2) {
 x = 4 #for (x in seq_along(MOD)) {
  for (p in seq_along(Per_Cal_FR)){
    for (i in seq_along(CRIT)) {

        # define calibration period
        if (b == 3) {
          Ind_Run_Cal <- Ind_Run_AU[[p]]
          Per_Cal <- Per_Cal_AU[[p]]
          WuP <- 1:730
        } else  {
          Ind_Run_Cal <- Ind_Run_FR[[p]]
          Per_Cal <- Per_Cal_FR[[p]]
          WuP <- 1:(as.numeric(difftime(strptime("1960-12-31", format = "%Y-%m-%d"),
                                        strptime(Obs_X0310010[1, "Time"], format =
                                        "%Y-%m-%d"),units="days")))
        }

        # define inputs model and run options
        InputsModel <- CreateInputsModel(FUN_MOD = MOD[[x]], DatesR = BASIN[[b]]$Time,
                                         Precip = BASIN[[b]]$Ptot, PotEvap = BASIN[[b]]$ETP_O,
                                         TempMean = BASIN[[b]]$Temp, ZInputs =
                                         Input_CN["Zmean", b],
                                         HypsoData = Input_CN[2:102, b], NLayers = 5)

        RunOptions_Cal <- CreateRunOptions(FUN_MOD = MOD[[x]], InputsModel = InputsModel,
                                           IndPeriod_WarmUp = WuP, IndPeriod_Run =
                                           Ind_Run_Cal) #warning solid precipitation
        # define warm period here in CreateRunOptions!!! warmup period if cycle

        ## CALIBRATION
        InputsCrit <- CreateInputsCrit(FUN_CRIT = CRIT[[i]], InputsModel = InputsModel,
                                       RunOptions = RunOptions_Cal, Qobs =
                                       BASIN[[b]]$Qmm[Ind_Run_Cal])

        # "", "sqrt", "log", "inv", "sort"
        CalibOptions <- CreateCalibOptions(FUN_MOD = MOD[[x]], FUN_CALIB =
        Calibration_Michel, FUN_TRANSFO = NULL)
```

```
        OutputsCalib <- Calibration_Michel(InputsModel = InputsModel, RunOptions =
        RunOptions_Cal,
                                           InputsCrit = InputsCrit, CalibOptions =
                                           CalibOptions,
                                           FUN_MOD = MOD[[x]], FUN_CRIT = CRIT[[i]],
                                           FUN_TRANSFO = NULL)

        ## SIMULATION
          for (ii in seq_along(CRIT)) {

            if (CRIT_char[[ii]] %in% c("ErrorCrit_KGE", "ErrorCrit_KGE2")) {
              tr <-  c("", "sqrt", "inv")
            } else {
              tr <-  c("", "sqrt", "log", "inv")
            }

            for (tt in seq_along(tr)) {

              if (b == 3) {
              Ind_Run_Val <- seq(which(format(Obs_X0310010$Time, format = "%Y-%m-%d") ==
              "1976-01-01"),
                                 which(format(Obs_X0310010$Time, format = "%Y-%m-%d") ==
                                 "2008-12-31"))
              } else  {
              Ind_Run_Val <- seq(which(format(Obs_X0310010$Time, format = "%Y-%m-%d") ==
              "1961-01-01"),
                                  which(format(Obs_X0310010$Time, format = "%Y-%m-%d") ==
                                  "2009-06-29"))
              }

              RunOptions_Val <- CreateRunOptions(FUN_MOD = MOD[[x]], InputsModel =
              InputsModel,
                                                 IndPeriod_WarmUp = WuP, IndPeriod_Run =
                                                 Ind_Run_Val)
              Param <- OutputsCalib$`ParamFinalR`
              OutputsModel <- MOD[[x]](InputsModel = InputsModel,
                                       RunOptions = RunOptions_Val, Param = Param)

              q20 <- quantile(BASIN[[b]]$Qmm, probs = 0.2, na.rm = TRUE)
              q80 <- quantile(BASIN[[b]]$Qmm, probs = 0.8, na.rm = TRUE)

              #Low flows
              boolcrit <- BASIN[[b]]$Qmm < q20 & !is.na(BASIN[[b]]$Qmm)
              #High flows
              #boolcrit <- BASIN[[b]]$Qmm > q80 & !is.na(BASIN[[b]]$Qmm)

              InputsCrit <- CreateInputsCrit(FUN_CRIT = CRIT[[ii]], InputsModel = InputsModel,
                                             RunOptions = RunOptions_Val, Qobs =
                                             BASIN[[b]]$Qmm[Ind_Run_Val],
                                             transfo = tr[[tt]], BoolCrit =
                                             boolcrit[Ind_Run_Val])
              OutputsCrit <- CRIT[[ii]](InputsCrit = InputsCrit, OutputsModel = OutputsModel)


              diffe <- abs(OutputsCrit$CritVal - OutputsCrit$CritBestValue)
              result_row_hl <- c(BASIN_char[[b]], MOD_char[[x]], Per_Cal, CRIT_char[[i]],
                                  OutputsCrit$CritName, tr[[tt]], OutputsCrit$CritVal, diffe)
              results_df_hl <- rbind(results_df_hl, result_row_hl)
              results_df_hl <- results_df_hl[!is.na(results_df_hl),]
            }
          }
      }
  }
}

write.csv(results_df_hl, "Criteria_Values_HighFlows")

# RESULTS
results_df_hf <- read.csv(file = "C:/Data/chiara/03_CODES/Criteria_Values_HighFlows")
# OR
results_df_lf <- read.csv(file = "C:/Data/chiara/03_CODES/Criteria_Values_LowFlows")
```

```
tr <-  c("", "sqrt", "log", "inv")
mean_tr <- as.data.frame(matrix(ncol = 5))
colnames(mean_tr) <-  c("Mean.Dev", "Q", "sqrt", "log", "inv")
mean_tr[1:2,1] <- c("High Flows", "Low Flows")

for (tt in seq_along(tr)){
  mean_tr[1, tt+1] <-  mean(as.numeric(results_df_hf[results_df_hf$Transf == tr[[tt]],
  colnames(results_df_hf) == "Deviation"]),
                          na.rm = TRUE)
  mean_tr[2, tt+1] <-  mean(as.numeric(results_df_lf[results_df_lf$Transf == tr[[tt]],
  colnames(results_df_lf) == "Deviation"]),
                         na.rm = TRUE)
}

library(gridExtra)
pdf("mean_tr.pdf", height=1.5, width=4)
grid.table(mean_tr)
dev.off()

# plot(obse, type = "l")
# lines(simu, type = "l", col = "orange")
```

Appendix D

Script used for the *dam module* analysis

```
rm(list = ls())
library(airGR)
library(RColorBrewer)

crit = "KGE"

basin_name <- "FR_A1320310"
name_crit = paste("ErrorCrit_", crit, sep = "")
my_crit = get(paste("ErrorCrit_", crit, sep = ""))
start_date <- "1958-08-01"
end_date <- "2016-07-31"

BasinObs <- read.table(file =
"C:/Data/chiara/03_CODES/Dam_analysis/donnees_Morgane/FR_A1320310.txt",
 sep = ";", header = TRUE)
BasinObs <- as.data.frame(BasinObs)
names(BasinObs) = c("Date", "Q", "Qmm", "P", "E", "V", "Vmm", "DVmm",
"CodeQ")
BasinObs$Qmm[BasinObs$Qmm < 0] <- NA
BasinObs$P[BasinObs$P < 0] <- NA
BasinObs$E[BasinObs$E < 0] <- NA
BasinObs$Date <- as.POSIXct(BasinObs$Date, format = "%Y-%m-%d", tz
="UTC")
BasinObs <- BasinObs[-c(33970, 34702, 35068, 35434, 36166, 36532,
36898), ]
#BasinObs <- BasinObs[!is.na(BasinObs$P) & !is.na(BasinObs$E), ]
BasinObs <- BasinObs[28856:31332,]

jourJulien <- paste(substr(BasinObs$Date, 6, 7), substr(BasinObs$Date,
9, 10), sep = "")
DVmm_interannuel <- aggregate(BasinObs$DVmm, by = list(jourJulien),
FUN = mean, na.rm = TRUE)$x
Vmm_interannuel <- aggregate(BasinObs$Vmm, by = list(jourJulien), FUN
= mean, na.rm = TRUE)$x
P_interannuel <- aggregate(BasinObs$P, by = list(jourJulien), FUN =
mean, na.rm = TRUE)$x
E_interannuel <- aggregate(BasinObs$E, by = list(jourJulien), FUN =
mean, na.rm = TRUE)$x
Qobs_interannuel <- aggregate(BasinObs$Qmm, by = list(jourJulien), FUN
= mean, na.rm = TRUE)$x
month <- paste(substr(aggregate(BasinObs$P, by = list(jourJulien), FUN
= mean, na.rm = TRUE)$Group.1, 1, 2), sep = "")
P_month  <- aggregate(P_interannuel, by = list(month), FUN = mean,
na.rm = TRUE)$x

dates1 <- seq(which(format(BasinObs$Date, format = "%Y-%m-%d")=="1994-
01-01"),
                which(format(BasinObs$Date, format = "%Y-%m-
%d")=="1996-12-31"))
dates2 <- seq(which(format(BasinObs$Date, format = "%Y-%m-%d")=="1997-
01-01"),
                which(format(BasinObs$Date, format = "%Y-%m-
%d")=="1998-12-31"))
dates3 <- seq(which(format(BasinObs$Date, format = "%Y-%m-%d")=="1999-
```

```
01-01"),
                which(format(BasinObs$Date, format = "%Y-%m-
%d")=="2000-10-12"))
dates <- c(dates1, dates2, dates3)


pdf("plots_V&Qobs.pdf")

##### PLOT TOT ######
par(oma = c(1,1,1,3), xpd=TRUE)
ylim = c(min(BasinObs$Qmm, BasinObs$DVmm, na.rm = TRUE)-3,
         max(BasinObs$Qmm, BasinObs$DVmm, na.rm = TRUE))
plot(BasinObs$Date, BasinObs$Qmm,
     type = "l", ylim = ylim,
     main = paste("Observed discharges and volume variation \n in
reservoir of basin", basin_name),
     ylab = "Qobs [mm/day]", xlab = "Time [years]")
par(new = TRUE)
plot(BasinObs$Date, BasinObs$DVmm,
     type = "l", col = "blue", ylim = ylim*c(0.2, 0.2), xaxt = "n",
yaxt = "n", ylab = "", xlab = "")
axis(side = 4, col.axis = "blue")
mtext(paste("Delta_V [mm/day]"), side = 4, line = 3, col = "blue")
legend("top", legend = c("Qobs", "Delta_V"),
       col = c("black", "blue"), lty = c(1, 1), lwd = c(2, 2), cex =
0.8)

##### PLOT period 1 ######
par(oma = c(1,1,1,3), xpd=TRUE)
ylim = c(min(BasinObs$Qmm[dates1], BasinObs$DVmm[dates1], na.rm =
TRUE)-2.5,
         max(BasinObs$Qmm[dates1], BasinObs$DVmm[dates1], na.rm =
TRUE))
plot(BasinObs$Date[dates1], BasinObs$Qmm[dates1],
     type = "l", ylim = (ylim*c(1.2, 1.2))-c(1.2, 1.2),
     main = paste("Observed discharges and volume variation \n in
reservoir of basin", basin_name, "\n(1994 - 1996)"),
     ylab = "Q [mm/day]", xlab = "Time [years]")
par(new = TRUE)
plot(BasinObs$Date[dates1], BasinObs$DVmm[dates1],
     type = "l", col = "blue", ylim = ylim*c(0.2, 0.2), xaxt = "n",
yaxt = "n", ylab = "", xlab = "")
axis(side = 4, col.axis = "blue")
mtext(paste("Delta_V [mm/day]"), side = 4, line = 3, col = "blue")
legend("topright", legend = c("Qobs", "Delta_V"),
       col = c("black", "blue"), lty = c(1, 1), lwd = c(2, 2), cex =
0.8)


##### PLOT period 2 ######
par(oma = c(7,1,7,3), xpd=TRUE)
ylim = c(min(BasinObs$Qmm[dates2], BasinObs$DVmm[dates2], na.rm =
TRUE)-2.5,
         max(BasinObs$Qmm[dates2], BasinObs$DVmm[dates2], na.rm =
TRUE))
```

```
plot(BasinObs$Date[dates2], BasinObs$Qmm[dates2],
     type = "l", ylim = (ylim*c(1.2, 1.2)),
     ylab = "Q [mm/day]", xlab = "Time [years]")
mtext(paste("Observed discharges and volume variation \n in reservoir
of basin", basin_name, "\n(1997 - 1998)"),
      side = 3, line = 1, font = 2, cex = 0.9)
par(new = TRUE)
plot(BasinObs$Date[dates2], BasinObs$DVmm[dates2],
     type = "l", col = "blue", ylim = ylim*c(0.2, 0.2), xaxt = "n",
yaxt = "n", ylab = "", xlab = "")
axis(side = 4, col.axis = "blue")
mtext(paste("Delta_V [mm/day]"), side = 4, line = 3, col = "blue")
legend("top", legend = c("Qobs", "Delta_V"),
       col = c("black", "blue"), lty = c(1, 1), lwd = c(2, 2), cex =
0.7)

##### PLOT period 2 ZOOM ######
par(oma = c(7,1,7,3), xpd=TRUE)
dates2_zoom <- seq(which(format(BasinObs$Date, format = "%Y-%m-
%d")=="1998-09-01"),
              which(format(BasinObs$Date, format = "%Y-%m-%d")=="1998-
11-30"))
ylim = c(min(BasinObs$Qmm[dates2_zoom], BasinObs$DVmm[dates2_zoom],
na.rm = TRUE)-2.5,
         max(BasinObs$Qmm[dates2_zoom], BasinObs$DVmm[dates2_zoom],
na.rm = TRUE))
plot(BasinObs$Date[dates2_zoom], BasinObs$Qmm[dates2_zoom],
     type = "l", ylim = (ylim*c(1.2, 1.2)),
     ylab = "Q [mm/day]", xlab = "Time [years]")
mtext(paste("Observed discharges and volume variation \n in reservoir
of basin", basin_name, "\n(1998)"),
      side = 3, line = 1, font = 2, cex = 0.9)
par(new = TRUE)
plot(BasinObs$Date[dates2_zoom], BasinObs$DVmm[dates2_zoom],
     type = "l", col = "blue", ylim = ylim*c(0.2, 0.2), xaxt = "n",
yaxt = "n", ylab = "", xlab = "")
axis(side = 4, col.axis = "blue")
mtext(paste("Delta_V [mm/day]"), side = 4, line = 3, col = "blue")
legend("topleft", legend = c("Qobs", "Delta_V"),
       col = c("black", "blue"), lty = c(1, 1), lwd = c(2, 2), cex =
0.8)

##### PLOT period 3 ######
par(oma = c(1,1,1,3), xpd=TRUE)
ylim = c(min(BasinObs$Qmm[dates3], BasinObs$DVmm[dates3], na.rm =
TRUE)-2.5,
         max(BasinObs$Qmm[dates3], BasinObs$DVmm[dates3], na.rm =
TRUE))
plot(BasinObs$Date[dates3], BasinObs$Qmm[dates3],
     type = "l", ylim = (ylim*c(1.2, 1.2))-c(1.2, 1.2),
     main = paste("Observed discharges and volume variation \n in
reservoir of basin", basin_name, "\n(1999 - 2000)"),
     ylab = "Q [mm/day]", xlab = "Time [years]")
par(new = TRUE)
plot(BasinObs$Date[dates3], BasinObs$DVmm[dates3],
```

```
     type = "l", col = "blue", ylim = ylim*c(0.2, 0.2), xaxt = "n",
yaxt = "n", ylab = "", xlab = "")
axis(side = 4, col.axis = "blue")
mtext(paste("Delta_V [mm/day]"), side = 4, line = 3, col = "blue")
legend("topright", legend = c("Qobs", "Delta_V"),
       col = c("black", "blue"), lty = c(1, 1), lwd = c(2, 2), cex =
0.8)

##### PLOT V_interann and Q_obs interann ######
par(oma = c(0,2,0,2), xpd=TRUE)
w1 <- matrix(c(1,2), ncol = 1)
layout(w1, widths = 7, heights = c(5, 4), respect = TRUE)
# par(mfrow = c(2, 1))
plot(DVmm_interannuel,
     type = "n", col = "blue", xaxt = "n", yaxt = "n", xlab = "", ylab
= "",
     main = paste(c("Interannual variation of volume \n in the
reservoir")))
segments(-15,0,length(DVmm_interannuel)+15,0, col = "grey")
par(new = TRUE)
plot(DVmm_interannuel,
     type = "l", col = "blue", xaxt = "n", yaxt = "n", xlab = "", ylab
= "")
mtext(paste("Delta_V \n [mm/day]"), side = 2, line = 3, col = "black")
axis(1, labels = c(month.abb), at = (0:11)*30)
axis(2, labels = c(min(DVmm_interannuel),"0",
round(max(DVmm_interannuel), digits = 2)),
     at = c(min(DVmm_interannuel), 0, max(DVmm_interannuel)))
ylim = c(min(Vmm_interannuel,
             Qobs_interannuel, na.rm = TRUE),
         max(Vmm_interannuel,
             Qobs_interannuel, na.rm = TRUE))
plot(Vmm_interannuel,
     type = "l", ylim = ylim, col = "red", xaxt = "n", xlab ="",
     ylab = "", main = paste(c("Interannual observed discharge and
volume \n in the reservoir")))
mtext(paste("V and Qobs \n [mm/day]"), side = 2, line = 3, col =
"black")
axis(1, labels = c(month.abb), at = (0:11)*30)
par(new = TRUE)
plot(Qobs_interannuel,
     type = "l", col = "black", ylim = ylim, xaxt = "n", yaxt = "n",
xlab = "", ylab = "")
legend("topright", legend = c("V_interann", "Qobs_interann"),
       col = c("red", "black"), lty = c(1, 1), lwd = c(2, 2), cex =
0.6)

##### PLOT V_interann and Q_obs interann ######
par(mfrow = c(2, 1))
plot(P_interannuel, type = "l",
     col = "navy", xaxt = "n", xlab = "", ylab = "P [mm/day]",
     main = "Interannual rainfall")
axis(1, labels = c(month.abb), at = (1:12)*30)
plot(E_interannuel,
     type = "l", col = "navy", xaxt = "n", xlab = "", ylab = "EP
```

```
[mm/day]",
     main = "Interannual evapotranspiration")
axis(1, labels = c(month.abb), at = (1:12)*30)
dev.off()
```

Bibliography

Equipe Hydrologie (Catchment hydrology research group) Irstea webpage. `https://webgr.irstea.fr/en/`.

Wikipedia. `https://en.wikipedia.org`.

V. Andréassian, N. Le Moine, C. Perrin, M.H. Ramos, T. Mathevet L. Oudin, J. Lerat, and L. Berthet. All that glitter is not gold: the case of calibrating hydrological models. *Hydrological processes*, 26:2206–2210, 2012.

L. Coron, V. Andréassian, M. Bourqui, C. Perrin, and F. Hendrickx. Pathologies of hydrological model used in changinc climatic conditions: a review. *Hydro-climatology: Variability and Change*, 344, 2011.

L. Coron, G. Thirel, O. Delaigue, C. Perrin, and V. Andréassian. The suite of lumped GR hydrological models in an R package. *Environmental Modelling & Software*, 94:166–171, 2017.

L. Coron, C. Perrin, O. Delaigue, G. Thirel, and C. Michel. airGR: Suite of GR Hydrological Models for Precipitation-Runoff Modelling. R package version 1.0.15.2. `https://webgr.irstea.fr/en/airGR/`, 2018.

A. de Lavenne, G. Thirel, L. Oudin, V. Andréassian, C. Perrin, and M.H. Ramos. Spatial variability of the parameters of a semi-distributed hydrological model. *Proc. IAHS*, 373:87–94, 2016.

O. Delaigue, L. Coron, and P. Brigode. airGRteaching: Teaching Hydrological Modelling with GR (Shiny Interface Included). R package version 0.2.3.2. `https://webgr.irstea.fr/en/airGRteaching/`, 2018a.

O. Delaigue, G. Thirel, L. Coron, and P. Brigode. airGR and airGRteaching: Two Open-Source Tools for Rainfall-Runoff Modeling and Teaching Hydrology. In: HIC 2018. 13th International Conference on Hydroinformatics. EPiC Series in Engineering, (Eds G.L. Loggia, G. Freni, V. Puleo M.D. Marchis). *EasyChair*, page 541–548, 2018b. doi: 10.29007/qsqj.

K.D. Gayathri, B.P. Ganasri, and G.S. Dwarakish. A Review on Hydrological Models. *Aquatic Procedia*, 4:1001 – 1007, 2010.

H.V. Gupta, H. Kling, K.K. Yilmaz, and G.F. Martinez. Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. *Journal of Hydrology*, 377:80–91, 2009.

B. Guse, M. Pfannerstill, Gafurov, J. Kiesel, C. Lehr, and N. Fohrer. Identifying the connectivity strength between model parameters and performance criteria. *Hydrology and Earth System Sciences*, 21:5663–5679, 2017. doi: doi.org/10.5194/hess-21-5663-2017.

H. Kling, M. Fuchs, and M. Paulin. Runoff conditions in the upper danube basin under an ensemble of climate change scenarios. *Journal of Hydrology*, 424-425:264–277, 2012.

T. Mathevet. *Quel modéles pluie-debit globaux au pas de temps horaire? Développements empiriques et comparison de modéles sur un large échantillon de bassins versants*. PhD thesis, ENGREF (Paris), Cemagref (Antony), The address of the publisher, 2005. 463 pp.

Alberto Montanari. Tutorials of sustainable design of water resources systems. `https://distart119.ing.unibo.it/albertonew/?q=node/158`.

L. Oudin, F. Hervieu, C. Michel, C. Perrin, V. Andréassian, F. Anctil, and C. Loumagne. Which potential evaporation input for a lumped rainfall-runoff model? Part 2 - Towards a simple and efficient potential evapotranspiration model for rainfall-runoff modelling. *Journal of Hydrology*, 303: 290–306, 2005.

J.L. Payan. *Prise en compte de barrages-réservoirs dans un modèle global pluie-débit*. PhD thesis, ENGREF (Paris), Cemagref (Antony), 2007. 256 pp.

J.L. Payan, C. Perrin, V. Andréassian, and C. Michel. How can man-made water reservoirs be accounted for in a lumped rainfall-runoff model? *Water resources research*, 44, 2008. doi: 10.1029/2007WR005971.

C. Perrin. *Vers une amélioration d'un modèle global pluie-débit*. PhD thesis, INPG (Grenoble), Cemagref (Antony), 2000. 256 pp.

C. Perrin, C. Michel, and V. Andréassian. Does a large number of parameters enhance model performance? comparative assessment of common catchment model structures on 429 catchments. *Journal of Hydrology*, 242: 275–301, 2001. doi: doi.org/10.1016/S0022-1694(00)00393-0.

C. Perrin, C. Michel, and V. Andréassian. Improvement of a parsimonious model for streamflow simulation. *Journal of Hydrology*, 279:275–289, 2003.

R. Pushpalatha, C. Perrin, N. Le Moine, T. Mathevet, and V. Andréassian. A downward structural sensitivity analysis of hydrological models to improve low-flow simulation. *Journal of Hydrology*, 411:66–76, 2011.

R Development Core Team. *R: A language and environment for statistical computing. R Foundation for Statistical Computing*. Vienna, Austria, 2008. ISBN 3-900051-07-0. `http://www.R-project.org`.

P. Riboust, G. Thirel, N. Le Moine, and Ribstein P. Revisiting a simple degree-day model for integrating satellite data: implementation of SWE-SCA hystereses. *Hydrol. Hydromech*, 67:70–81, 2019.

L. Santos, G. Thirel, C., and Perrin. Technical note: Pitfalls in using log-transformed flows within the KGE criterion. *Hydrol. Earth Syst. Sci.*, 22: 4583–4591, 2018. doi: doi.org/10.5194/hess-22-4583-2018.

B. Schaefli and H.V. Gupta. Do Nash values have value? *Hydrol. processes*, 21:2075–2080, 2007. doi: doi/epdf/10.1002/hyp.6825.

A. Valéry, V. Andréassian, and C. Perrin. Regionalization of precipitation and air temperature over high-altitude catchments-learning from outliers. *Hydrological Sciences Journal*, 55, 2010.

A. Valéry, V. Andréassian, and C. Perrin. 'As Simple as Possible but not simpler': what is useful in a temperature-based snow-accounting routine? Part 2 - Sensitivity Analysis of the Cemaneige snow accounting routine on 380 catchments. *Journal of Hydrology*, 517:1176–1187, 2014.

J.P. Vidal, E. Martin, L. Franchistéguy, M. Baillon, and J.M. Soubeyrou. A 50-year high-resolution atmospheric reanalysis over France with the Safran system. *International Journal of Climatology*, 30:1627–1644, 2009. doi: doi.org/10.1002/joc.2003.

H.S. Wheater. Modelling hydrological processes in arid and semi-arid areas: An introduction to the workshop. *Hydrological Modelling in Arid and Semi-Arid Areas*, pages 1–20, 1 2010.

Islam Zahidul. A Review on Physically Based Hydrologic Modelings. 5 2011. doi: 10.13140/2.1.4544.5924.

Acknowledgments

I would like to thank my supervisor, Prof.ssa Elena Toth, for the precious advice she has provided throughout my time as her student and for the opportunity that she gave me to accomplish my master thesis work abroad.
I would also like to express my gratitude to the Dr. Guillaume Thirel, my co-supervisor, for his patient guidance and encouragement throughout these last months that I spent at the IRSTEA research center in Paris.