User Manual: HOOPLA version 1.0.2

Original version – Date: 2018/05/09 Authors: Antoine Thiboult, Carine Poncelet, François Anctil

Version 1.0.1 – Date: 2019/04/15 Author: Antoine Thiboult – Improved description of hydrological model changes

Version 1.0.2 – Date: 2019/10/22 Author: Antoine Thiboult – Correct typos, Improve description of state variable updating, Update flowcharts

Abstract

This tutorial presents an overview the HydrOlOgical Prediction LAboratory (HOOPLA) developed at Université Laval. HOOPLA is modular framework presented as a standalone software coded in Matlab. It performs calibration, simulation, and forecast for multiple hydrological models and various time steps, but also provides an array of options and features. This tutorial provides potential users with the basic knowledge to format input data, run computations, and understand the outputs, and briefly describes the available tools.

Contents

1 Introduction

1.1 What is HOOPLA?

HOOPLA is the acronym for "HydrOlOgical Prediction LAboratory", a framework providing a modular system for lumped hydrological modeling. HOOPLA is developed and used at the Department of Civil and Water Engineering, at Université Laval (Canada) and is fully coded in Matlab. It is designed as a platform to experiment with hydrological tools and is available as a standalone software including a collection of modeling tools:

1. Twenty hydrological models
2. One snow accounting routine (SAR)
3. Three potential evapotranspiration (PET) formulas
4. Two calibration algorithms
5. Two data assimilation schemes
6. Seventeen performance criteria

HOOPLA has been designed to fulfill three major requirements: to provide a user-friendly environment, to be flexible regarding user's modeling preferences, and to allow easy integration of tools created by the user for testing or development purposes. In its current state, HOOPLA provides only limited functionalities to format input data and to produce graphical outputs of the results. Note as well that HOOPLA does not include any other data than the demo data.

1.2 Does HOOPLA suit my needs?

Hydrometeorological research or development

Modular modeling options are easily available to the user for inter-comparison or development purposes. For instance, to test a novel hydrological model structure, one needs only to add this model to the pool of already available ones. Computations will be fully carried out by HOOPLA and the already available models may then serve as a benchmark to evaluate the model performance. In the same way, it is possible to test alternatives to the hydrological tools that are implemented in HOOPLA (calibration algorithm, PET formulation, etc.)

Operational forecasting

HOOPLA showed promising forecast performance in offline mode [Thiboult et al., 2017]. Tests are currently carried out by the Direction de l'Expertise Hydrique du Québec (formerly Centre d'Expertise Hydrique du Québec) for a possible integration in their real-time operational streamflow forecast system.

Model coupling

HOOPLA outputs can be used in combination with external programs and applications that require hydrological inputs. For instance, forecasting results can be used to retrospectively assess the quality of decision rules or as input for decision models.

Education

HOOPLA was not designed for teaching or learning hydrology; however HOOPLA is easy to use and provides demo data as well as hydrological modeling tools. Therefore basic computations and hypothesis testing can be carried out in a training context. Moreover, HOOPLA has the potential to be combined with a graphical interface to enhance communication between advanced modeling capabilities and student users.

1.3 System requirements

Software requirements

Matlab has to be installed to run HOOPLA. Note that backward compatibility is not ensured for Matlab releases older than version 7.13 (R2011b). Additionally, the statistical and financial toolboxes must be installed. The parallel computing toolbox is strongly advised to benefit from the parallel computing features implemented in HOOPLA and save computational time.

Hardware recommendations

There is not minimal hardware system requirements, but :

- Faster CPU are advised, even if HOOPLA also works on a modest computer. We recommend a multi-core computer to take advantage of the parallel computing feature.
- We highly recommend to using a 64-bit operating system and a 64-bit Matlab version to minimize RAM memory constraints. In the case you wish to use an older operating system and/or an older Matlab, maximum matrix size could be limited to one to few Gb. This type of memory limitation can arise quickly when using hydrometeorological ensemble tools such as the Ensemble Kalman filter or meteorological ensemble forcing.

1.4 Copyright and citing this work

HOOPLA is protected by the BSD 2-Clause License (https://opensource.org/licenses/BSD-2-Clause). You are free (and encouraged!) to run, study, share, modify and redistribute the software as long as you respect the copyright conditions. We would also greatly appreciate if you share with us improvements you made to HOOPLA.

If you use HOOPLA and produce a communication, please refer both to the user manual corresponding to you version of HOOPLA and to the following article: Thiboult et al. [2018].

2 Tools

This section presents the tools provided alongside HOOPLA.

2.1 Hydrological models

A collection of 20 lumped conceptual hydrological models is provided, based on the initial bibliographic selection of Perrin [2000]. This selection was revisited by Seiller et al. [2012] to retain models based on their:

1. Complexity: only lowly to moderately complex models were selected providing that they perform well. Therefore, the retained models possess a rather low number of free parameters and reservoirs (see Table 1).

2. Diversity: the models were developed by different research and engineering teams in different contexts and for different purposes. Differences in model structure is therefore fostered and should reflect in dissimilar behaviors.

Table 1: Main characteristics of the 20 lumped models

Model name	Number of parameters	Number of reservoirs	Modif.	Derived from
HydroMod1	6	3	Slightly	BUCKET [Thornthwaite and Mather, 1955]
HydroMod2	9	2	Slightly	CEQUEAU [Girard et al., 1972]
HydroMod3	6	3	Slightly	CREC [Cormary and Guilbot, 1973]
HydroMod4	6	3	Slightly	GARDENIA [Thiery, 1982]
HydroMod5	4	2	Similar	GR4H [Mathevet, 2005]
HydroMod6	9	3	Substant.	HBV [Bergström and Forsman, 1973]
HydroMod7	6	5	Slightly	HYMOD [Wagener et al., 2001]
HydroMod8	7	3	Slightly	IHACRES [Jakeman et al., 1990]
HydroMod9	7	4	Slightly	MARTINE [Mazenc et al., 1984]
HydroMod10	7	2	Similar	MOHYSE [Fortin and Turcotte, 2007]
HydroMod11	6	4	Similar	MORDOR [Garçon, 1999]
HydroMod12	10	7	Substant.	NAM [Nielsen and Hansen, 1973]
HydroMod13	8	4	Slightly	PDM [Moore and Clarke, 1981]
HydroMod14	9	5	Substant.	SACRAMENTO [Burnash et al., 1973]
HydroMod15	8	3	Slightly	SIMHYD [Chiew et al., 2002]
HydroMod16	8	3	Substant.	SMAR [O'Connell et al., 1970]
HydroMod17	7	4	Substant.	TANK [Sugawara, 1979]
HydroMod18	7	3	Substant.	TOPMODEL [Beven et al., 1984]
HydroMod19	8	3	Slightly	WAGENINGEN [Warmerdam et al., 1997]
HydroMod20	8	4	Substant.	XINANJIANG [Zhao et al., 1980]

We strongly emphasize that the hydrological models implemented in HOOPLA are not the original models nor their most up-to-date versions. The only exception is GR4H that is identical to the version provided in the airGR package (v1.0.9.64) proposed by Irstea [Coron et al., 2017]. The models were implemented based on books, journal or conference publications, opening the door to some subjective programming. The column "Modif" in Table 1 gives a qualitative description of the changes that have been performed on the models (Similar model, slightly modified, substantially modified). Models were coded by Seiller et al. [2012] based on Perrin [2000] and the original articles (column "Derived from" in Table 1).

In addition, the models were intentionally modified to fit within the HOOPLA framework. These changes aim at:

- Adapting model structure to lumped modeling (CEQUEAU, PDM, TOPMODEL, XINANJIANG). Some models were initially distributed and have been modified to model the hydrological response at the catchment scale. Therefore spatial heterogeneity of meteorological forcing and catchment properties were discarded.

- Applying models at multiple time steps. The initial temporal resolution of the models were varied even if models could theoretically have been used "as-is" at different time steps. Tests carried out within the Université Laval hydrology team have shown that model parameters recalibration can account for most of the impacts of a temporal resolution change. Some additional modifications have been performed to improve model performance when the recalibration failed.
- Reducing the number of free parameters. Based on user or developers' comments and additional sensitivity analyses carried out within the team, some parameters have been set to a fixed value.
- Focusing on rainfall-runoff modeling. Several models included a snow accounting routine and|or a PET formulation. Only the rainfall-runoff part of these models has been kept, snowmelt and PET computations being handled externally by HOOPLA.
- Standardizing data requirements. All models share the same input, namely precipitation (the sum of the rainfall and snowmelt) and PET. For instance, some models initially used soil moisture as input data. This latter was dismissed and soil moisture was inferred using parametric logistic functions.
- Standardizing model outputs. All models calculate a streamflow time series. State variables can additionally be retrieved by the user (see Section 4.2 and 5.2).
- Any other simplification to the model structure that does not infer with performance is preserved.

Most models were initially developed to be used at a daily time step, while HOOPLA supports multiple time steps. For example, the user may decide to run the model at a 3 hour time step, at a 24h, 48h time step, or any other time step. Yet, HOOPLA has only been tested for a 3h time step and a 24h time step. Therefore, performing calculation based on a temporal resolution shorter than 3h or longer than 24h is more exploratory work. Hydrological model production and transfer functions may possibly not be appropriate for different time step. The model structure remains the same, regardless of the specified time step. Changes in the characteristic times of the processes is accounted solely by the model parameters. This means that the input data are not temporally aggregated or disaggregated prior to be processed by the models, they remain unaltered.

2.2 Potential evapotranspiration formulas

Potential evapotranspiration is used to quantify the atmospheric demand in evaporation. In hydrology, potential evapotranspiration is primarily used as an input data to compute the actual evapotranspiration that refers to the water quantities transfered to the atmosphere as vapor. Potential evapotranspiration is preferred over actual evapotranspiration mainly because it requires no prior knowledge of vegetation and soil properties, i.e. it can be easily estimated from meteorological variables. A large number of potential evapotranspiration formula exist. Only three were implemented in HOOPLA (see Table 2) for two reasons: hydrological models usually express little sensitivity towards the PET input and to ensure that PET can be computed from a limited set of inputs.

Table 2: Potential evapotranspiration formulas

Name	Required input data	Reference
Oudin	Date, T, Lat	Oudin et al. [2005]
Kharrufa	Date, T, Lat	Kharrufa [1985]
Penman	Date, T, Lat, Glorad, Relhum, Tmax, Tmin, Wndspd, z	Penman [1948]

Kharrufa and Oudin formulas rely on an energy balance rationale, their main inputs being temperature (and extraterrestrial radiation for Oudin). The Penman formula rely on a combinational approach (energy balance and aerodynamic) and is more data intensive.

These formulas can be theoretically applied to various time step duration. For instance, Oudin was successfully tested for 3-hour and 24-hour time steps by adapting the way the extraterrestrial radiation is computed. Note that this adaptation is already included in the initialization phase for Oudin while it seems that it is not necessary to perform any correction for the Kharrufa formula. Last but not least, the Penman formula is implemented in HOOPLA but could not be verified due to lack of data.

2.3 Snow accounting routine

The CemaNeige [Valéry et al., 2014] snow accounting routine (SAR) is implemented in HOOPLA. CemaNeige is usually used to provide a hydrological model with precipitation inputs that account for snow melt/accumulation and the partition of precipitation between liquid and solid phases. It is now the only snow accounting routine included in HOOPLA as other tested formulation did not match CemaNeige performance. Yet, we do not rule out the possibility to add other SAR in the future.

This is firstly achieved through a nonparametric extrapolation of temperature and precipitation into five altitudinal layers of equal area. Then the distinction between liquid and solid precipitations relies on the air temperature at each altitudinal zone. Minimum and maximum air temperature can be used in addition to improve this estimation. For catchment with low elevation differences, precipitation is assumed independent from altitude [Valéry et al., 2014].

Lastly, CemaNeige estimates the snow water equivalent of the snow pack based on the distributed forcing, a base temperature for the degree-day scheme and two free parameters. The based temperature is set at 0°C. The two parameters are: Ctg [-] that describes the thermal inertia of the snow pack and Kf [mm/t] a degree-day melting factor, with t being the duration of the modeling time step. Higher Ctg values delay snowmelt and higher Kf values induce larger snowmelt. CemaNeige works well with default (uncalibrated) values for Ctg and Kf: $Ctg = 0.25$ and $Kf = 3.74mm/t$ are the default daily values and $Ctg = 0.93$ and $Kf = 0.40mm/t$ are the default 3-h values.

As for hydrological models and PET formulas, CemaNeige can be run at multiple time steps. Results indicated that changes in the snow accumulation and melt characteristic times can be accounted by the SAR parameters. In other words, CemaNeige structure remains unchanged regardless of the time step in use, and only its parameters will change accordingly. However, the concept of degree day on which CemaNeige relies becomes more fuzzy when calculations are performed with a time step different from 24h. We believe that in such case it should be rather referred as degree-3h for instance.

Lastly, HOOPLA includes (time-dependant) default sets of parameters for CemaNeige. The 24h default parameters are defined from [Valéry et al., 2014], while the 3h parameters were identified by calibrating the 20 hydrological models over 31 catchments located in the southern part of the Province of Québec. Results indicated that the SAR parameters vary little across catchments and models (this is especially true for the 3h time step). The default values are :

- Snowmelt factor: 3.74mm/d (24h) and 0.4mm/3h (3h)
- Cold-content factor: 0.25 (24h) and 0.93 (3h)

2.4 Data assimilation schemes

Data assimilation aims at integrating information contained in observations to improve simulation accuracy while accounting for uncertainties in the measurements and model. More precisely, data assimilation schemes reinitialize the model states based on the latest available observations to provide better initial conditions for the next modeling step.

Two probabilistic, sequential data assimilation schemes are provided in HOOPLA:

- The Sequential Importance Resampling filter (SIR), belonging to the class of particle filters (PF) and also referred to as Sequential Monte Carlo or Bootstrap filter.
- The Ensemble Kalman Filter (EnKF)

These schemes are referred to as probabilistic because the quantities of interest, the observation and the model states, are described by probability density functions. Since it is not possible to fully estimate these densities, they are approximated by representative samples. In HOOPLA,

the model state distribution is obtained by adding random perturbations to the model's input (precipitation and temperature and/or PET) as opposed to direct model state perturbations. These perturbations are randomly drawn from densities with zero mean and a standard deviation that describes the variables uncertainty. Also, in HOOPLA, the nature of the observation is limited to the streamflow measurements. The standard deviations representing the streamflow and the meteorological forcing uncertainties must be set by the user (see Section 5.2 and 6.2). Parameters describing these uncertainties are often called hyperparameters: they influence the success of both assimilation schemes to a great extent [Thiboult and Anctil, 2015]. As a result to the probabilistic forcing, the model outputs are also probabilistic and form a collection of individual simulations called an ensemble.

In a data assimilation scheme, it is important to properly identify the rainfall-runoff model states that should be updated. A default configuration is available for the 20 models presented in Sec. 2.1. State variable selection was carefuly carried out based on the procedure described in Thiboult and Anctil [2015]. However, HOOPLA offers some flexibility regarding this aspect. The user may specify different state variables by editing the file *./Data/<TimeStep>/Misc/reservoir_to_update.mat* (see also Sec. 7.1. Note that HOOPLA does not currently allow updating the SAR states.

The EnKF and the PF are sequential in nature because the hydrological model is seen as a hidden Markov model. All the necessary information to characterize the system are contained in the previous and current time steps. Therefore, every time step can be processed sequentially. Any sequential data assimilation scheme consists in two main stages. Firstly, the prediction step generates the prior distribution for the system states by integrating the forward operator (the hydrological model). Secondly, the updating step creates the posterior distribution by reshaping the prior distribution according to the observation.

The idea of the PF and EnKF is to derive a solution to this recursive Bayesian estimation problem, where one aims to sequentially update some initial density function as more information become available to construct the posterior probability density function. Even if both the PF and the EnKF adopt a Monte Carlo estimation method, they use a slightly different approach and resort to different approximations.

- PF: The principle is to approximate the true posterior distribution by a set of random measures (particles) and associated weights. Because it may be impossible to directly sample from the true distribution, it is convenient to resort to the principle of importance sampling that allows to derive the properties of a particular distribution from a different one. The latter is called the importance, the proposal, or the sample distribution. Several importance densities may be selected, but choosing the prior as the importance density is often a convenient choice. This is the case in HOOPLA. Practically, particles weight are derived according to the proximity between the particles and the observation, i.e. particles receive a higher weight if they fall closely to the observation. Yet, the weight of all particles (except one) will almost systematically tend to zero within a few iterations. To prevent this degeneracy, particles are resampled according to their respective weights. More details about Bayesian inference and the PF derivation can be found in Arulampalam et al. [2002]. The latter was the main inspiration for the SIR code implemented in HOOPLA.

- EnKF: The Kalman Filter is the optimal solution to the recursive Bayesian tracking problem provided that the system is linear and all distributions Gaussian. Such restrictive assumptions are rarely met in real world problems but the EnKF allows to get free from the linearity requirement. Even if the EnKF can resort to various distributions, it works best in a Gaussian environment. The updating scheme relies on the computation of the Kalman gain from the error covariance matrix, which is a measure of the accuracy of the state estimate. This Kalman gain is subsequently used with the innovation (the difference between the observation and the prior simulation) to update the hidden model states. A sound description of the EnKF is presented in Evensen [2003] and the EnKF code in HOOPLA was written according to Mandel [2006] recommendations.

2.5 Calibration algorithms

Calibration aims at identifying an optimal parameter set with regards to a chosen cost function (or efficiency criterion). The parameter space is an n-dimensions space (n being the number of free parameters in the model) corresponding to the combination of all possible values each parameter

can take. A point in the parameter space is hence equivalent to a parameter set. The response surface consists in an ensemble of z cost function values evaluated at z points of the parameter space. All calibration algorithms somehow sample the parameter space to build a corresponding response surface for identifying an optimal parameter set. HOOPLA provides two calibration algorithms: the Shuffle Complex Evolution [SCE, Duan et al., 1992] and the Dynamically Dimensioned Search algorithm [DDS, Tolson and Shoemaker, 2007]. Both methods are iterative and global (they seek to find the optimal set among the entire parameter space). They mainly differ in the way the parameter space is sampled.

SCE is based on several groups of points (parameter sets called complexes) spanning the parameter space. At each iteration, these complexes evolve along several mechanisms (namely reflection, contraction, and mutation) and are subsequently recombined. These steps are iterated until the algorithm converges or a preset maximum of iteration is reached.

DDS aims at finding the best possible solution with regards to a preset maximum number of iterations. In other words, the convergence does not need to be achieved for the algorithm to stop. Therefore DDS is well suited for calibrating models with a large number of free parameters or that requires important computing ressources. The parameter space is sampled via random perturbations and reflections. During the optimization, DDS progressively fixes parameter values, hence reducing the parameter space dimension.

The cost function, the maximum number of iterations, and other specificities are controlled by the user (see section 4.2).

3 Getting started

3.1 How is HOOPLA organized?

For most users, HOOPLA works as a standalone software: one enters its preferences and HOOPLA automatically executes the corresponding functions to produce the results. All functions are heavily commented for ease the use and understanding of the software. There is no graphical interface. The user controls HOOPLA via so-called "switches" defined by the masterfile launch_HOOPLA, which is to be edited and executed. HOOPLA functions according to the specific directories organization given in Figure 1.

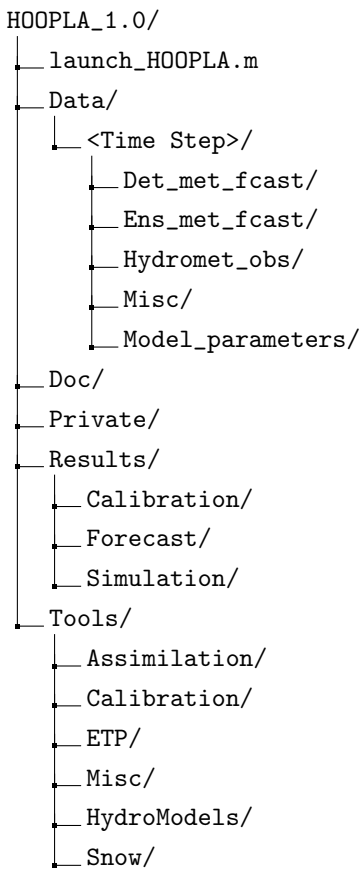

Figure 1: Directories organization of HOOPLA.

The directories content are as follows:

- *Data* contains all data needed to run the calculations: catchment names, model names, observations and forecasts, model parameters, etc. Note that there must be one subdirectory by time step named *Nh* where *N* refers to the number of hours by time step (ex: 3h). The reader is referred to Sections 4.1, 5.1, and 6.1 for a more detailed description of the data.
- *Doc* contains the documentation about HOOPLA.
- *Private* contains additional (and optional) user-defined functions to expand HOOPLA's functionalities.
- *Results* receives the outputs of HOOPLA computations, see Sections 4.3, 5.3, and 6.3 for more details.
- *Tools* gathers the functions HOOPLA needs to perform the requested computations. Section 2 provides more details about the tools incorporated in HOOPLA. Some functions used internally by HOOPLA and contained in the Misc directory are described along with the relevant switches (Sections 4, 5, and 6).

3.2 Data requirement

Table 3 lists the required datasets for various tasks. Their content and format are described in dedicated tables: Table 5 for calibration, simulation, and forecast, and Tables 11 and 10 for meteorological ensemble and deterministic forecasts, respectively.

Table 3: List of required datasets. The abbreviations OL and DA refers to Open Loop and Data Assimilation, respectively. The file location is defined from *HOOPLA/Data/<TimeStep>*.

Mode / Data	Cal.	Sim.		Fcast.						File Location
				Perfect		Determin.		Ensemble		
		OL	DA	OL	DA	OL	DA	OL	DA	
Hydrometeo. data obs.	✔	✔	✔	✔	✔	✔	✔	✔	✔	Hydromet_obs
Deterministic Meteo. forecast						✔	✔			Det_met_fcast
Ensemble Meteo. forecast								✔	✔	Ens_met_fcast
Catchment Names	✔	✔	✔	✔	✔	✔	✔	✔	✔	Misc
Hydro Model Names	✔	✔	✔	✔	✔	✔	✔	✔	✔	
PET Model Names	✔	✔	✔	✔	✔	✔	✔	✔	✔	
SAR Names	✔	✔	✔	✔	✔	✔	✔	✔	✔	
Reservoir to Update			✔		✔		✔		✔	
Calibrated Parameters		✔	✔	✔	✔	✔	✔	✔	✔	Model_parameters
Hydro Model Param Bound.	✔									
Snow Model Param Bound.	✔									

3.3 The Launch_HOOPLA file

Launch_HOOPLA.m is HOOPLA's masterfile. It assigns values to various switches translating user preferences into HOOPLA computations. For regular users (i.e. not advanced users), this is the only file that needs editing. An extract of launch_HOOPLA.m is reproduced in Listing 1 to illustrate its structure:

Listing 1: Organization of HOOPLA's masterfile (extract of launch_HOOPLA.m).

```
%% Section 1: Switches: 1=yes/on, 0=no/off [EDIT THIS SECTION]

% Calibration / Simulation / Forecast
Switches.calibration.on =1; % Run calibration
...
% Study period
Switches.calStart = '1994/01/01/03:00:00';
Switches.calEnd =   '2007/01/01/00:00:00';
...
% General switches
Switches.timeStep        ='3h'; % Computation time step.
Switches.snowmeltCompute.on =1;  % Compute snowmelt
...
% Calibration switches
Switches.calibration.method ='SCE'; % Choose between 'DDS' and 'SCE'
Switches.calibration.score ='NSE'; % Performance criterion (RMSE, NSE, KGE...)
```

```
...
% Forecast switches
Switches.forecast.perfectFcast.on =0; % If on, observations are used as weather forecast
Switches.forecast.metEns.on     =1;   % Uses meteorological ensemble forecasting
...
% Data Assimilation switches
Switches.DA.on=0;            % switch assimilation on/off
Switches.DA.tech='PF';   % 'EnKF', 'PF' or 'PertOnly'
...

%% Section 2: Loading framework tools [DO NOT EDIT THIS SECTION]
...

%% Section 3: Catchment, hydrological model, PET model, and snow model selection [EDIT
    THIS SECTION]

% Catchments
Switches.isC=zeros(numel(nameC),1); % If isC(n)==1, catchment n is ran
Switches.isC(:)=1;

% Hydrological models
Switches.isM=zeros(numel(nameM),1); % If isM(n)==1, model n is ran
Switches.isM(:)=1;
...

%% Section 4: Run HOOPLA [DO NOT EDIT THIS SECTION]
...
```

The file is organized in four main sections:

- Section 1: *Switches* defines the user preferences, as detailed in Sections 4.2, 5.2, and 6.2. This section **should be edited**.
- Section 2: $Loading framework tools$ makes available the hydrological tools detailed in Section 2 and defines the catchments and model names. This section **should NOT be edited**.
- Section 3: $Catchment, hydrological model, PET model, and SAR selection$ identifies the catchments, models, PET, and SAR to be used for computations. This section **should be edited**.
- Section 4: $Run HOOPLA$ automatically launches the computations. It **should NOT be edited**. Result files are described Sections 4.3, 5.3, and 6.3.

3.4 General users switches

The general switches presented in Table 4 are mandatory. They control how streamflow simulations are issued, should it be for calibration, simulation, or forecast purposes.

Several switches and internal HOOPLA functions are described here in more details:

- *timeStep* informs on the temporal resolution of the observations (the resolution of the resulting simulations will match it). For the models currently available in HOOPLA, only the '3h' and '24h' cases have been tested and implemented.
- *petCompute.on* allows the user to provide HOOPLA with externally obtained PET inputs (if deactivated) or to provide HOOPLA with the inputs to the chosen PET formulas and let HOOPLA produce the PET series (if activated). Section 4.1 details how to provide HOOPLA with an externally-computed PET series.
- *snowmeltCompute.on* allows the user to model snow accumulation and melt using a SAR (if activated). If deactivated, snow accumulation and melt effects will be discarded unless the user provides HOOPLA with a precipitation input that already account for these effects (i.e. if the user has run a SAR externally).
- *explortLight.on*: when activated, fewer results are exported resulting in a gain of memory and computing time. If deactivated, information about hydrological model intermediate flow

Table 4: General switches. Switches ending by ".on" can be set to only two values, either 0 or 1 which correspond respectively to yes/on/activate and no/off/deactivate.

Switches name	Description	Possible values	Advised values	Format
timeStep	Computation time step in hours	nh	3h or 24h	Char.
petCompute.on	Compute PET	1 or 0	-	Scalar
snowmeltCompute.on	Compute snowmelt	1 or 0	-	Scalar
warmUpCompute.on	Add warm up before modeling	1 or 0	1	Scalar
verb.on	Display extensive information	1 or 0	-	Scalar
exportLight.on	Export fewer results	1 or 0	-	Scalar
overWrite.on	Overwrite preexisting files	1 or 0	-	Scalar
parallelCompute.on	Enable parallel computing	1 or 0	-	Scalar
isC	Indicates on which catchments computation will be performed	1 or 0	-	Vector
isM	Indicates which hydrological models are used for computation	1 or 0	-	Vector
isE	Indicate which PET models are used for computation	1 or 0	isE(1)=1[1]	Vector
isS	Indicate which SAR are used for computation	1 or 0	isS(1)=1[2]	Vector

[1]This corresponds to the Oudin PET formulation
[2]This corresponds to the CemaNeige SAR model. Note that in this version of HOOPLA, CemaNeige is the only snow routine available

components (and SAR states if snowmeltCompute.on is activated) are saved. For a more detailed meaning of these additional results, the user is referred to the source code where the header of the model function describes every possible output.

- *isC*, *isM*, *isE*, and *isS* specify which catchment, hydrological model, PET model, and SAR are included in the computations. They are vectors of size identical to the following variables (stored in the *catchment_ names.mat*, *hydro_ model_ names.mat*, ...): *nameC*, *nameM*, *nameE*, and *nameS*, respectively. For example, if the i^{th} element of *isC* is set to 1, computations will be performed for the i^{th} catchment.

- *warmUpCompute.on*: when activated (=1) the function warmup.m is called to initiate the states of the hydrological models/snow accounting routine before running computations over the study period. This procedure is also often called the model spin up. If the meteorological forcing are available for approximatively 3000 time steps prior to the study period, they are used for the warm up. If they are not available, a mean year a precipitation is identified among the database. The forcing corresponding to the mean precipitation year is then replicated to reach 3000 time steps and used to warm the model up. It is recommended to set this switch to 1.

3.5 Catchment database

HOOPLA requires to know the name of the catchments that form the user-defined database. To do so, the file catchment_names.mat located in the *HOOPLA/Data/<TimeStep>/Misc* directory needs to be defined. It acts like a dictionary that lists all the catchments the user may want to exploit but not the data themselves. This file should contain a cell array named *nameC* of size $nC \times 1$ where nC refers to the number of catchment in the user database. *nameC* should contain the name of the catchment *xxx* stored as a string as specified in the name *Hydromet_ obs_ xxx* (see Section 4.1, 5.1, and 6.1). Note that switch *Switches.isC* defined in Table 4 indicates which catchment of *nameC* should be used.

HOOPLA comes with a demo database containing hydrometeorological data for one catchment for 3-hour and 24-hour time steps, to help new users to understand how data management is performed. In addition, the script (*FormatCatchDataObs.m*) is provided in the *Private* directory

to suggest default values for input data and help shape the database. It is suggested to closely match the demo format. Moreover, it is advised to store all variable in matrices of double precision, since they are more compatible with MatLab functions. As a reminder, HOOPLA does not include any function to format inputs and it is therefore the user's responsibility to create a dataset that matches HOOPLA format. Detailed data requirement and formats are presented in the following sections.

4 Calibration

4.1 Data

Hydrometeorological data

Table 5 presents typical data requirements when CemaNeige is used as snow accounting routine and the Oudin formula is used for PET computations. Hydrometeorological data are common to calibration, simulation (Sec. 5), and forecast (Sec. 6). They can be classified into several groups: data necessary in all situations (Table 5a), those necessary for the SAR (Table 5b), and those needed by the PET formulation (Table 5c). Therefore, one should note that required inputs may vary depending on the SAR and the PET used. A short code named *FormatCtchData.m* is provided in the *HOOPLA/Private/* directory to help the user create a hydrometeorological file with suitable format and provide default values, in particular for CemaNeige. Additionally, a demo file (*Hydromet_obs_demo*) is provided for the 3-hour and the 24-hour time step in the *HOOPLA/Data/3h/Hydromet_obs* and the *HOOPLA/Data/24h/Hydromet_obs directories*, respectively. The user is encouraged to take a closer look at these demo files prior to formating its own data.

Files containing hydrometeorological data must respect the following rules:

- The file should contain the data for a single catchment
- All necessary data for a catchment should be in a single file
- The file must be located in the *HOOPLA/Data/<Time Step>/Hydromet_obs* directory
- The name of the file should respect the format *Hydromet_obs_xxx* where *xxx* is the catchment identifier defined by the user. Your catchment identifier should only contain letters and numerals (no accentuated letters nor any special character or punctuation).
- Your catchment identifier should only contain letters and numerals (no accentuated letters nor any special character or punctuation).

The many hydrometeorological variables contained in the *Hydromet_obs_xxx* file should be stored as matrices that respect names and sizes as presented in Table 5. We recommend using a double precision for all variables since some Matlab functions may be significantly slower when performed on single precision, or even fail...

Meteorological data used as hydrological model inputs, namely the PET and the runoff depth (sum of liquid precipitation and snowmelt), or any variable that will be used to compute the PET and the runoff depth must be positive, continuous, and should not contain any NaN. It is the user responsibility to make sure that data are coherent and match the aforementioned requirements since HOOPLA does not perform such verifications. HOOPLA solely checks if all mandatory data, given the user selection (hydrological model, CemaNeige, PET formula), are included in the *Hydromet_obs_xxx files*. It does not check for dates discontinuity, negative or missing precipitation/PET/streamflow values. Missing values in the streamflow record are allowed if set as NaN: the objective function will be computed only on time steps were streamflow values are available.

Data specific to the calibration algorithm

Both calibration algorithms implemented in HOOPLA require parameter boundaries (minimum and maximum values of the hydrological model parameters). Default boundary values for the 20 lumped models and CemaNeige model are provided alongside HOOPLA (files *model_param_boundaries.mat* and *snow_model_param_boundaries.mat* located in the *HOOPLA/Data/<Time Step>/Model_parameters* directory).

Table 5: Content of the *Hydromet_ obs* datafile for calibration. nTS is the number of time steps (not the temporal resolution) the calibration is performed on.

(a) Mandatory data

Name	Description	Units	Size
$Date$	Date[1]	[-]	$nTS \times 6$
Pt	Total precipitation	[mm]	$nTS \times 1$
PET	Potential evapotranspiration[2]	[mm]	$nTS \times 1$
Q	Observed streamflow	[mm]	$nTS \times 1$

[1]The '*Date*' format corresponds to the datevec function in Matlab. First column contains the year, the second the month, the third the day, the fourth the hour, the fifth the minutes, and the sixth the seconds, all stored as double. If the user works at a daily time step, only the column containing the year, month and day are needed.
[2]PET can also be computed internally by HOOPLA. In this case, the user must provide the input data for the PET formula (see exemple for the Oudin formula in subtable 5c).

(b) SAR data (optional, example for CemaNeige)

Name	Description	Units	Size
$Beta$	Correction of precipitation according to elevation[1]	[m^{-1}]	1×1
$gradT$	Temperature gradient[1]	[°C/100m]	365×1
Pt	Total precipitation	[mm]	$nTS \times 1$
$QNBV$	Average annual snow accumulation[1]	[mm]	1×1
T	Air temperature	[°C]	$nTS \times 1$
$Tmax$	Maximum temperature	[°C]	$nTS \times 1$
$Tmin$	Minimum temperature	[°C]	$nTS \times 1$
$Vmin$	Percentage of Kf that corresponds to the minimal melting rate[1]	[-]	1×1
$Zz5$	Altitude for the five altitude zones	[m]	1×5

[1]Default values for CemaNeige data are provided in the *FormatCatchDataObs.m* script (located in the *HOOPLA/Private directory*).

(c) PET data (optional, example for the Oudin formula)

Name	Description	Units	Size
$Date$	Date	-	$nTS \times 1$
Lat	Catchment Latitude	[°]	1×1
T	Air temperature	[°C]	$nTS \times 1$

4.2 Switches

This section describes switches specific to hydrological model calibration. Please note that the general switches presented in Table 4 also need to be set.

Table 6: User switches for calibration. Switches ending by ".on" can be set to only two values: 0 or 1, with 0=no/off/deactivate and 1=yes/on/activate.

Switches name	Description	Possible values	Advised values	Format
calibration.on	Run calibration	1 or 0	-	Scalar
calStart	Beginning of calibration period[1]	-	-	Char.
calEnd	End of calibration period[1]	-	-	Char.
calibration.export.on	Export calibrated parameters[2]	1 or 0	1	Scalar
calibration.snowCal.on	Calibrate snow module or use default value	1 or 0	0	Scalar
calibration.method	Calibration algorithm	SCE or DDS	SCE[3]	Char.
calibration.rmWinter.on	Remove Quebec "ice months" (Dec, Jan, Feb, Mar)	1 or 0	1	Scalar
calibration.score	Performance criterion for calibration[4]	-	-	Char.
calibration.maxiter	Maximum number of iteration during calibration	-	500	Scalar
calibration.SCE.ngs	Number of complexes for the SCE optimization	-	25	Scalar

[1]The format for *CalStart* and *CalEnd* must be 'yyyy/mm/dd/HH:MM:SS' and can be alternatively 'yyyy/mm/dd' for the 24-hour time step. It is important to make these dates coincide with the hydrological time step and the beginning of the day. For example, if one uses a 3h time step, one should define the start and the end date as a multiple of 3 (i.e 0am, 3am, 6am,..., 9pm). In the case one would use a 24h time step, the start and end date should coincide with 0am (midnight).

[2]The switch calibration.export.on calls the `export_calibration_results.m` function that adequately formats and stores the result of calibration in the *HOOPLA/Data/<Time Step>/Model_parameters/calibrated_param.mat* file. This file is mandatory if simulation or forecast are to be issued. Note that it can also be called independently, without any argument.

[3]In the case where computational time is not a strong constrain and hydrological models have few degrees of freedom (such as the ones provided by default in HOOPLA), SCE usually provides better results. In the opposite case, DDS is suggested.

[4]All scores available for calibration are computed with the `det_scores.m` function. It calculates 17 deterministic scores to assess the goodness of fit between observed and simulated streamflow. Many popular scores are included such as, for example, the Root Mean Square Error (RMSE), the Nash-Sutcliffe Efficiency (NSE), or the Kling-Gupta Efficiency (KGE). To list all scores, you can enter help `det_scores` in your Matlab terminal.

4.3 Results

Table 7 lists HOOPLA calibration outputs. The result file is named *Cwww_Hxxx_Eyyy_Szzz.mat*, where www, xxx, yyy, and zzz refer to the catchment, the hydrological model, the PET, and the SAR name, respectively. This file is situated in the *Result/Calibration* folder. It includes several arrays with named fields (Matlab structures) that contain the data of interest. For instance, the structure '*Result*' has a '*bestParam*' field that holds the values of the best hydrological model parameters set identified through calibration. The content of this file may vary according to the specification of the switches. For example if the *Switches.exportLight.on* is set to 0, additional results will be recorded and saved.

Once calibrations are performed, we recommend the user to run the *export_calibration_results.m* function to create and save the file that is necessary for simulation and forecast, either by setting the *Switches.export.on* to 1 during calibration or by calling this function directly from the Matlab terminal. The created file is named *calibrated_param.mat* file and located in the *HOOPLA/-Data/<TimeStep>/Model_parameters* directory. A demo set of results can be downloaded from https://github.com/AntoineThiboult/HOOPLA_Demo_Catchment_Result.

Table 7: Content of the result file for calibration. nTS is the number of time steps the calibration is performed on.

(a) Systematic results

Name	Description	Units	Size
DataCal.			
...	All hydrometeorological input data		
Result.			
Qs	Simulated streamflow with the best set of parameters identified during calibration	[mm]	$nTS \times 1$
bestParam	Best hydrological model parameters[1]	[-]	$nParam \times 1$
bestf	Cost/score of the best model parameters	[-]	$nTS \times 1$
allBestf	Evolution of the Cost/score[1]	[-]	$nIter \times 1$
DateCal	Dates on which the calibration is performed	[-]	$nTS \times 6$
Switches.			
...	All user switches		

[1] If a SAR is calibrated along with the hydrological model, the SAR parameters are added at the end of the bestParam vector

(b) SAR results (optional)

Name	Description	Units	Size
SarResult.			
runoffD	SAR runoff depth	[mm]	$nTS \times 1$

4.4 Workflow

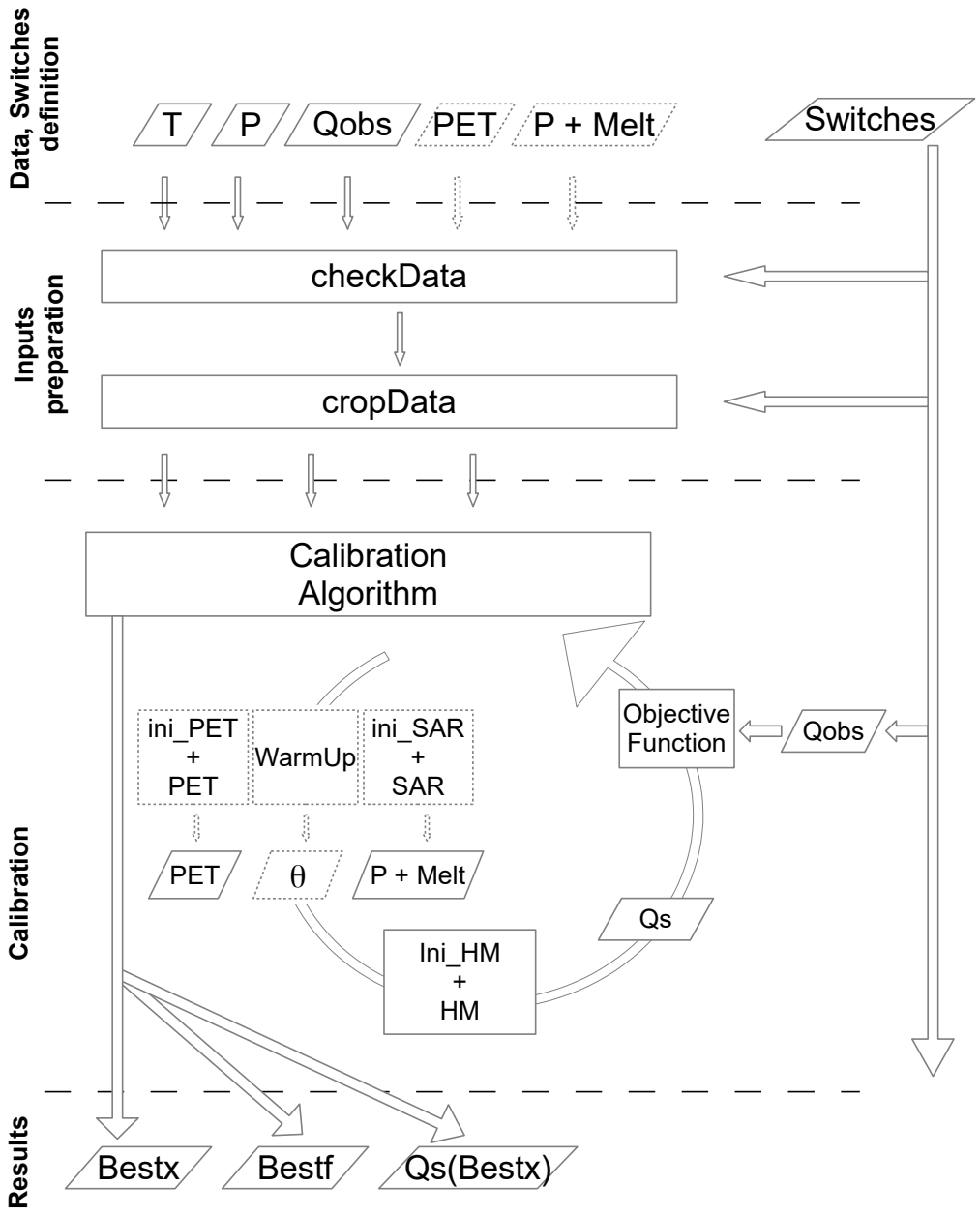

Figure 2: Workflow of the calibration. Boxes represent functions and parallelogram input/ouput. Dashed lines depict optional features.

5 Simulation

5.1 Data

The hydrometeorological data necessary for simulation are the same as for the calibration (see Sec 4.1). However, hydrological model (and optionally SAR) parameters must be provided from the calibrated_param.mat file located in the *HOOPLA/Data/<TimeStep>/Model_parameters* directory (Sec 4.3).

5.2 Switches

This section describes which switches are specifically needed to perform the simulations with HOOPLA. Please note the general switches presented in Table 4 also need to be set.

Table 8: User switches for simulation. Switches ending by ".on" can be set to only two values: 0 or 1, with 0=no/off/deactivate and 1=yes/on/activate.

(a) Simulation switches

Switches name	Description	Possible values	Advised values	Format
simulation.on	Run simulation	1, 0	-	Scalar
simStart	Beginning of simulation period[1]	-	-	Char.
simEnd	End of simulation period[1]	-	-	Char.

[1]The format for *simStart* and *simEnd* must be 'yyyy/mm/dd/HH:MM:SS' and can be alternatively 'yyyy/mm/dd' for a 24-hour time step. It is important that these dates coincide with the hydrological time step and the beginning of the day. For example, if one uses a 3h time step, one should define the start and the end date as a multiple of 3 (i.e 0am, 3am, 6am,..., 9pm). In the case one would use a 24h time step, the start and end date should be 0am (midnight).

(b) Data assimilation switches.

Switches name	Description	Possible values	Advised values	Format
DA.on	Perform data assimilation	1, 0	-	Scalar
DA.tech	Data assimilation scheme	EnKF, PF PertOnly[1]	-	Char.
DA.Uc_Q	Streamflow uncertainty [%]	-	0.1	Scalar
DA.Uc_Pt	Precipitation uncertainty [%]	-	0.5	Scalar
DA.Uc_Tpet	Temperature uncertainty [°C]	-	2	Scalar
DA.Uc_Tsno	Temperature uncertainty [°C]	-	2	Scalar
DA.Uc_Tmax	Max. temp. uncertainty [°C]	-	2	Scalar
DA.Uc_Tmin	Min. temp. uncertainty [°C]	-	2	Scalar
DA.Uc_E	PET uncertainty [%]	-	0.1	Scalar
DA.dt	Number of time steps between two state variable updates	-	[1-10]	Integer
DA.N	Ensemble size	-	[50 100]	Integer
DA.PF.ResampTech	Resampling technique	mult_res, syst_res[2]	-	Char.
DA.PF.resampleThresh	Effective sample size of the particles before resampling	[0 ∞]	∞	Char.

[1] EnKF, PF, and PertOnly refer to Ensemble Kalman Filter, Particle Filter, and (random) Perturbation Only (open loop simulation), respectively
[2] mult_res stands and syst_res stand for multinomial_resampling systematic_resampling respectively.

When resorting to data assimilation, all hydrometeorological input uncertainties must be specified by the user. For streamflow, precipitation, and PET, uncertainties are assumed to be proportional to the value and are expressed in terms of percentage of the observed quantity. For instance,

if $DA.Uc\_Q$ is set to 0.1, random sampling will be performed from a Gaussian distribution with a mean $\mu = Q$ and a standard deviation $\sigma = 0.1 \times Q$. For temperature, uncertainty is considered constant, and the standard deviation $DA.Uc\_T$ must be expressed in degrees. All distributions are assumed to be Gaussian except for precipitation uncertainty that is described by a gamma law. Finally, even if PertOnly is defined in the Switches as a possible data assimilation technique, it does not perform any type of updating, and should rather be seen as a possible benchmark. It only propagates estimated uncertainty defined by the switches, allowing a fairer and fully probabilistic assessment of data assimilation algorithms.

5.3 Results

Table 9 lists HOOPLA simulation outputs. The result file is named *Cwww_Hxxx_Eyyy_Szzz.mat*, where www, xxx, yyy, and zzz refer to the catchment, the hydrological model, the PET, and the SAR name, respectively. This file is situated in the *Result/Simulation* folder. The result files includes several arrays with named fields (Matlab structures) that contain the data of interest. For instance, the structure '*Result*' has a '*Qs*' field that holds the streamflow simulation. The content of this file may vary according to the specification of the switches. For example if the *Switches.SnowmeltCompute.on* is set to 1, additional results related to snow accounting will be recorded and saved. Also, matrices size may change depending on the data assimilation preferences. For example, if data assimilation is disabled, matrix Qs will have a $[nTS \times 1]$ size, while it will have a $[nTS \times nDA]$ size with nTS being the number of time steps and nDA the number of members/particles used in data assimilation. A demo set of results can be downloaded from https://github.com/AntoineThiboult/HOOPLA_Demo_Catchment_Result.

Table 9: Content of the result file for simulation. nTS is the number of time steps the simulation is performed on and nDA is the data assimilation ensemble size.

(a) Systematic results

Name	Description	Units	Size
DataSim	Structure that contains all hydrometeorological input data cropped on the simulation period	[-]	
Result.			
Qs	Simulated streamflow (with the parameter set contained in the file calibrated_param.mat)	[mm]	$nTS \times 1$ *or* $nTS \times nDA$
DateSim	Dates of the simulation	[-]	$nTS \times 6$

(b) SAR results (optional)

Name	Description	Units	Size
SarResult.	Structure that contains SAR results		
runoffD	SAR runoff depth	[mm]	$nTS \times 1$ *or* $nTS \times nDA$

5.4 Workflow

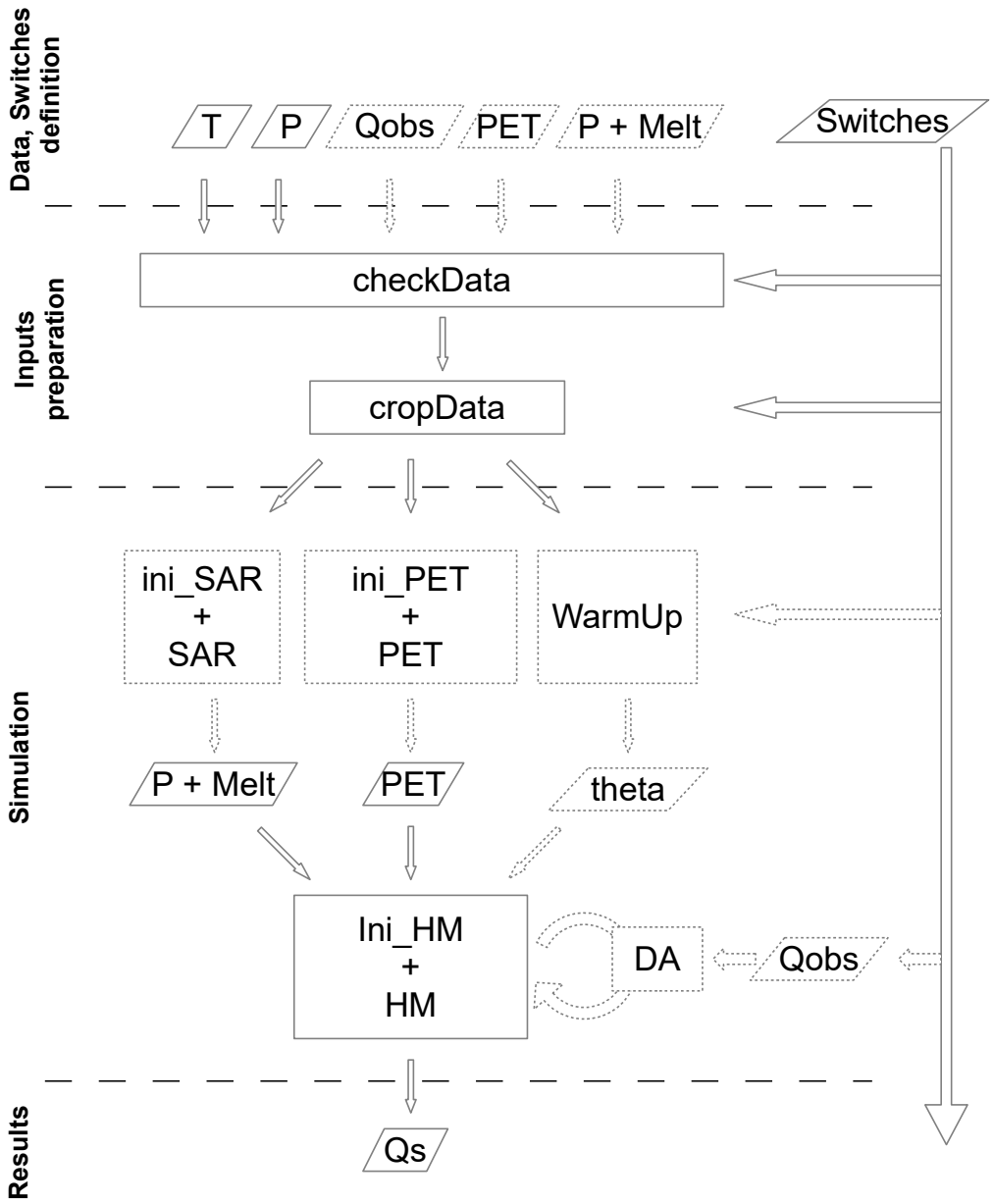

Figure 3: Workflow of the simulation. Boxes represent functions and parallelogram input/ouput. Dashed lines depict optional features. Note that observed streamflow is mandatory whenever data assimilation is to be performed.

6 Forecast

6.1 Data

6.1.1 Hydrometeorological data

The hydrometeorological data necessary to the forecast are the same as for the calibration (see Sec. 4.1). However, hydrological model (and SAR) parameters must be provided from the calibrated_param.mat file located in the *HOOPLA/Data/<TimeStep>/Model_parameters* directory (Sec. 4.3). Additionally, if 'real' forecast is requested (as opposed to 'perfect forecast' based on past observations), meteorological forecast, either deterministic (Sec. 6.1.2) or probabilistic (Sec. 6.1.3) must be provided and stored in *HOOPLA/Data/<TimeStep>/Det_met_fcast/* and *HOOPLA/Data/<TimeStep>/Ens_met_fcast/*, respectively.

Conversely, by setting the switch $Switches.forecast.perfectFcast.on$ to 1, meteorological observations will be used in place of the meteorological forecasts, which are then not needed.

6.1.2 Deterministic forecast data

This section indicates how deterministic meteorological forecasts should be formated to be used by HOOPLA. The file containing the deterministic forecast data should respect the following rules:

- The file should contain the data for a single catchment
- All necessary data for a catchment should be in a single file
- The file must be located in the *HOOPLA/Data/<TimeStep>/Det_met_fcast/* directory
- The name of the file should respect the format *Met_fcast_xxx.mat* where *xxx* is the catchment identifier defined by the user. Your catchment identifier should only contain letters and numerals (no accentuated letters nor any special character or punctuation).
- Please use flag '-v7.3' to save your meteorological forecast data. This allows improved performance in Matlab loadings and readings.

The content of a deterministic meteorological forecast datafile is summarized by Table 10, where nLT is the number of forecast lead times and nFI is the number of time a forecast is issued.

Table 10: Content of the Met_fcast datafile for deterministic ensemble forecast. nLT is the number of forecast lead times and nFI is the number of time a forecast is issued.

(a) Mandatory data

Name	Description	Units	Size
Date	Date of meteo. issue[1]	[-]	$nFI \times 6$
leadTime	Lead times	[day]	$1 \times nLT$
Pt	Total precipitation	[mm]	$nFI \times nLT$
PET	Potential evapotranspiration[2]	[mm]	$nFI \times nLT$

[1]The '*Date*' format corresponds to the datevec function in Matlab. First column contains the year, the second the month, the third the day, the fourth the hour, the fifth the minutes, the sixth the seconds, all stored as double.
[2]PET can also be computed internally by HOOPLA. In this case the user must provide the input data for the PET formula (see exemple for the Oudin formula below).

Note that the variable *Date* refers to the meteorological issue date (FI) and no longer to the variable *Date* in the *Hydromet_obs_xxx* file located in the *HOOPLA/Data/<Time Step>/Hydromet_obs* directory which refers to the hydrological time step dates. For illustration purpose, let's assume a typical example where a meteorological forecast is made available by an agency over a 365-day period, from the 1^{rst} of January until the 31^{rst} of December, twice a day (at 6 p.m. and 6 a.m.) with a 10 day horizon and a 3-hour time step. In such case $nFI = NumberOfDays \times IssuePerDay = 365 \times 2 = 700$. nLT is expressed as a fraction of day, so for a horizon ranging from 3 hour to 10 days, $nLT = [0.125, 0.25, ..., 9.875, 10]$. The *Date* matrix is then:

(b) CemaNeige data (optional)

Name	Description	Units	Size
Pt	Total precipitation	[mm]	$nFI \times nLT$
T	Air temperature	[°C]	$nFI \times nLT$
Tmax	Maximum temperature	[°C]	$nFI \times nLT$
Tmin	Minimum temperature	[°C]	$nFI \times nLT$

(c) PET data (optional, exemple for the Oudin formula)

Name	Description	Units	Size
Date	Date	-	$nMB \times nTS \times 6$
T	Air temperature	[°C]	$nFI \times nLT$

$$\text{Date} = \begin{array}{c} \\ Date_1 \\ Date_2 \\ Date_3 \\ \\ Date_{nFI} \end{array} \begin{array}{c} \begin{array}{cccccc} YYYY & MM & DD & hh & mm & ss \end{array} \\ \begin{pmatrix} 2015 & 01 & 01 & 06 & 00 & 00 \\ 2015 & 01 & 01 & 18 & 00 & 00 \\ 2015 & 02 & 02 & 06 & 00 & 00 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 2015 & 12 & 31 & 18 & 00 & 00 \end{pmatrix} \end{array}$$

If computations are made at a daily time step, the meteorological issue hour must be set to 00. The following matrix uses variable P (precipitation) to illustrate how it should be strucured. It has two axes, the Date (with a dimension nFI) and the lead time (with a dimension nLT). The other meteorological forecast variables are organized in the same way.

$$\text{P} = \begin{array}{c} \\ Date_1 \\ Date_2 \\ \vdots \\ Date_{nFI} \end{array} \begin{array}{c} \begin{array}{cccc} LT_1 & LT_2 & \cdots & LT_{nLT} \end{array} \\ \begin{pmatrix} P_{(Date_1+LT_1)} & P_{(Date_1+LT_2)} & \cdots & P_{(Date_1+LT_{nLT})} \\ P_{(Date_2+LT_1)} & P_{(Date_2+LT_2)} & \cdots & P_{(Date_2+LT_{nLT})} \\ \vdots & \vdots & \ddots & \vdots \\ P_{(Date_{nFI}+LT_1)} & P_{(Date_{nFI}+LT_2)} & \cdots & P_{(Date_{nFI}+LT_{nLT})} \end{pmatrix} \end{array}$$

A demo data file for deterministic meteorological forecast is provided in the *HOOPLA/-Data/<TimeStep>/Det_met_fcast/ directory*. The user is encouraged to take a close look at the demo files format.

6.1.3 Probabilistic forecast data

This section indicates how probabilistic (ensemble) meteorological forecast data should be formated to be processed by HOOPLA. The file should respect the following rules:

- The file should contain the data for a single catchment
- All necessary data for a catchment should be in a single file
- The file must be located in the *HOOPLA/Data/<TimeStep>/Ens_met_fcast/* directory
- The name of the file should respect the format *Met_fcast_xxx.mat* where *xxx* is the catchment identifier defined by the user. Your catchment identifier should only contain letters and numerals (no accentuated letters nor any special character or punctuation).
- Please use flag '-v7.3' to save your meteorological forecast data. This allows improved performance in Matlab loadings and readings.

The content of an ensemble meteorological forecast datafile is summarized Table 11. The organization of the file is quite similar to the one used for deterministic forecast, except that all variables are stored within a structure named '*Met_fcast*'. This indexed structure has a dimension $1 \times nMM$ with nMM (the index) being the number of meteorological members and has

field names according to the variable names. For example, temperature of the iMM$^{\text{th}}$ meteorological member can be accessed by $Met\_fcast(iMM).T$. In Table 11, nLT is the number of forecast lead times and nFI is the number of time a forecast is issued. **As for deterministic meteorological forecasts, the variable** *Date* **refers to the meteorological issue date and no longer to the variable** *Date* **in the *Hydromet_obs_xxx* file located in the *HOOPLA/Data/<Time Step>/Hydromet_obs* directory, which refers to the hydrological time step dates.**

Table 11: Content of the Met_fcast datafile for ensemble forecast. nLT is the number of forecast lead times, nFI is the number of time a forecast is issued, and nMM the number of meteorological members.

(a) Mandatory data

Name	Description	Units	Size
Met_fcast.			$1 \times nMM$
Date	Date of meteo. issue[1]	[-]	$nFI \times 6$
leadTime	Lead times	[day]	$1 \times nLT$
Pt	Total precipitation	[mm]	$nFI \times nLT$
PET	Potential evapotranspiration[2]	[mm]	$nFI \times nLT$

[1]The '*Date*' format corresponds to the datevec function in Matlab. First column contains the year, the second the month, the third the day, the fourth the hour, the fifth the minutes, the sixth the seconds, all stored as double.
[2]PET can also be computed internally by HOOPLA. In this case the user must provide the input data for the PET formula (see example for the Oudin formula below).

(b) CemaNeige data (optional)

Name	Description	Units	Size
Met_fcast.			$1 \times nMM$
Pt	Total precipitation	[mm]	$nFI \times nLT$
T	Air temperature	[°C]	$nFI \times nLT$
Tmax	Maximum temperature	[°C]	$nFI \times nLT$
Tmin	Minimum temperature	[°C]	$nFI \times nLT$

(c) PET data (optional, example for the Oudin formula)

Name	Description	Units	Size
Met_fcast.			$1 \times nMM$
Date	Date	-	$nMB \times nTS \times 6$
T	Air temperature	[°C]	$nFI \times nLT$

The matrix containing the data for each meteorological member follows the same logic as for the deterministic meteorological forecast (Sec 6.1.2). For illustration purposes, let's assume the same typical example as in Sec 6.1.2 where a meteorological forecast is made available by an agency over a 365-day period, from the 1$^{\text{rst}}$ of January until the 31$^{\text{rst}}$ of December, twice a day (at 6 p.m. and 6 a.m.) with a 10 day horizon and a 3-hour time step. In such case $nFI = NumberOfDays \times IssuePerDay = 365 \times 2 = 700$. nLT is expressed as a fraction of day, so for a horizon ranging from 3 hour to 10 days, $nLT = [0.125, 0.25, ..., 9.875, 10]$. Thus the matrix containing the dates for the 15$^{\text{th}}$ meteorological member is:

$$\text{Met\_fcast(15).Date} = \begin{array}{c} \\ Date_1 \\ Date_2 \\ Date_3 \\ \\ Date_{nFI} \end{array} \begin{array}{c} \begin{array}{cccccc} YYYY & MM & DD & hh & mm & ss \end{array} \\ \begin{pmatrix} 2015 & 01 & 01 & 06 & 00 & 00 \\ 2015 & 01 & 01 & 18 & 00 & 00 \\ 2015 & 02 & 02 & 06 & 00 & 00 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 2015 & 12 & 31 & 18 & 00 & 00 \end{pmatrix} \end{array}$$

Of course the dates for all other members will be identical to these of meteorological member 15 but the matrix need to be repeated for each member in order for HOOPLA to process the data.

The following matrix illustrates this point for variable P (precipitation) and for the 15^{th} meteorological member. The matrix has two axes, the Date (with a dimension nFI) and the lead time (with a dimension nLT). The other meteorological forecast variables are organized in the same way.

$$\text{Met\_fcast(15).P} = \begin{array}{c} \\ Date_1 \\ Date_2 \\ \vdots \\ Date_{nFI} \end{array} \begin{array}{c} \begin{array}{cccc} LT_1 & LT_2 & \cdots & LT_{nLT} \end{array} \\ \begin{pmatrix} P_{(Date_1+LT_1)} & P_{(Date_1+LT_2)} & \cdots & P_{(Date_1+LT_{nLT})} \\ P_{(Date_2+LT_1)} & P_{(Date_2+LT_2)} & \cdots & P_{(Date_2+LT_{nLT})} \\ \vdots & \vdots & \ddots & \vdots \\ P_{(Date_{nFI}+LT_1)} & P_{(Date_{nFI}+LT_2)} & \cdots & P_{(Date_{nFI}+LT_{nLT})} \end{pmatrix} \end{array}$$

A demo ensemble forecast data file is provided in the *HOOPLA/Data/<TimeStep>/Ens_met_fcast/* directory. The user is encouraged to take a close look at the demo files format.

6.2 Switches

This section describes which switches (Table 12) to set in order to perform hydrological forecasts with HOOPLA. Please note the general switches presented in Table 4 need to be set as well.

Table 12: User switches for forecast. Switches ending by ".on" can be set to only two values: 0 or 1, with 0=no/off/deactivate and 1=yes/on/activate.

(a) Forecast switches

Switches name	Description	Possible values	Advised values	Format
$fcast.on$	Run forecast	1, 0	-	Scalar
$fcastStart$	Beginning of forecast period	-	-	Char.
$fcastEnd$	End of forecast period	-	-	Char.
$issueTime$	Hour of the day for which a forecast is issued	one or several times per day	-	Scalar or vector
$perfectFcast.on$	Use meteorological observations as meteorological forecast	1, 0	-	Scalar
hor	Horizon of the forecast (in forecasting time steps)	-	-	Scalar
$metEns.on$	Use meteorological ensemble forecast	1, 0	-	Scalar

Similarly to simulation (Sec. 5.2), when resorting to data assimilation, all hydrometeorological input uncertainties must be specified by the user. For the streamflow, precipitation, and PET, uncertainties are assumed to be proportional to the value and are expressed in terms of percentage of the observed quantity. For instance, if $DA.Uc\_Q$ is set to 0.1, random sampling will be performed from a Gaussian distribution with a mean μ = Q and a standard deviation $\sigma = 0.1\times$ Q. For temperature, uncertainty is considered static, and the standard deviation $DA.Uc\_T$ must be expressed in degrees. All distributions are assumed to be Gaussian except for precipitation uncertainty that follows a gamma law. Finally, even if PertOnly is defined in the Switches as a possible data assimilation technique, it does not perform any type of updating, but should rather be seen as a possible benchmark. It only propagates estimated uncertainty defined by the switches, allowing a fairer and fully probabilistic assessment of data assimilation algorithms.

A hydrological forecast can be issued at any time of the day by setting $Switches.issueTime$. Nonetheless, this time should be specified in accordance with the modeling time step (see Table 4) and (obviously) to the availability of the meteorological forecast. For example, if this switch is set to '3h', the issue time must be a multiple of 3, i.e., 0, 3, 6,..., 21. Additionally, it is possible to issue several hydrological forecast per day. Let's assume that a meteorological forecast is made

(b) Data assimilation switches.

Switches name	Description	Possible values	Advised values	Format
DA.on	Perform data assimilation	1, 0	-	Scalar
DA.tech	Data assimilation scheme	EnKF, PF, PertOnly[1]	-	Char.
DA.Uc_Q	Streamflow uncertainty [%]	-	0.1	Scalar
DA.Uc_Pt	Precipitation uncertainty [%]	-	0.5	Scalar
DA.Uc_Tpet	Temperature uncertainty [°C]	-	2	Scalar
DA.Uc_Tsno	Temperature uncertainty [°C]	-	2	Scalar
DA.Uc_Tmax	Max. temp. uncertainty [°C]	-	2	Scalar
DA.Uc_Tmin	Min. temp. uncertainty [°C]	-	2	Scalar
DA.Uc_E	PET uncertainty [%]	-	0.1	Scalar
DA.dt	Number of time step between two state variable updating	-	[1-10]	Integer
DA.N	Ensemble size	-	[50 100]	Integer
DA.PF.ResampTech	Resampling technique	mult_res, syst_res[2]	-	Char.
DA.PF.resampleThresh	Effective sample size of particles before resampling	[0 ∞]	-	Char.

[1] EnKF, PF, and PertOnly refer to Ensemble Kalman Filter, Particle Filter, and (random) Perturbation Only, respectively
[2] mult_res stands and syst_res stand for multinomial_resampling systematic_resampling respectively.

available twice a day, at 6am and 6pm, then two hydrological forecasts will be issued providing that the switch issue time is defined as follows: $Switches.issueTime = [6, 18]$. For a daily time step (*Switches.timeStep*='24'), *Switches.issueTime* should be set to 0.

6.3 Results

Table 13 lists HOOPLA's forecast outputs. The result file is named *Cwww_Mxxx_Eyyy_Szzz.mat*, where *www*, *xxx*, *yyy*, and *zzz* refer to the catchment, the model, the PET, and the SAR number or name, respectively. This file is located in the *HOOPLA/Result/Forecast* folder. The result files includes several arrays with named fields (Matlab structures) that contain the data of interest. For instance, the structure *Result* has a $Qfcast$ field that contains the streamflow forecast. The content of this file may vary according to the specification of the switches. For example if the *Switches.SnowmeltCompute.on* is set to 1, additional results related to snow accounting will be recorded and saved.

Table 13: Content of the result file for forecast. nTS is the number of time steps the forecast is performed on, nLT is the number of forecast lead times, nMM is the number of meteorological members, nDA is the number of data assimilation members/particles.

(a) Systematic results

Name	Description	Units	Size
DataFcast.	Structure containing all meteorological forecasts	[-]	
Switches.	Structure containing all user switches	[-]	
Result.			
Qs	Simulated streamflow	[mm]	Varying
$Qfcast$	Forecasted streamflow	[mm]	Varying
$DateFcast$	Validity time	[-]	$nTS \times nLT$

(b) SAR results (optional)

Name	Description	Units	Size
SarResult.	Structure that contains SAR results		
runoffD	SAR runoff depth	[mm]	Varying

Matrices size may change depending on the data assimilation settings and the type of meteorological forecast preferences. The generic size of the streamflow and runoff depth matrices are:

- *Result.Qfcast*: $(nMM \times nTS \times nLT \times nDA)$
- *Result.Qs*: $(nMM \times nTS \times nDA)$
- *SarResult.runOffD*: $(nMM \times nTS \times nDA)$

However, some dimensions may disappear if not required. For instance, if deterministic meteorological forecast is used (i.e. ensemble meteorological not used and $nMM = 1$), *Result.Qfcast* size will be $(nTS \times nLT \times nDA)$. Similarly, if data assimilation or ensemble meteorological forecasting are not used, then *Result.Qfcast* size will be reduced to $(nTS \times nLT)$. This also applies for the *Result.Qs* and the *SarResult.runOffD* matrices. The matrix *Result.Qfcast* contains the hydrological forecast results. To interpret it, let's take an example with deterministic meteorological forecast and no data assimilation. The *Result.Qfcast* matrix would then have a size of $nTS \times nLT$.

Once more, it is important to note that the hydrological dates and the meteorological forecast issue dates may not be identical. Here, we consider a case where a meteorological forecast is made available every 4 hydrological time steps, as it would be the case of a 3-hour time step and a meteorological forecast issued twice a day, i.e. every 12 hours. As an example, let us assume that the hydrological dates ($Date_{TS_1}$, $Date_{TS_2}$...) are 1$^{\text{rst}}$ January, 6am, 1$^{\text{rst}}$ January, 9am and the meteorological forecast dates ($Date_{FI_1}$ and the $Date_{FI_2}$...) are 1$^{\text{rst}}$ January, 6am, 1$^{\text{rst}}$ January, 6pm, etc. A hydrological forecast will be issued by HOOPLA at $Date_{TS_1}$ because a meteorological forecast is available at this time. However, no hydrological forecast will be issued at $Date_{TS_2}$ because there is no corresponding meteorological forecast. Therefore, the corresponding line in the *Result.Qfcast* matrix would be filled with NaN (not a number). The matrix *Result.Qfcast* could then be schematically represented like this:

$$\begin{array}{cc} & \\ Date_{FI_1} & Date_{TS_1} \\ & Date_{TS_2} \\ & Date_{TS_3} \\ & Date_{TS_4} \\ Date_{FI_2} & Date_{TS_5} \\ & Date_{TS_6} \\ \vdots & \vdots \\ Date_{nFI} & Date_{nTS} \end{array} \begin{array}{c} \begin{array}{cccc} LT_1 & LT_2 & \cdots & LT_{nLT} \end{array} \\ \begin{pmatrix} Qfcast_{(Date_1+LT_1)} & Qfcast_{(Date_1+LT_2)} & \dots & Qfcast_{(Date_1+LT_{nLT})} \\ NaN & NaN & \dots & NaN \\ NaN & NaN & \dots & NaN \\ NaN & NaN & \dots & NaN \\ Qfcast_{(Date_5+LT_1)} & Qfcast_{(Date_5+LT_2)} & \dots & Qfcast_{(Date_5+LT_{nLT})} \\ NaN & NaN & \dots & NaN \\ \vdots & \vdots & \ddots & \vdots \\ Qfcast_{(Date_{nTS}+LT_1)} & Qfcast_{(Date_{nTS}+LT_2)} & \dots & Qfcast_{(Date_{nTS}+LT_{nLT})} \end{pmatrix} \end{array}$$

The variable *Result.Date_fcast* may help the user to confirm the validity time of the hydrological forecast for each element of the Result.Qfcast matrix. It is a also a $nTS \times nLT$ matrix, like the *Result.Qfcast* matrix and contains dates stored as numerical values (see the Matlab documentation of `datenum`).

$$\text{Result.Date\_fcast} = \begin{array}{c} \begin{array}{cccc} LT_1 & LT_2 & \cdots & LT_{nLT} \end{array} \\ \begin{pmatrix} Date_1 + LT_1 & Date_1 + LT_2 & \dots & Date_1 + LT_{nLT} \\ Date_2 + LT_1 & Date_2 + LT_2 & \dots & Date_2 + LT_{nLT} \\ \vdots & \vdots & \ddots & \vdots \\ Date_{nFI} + LT_1 & Date_{nFI} + LT_2 & \dots & Date_{nFI} + LT_{nLT} \end{pmatrix} \end{array} \begin{array}{c} \\ Date_1 \\ Date_2 \\ \vdots \\ Date_{nFI} \end{array}$$

By applying either the Matlab function `datestr` or `datevec` to any element in $Result.Date\_fcast$, it is straightforward to verify the validity time of any corresponding element in the Result.Qfcast matrix. A demo set of results can be downloaded from https://github.com/AntoineThiboult/HOOPLA_Demo_Catchment_Result.

6.4 Workflow

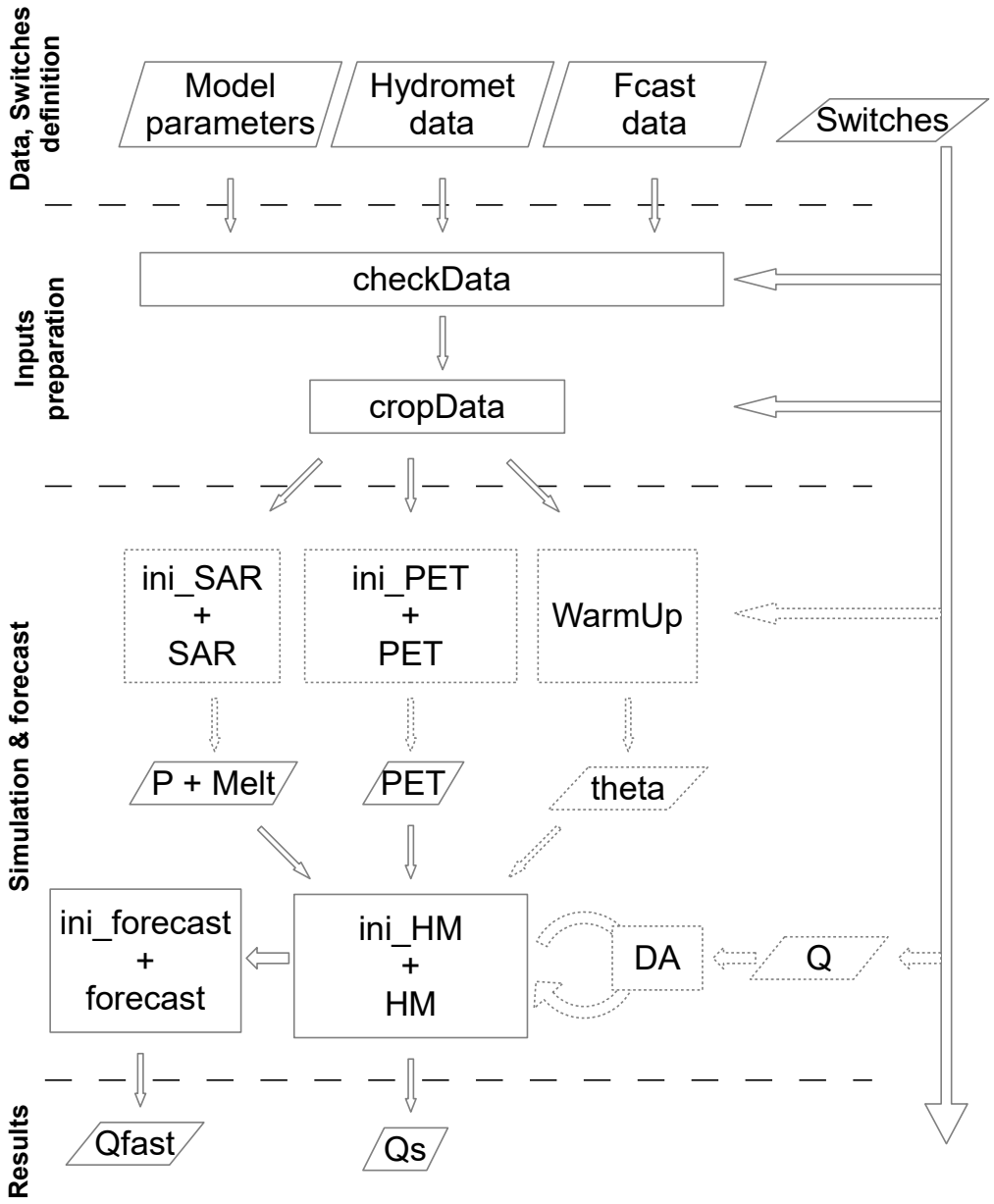

Figure 4: Workflow of the forecast. Boxes represent functions and parallelogram input/ouput. Dashed lines indicate optional features.

7 Expanding HOOPLA functionalities

If you read this, it means that you already have some knowledge and experience with HOOPLA. We strongly recommend being comfortable with using HOOPLA before adding extensions to the toolbox. This section doesn't provide you with an exhaustive list of modifications to perform but rather a guideline to key features that one may want to modify and to what should be preserved. Remember that all functions are heavily commented, and that input and output as well as the function task are clearly listed in every function header.

If you developed some additional functionalities, added extra models, etc., and wish to share them with the community, first thank you, that is much appreciated! Second, you can contact hoopla@fsg.ulaval.ca and we will examine your work for a possible implementation in the official version. You will obviously be mentioned as external contributor.

HOOPLA was developed as a framework that connects many modules (hydrological models, PET formulation, etc.) together. To add new features, one needs to do only three things:

1. Respect input and output conventions that are inherent to the feature type
2. Drop your file in the appropriate repository
3. Edit the appropriate configuration file to inform HOOPLA that a new feature is available

7.1 Hydrological models

Hydrological models are decomposed into two functions: one for the model initialization and one for the model per se. You are free to name your model as you wish as long as you respect Matlab naming restrictions. Here, let's consider a case where the hydrological model is name myHydroModel. The two functions that must be defined are *myHydroModel.m* (the model itself) and *ini_ myHydroModel.m* (the initialization function).

myHydroModel.m

Hydrological models are lumped rainfall-runoff models only. Therefore, the inputs to the `myHydroModel` function are limited to:

- P: precipitation
- E: Potential evapotranspiration
- $Param$: Structure that contains all fields necessary to run the model, typically the state variables and the model parameters

Similarly, the outputs must be:

- Qs: Simulated streamflow
- $Param$: Structure that contains all fields necessary to run the model, typically the state variables (updated by the modeling step) and the model parameters, etc.
- $varargout$: This is a Matlab function that allows to have flexible outputs. Varargout is triggered by the switch $Switches.exportLight$. When the switch is set to 0, varargout will be returned as output. The user may choose this variable to return and save any intermediate value within the hydrological model.

ini_myHydroModel.m

This function is necessary for model initialization, i.e. to define the value of initial state variables (before model warm up), and to preallocate the matrices. The name of the function should be the same as the model name preceded by ini_.

The inputs of `ini_myHydroModel` are:

- $Switches$: Structure that contains all calibration/simulation/forecast options
- $Date$: The calibration/simulation/forecast dates (used for matrices preallocation)

- x: the hydrological model parameters

The outputs of `ini_myHydroModel` are:

- *Result*: Structure that contains the preallocated matrices that will be filled with the outputs of the myHydroModel function
- *Param*: Structure that contains all fields necessary to run the model, typically the state variables and the model parameters

The files *myHydroModel.m* and *ini_ myHydroModel.m* should be placed in the *HOOPLA/Tools/HydroModels/<YourModelName>* directory (as also shown Figure 1).

Listing your model

To enable HOOPLA to use any added hydrological model, the file *hydro_ model_ names.mat* located in *HOOPLA/Data/<TimeStep>/Misc* directory should be edited. The file contains a cell array named *nameM* of size $nM \times 2$, where nM refers to the number of hydrological models available (twenty in the current version of HOOPLA). The first column contains the name of the hydrological model while the second lists the necessary data to run this hydrological model separated by underscores, both stored as a string. For instance, if the HydroMod1 model is used, the first column contains 'HydroMode1' while the second contains 'Pt_E'. Append the name of your model to the array as a string and save the file to make it available for computation with HOOPLA.

If you wish to calibrate your newly imported model with the Schuffle Complex Evolution or the Dynamically Dimensioned Search algorithm, you need to specify its parameter boundaries in the file *model_ param_ boundaries.mat* located in the *HOOPLA/Data/<TimeStep>/Model_ parameters* directory. The file contains multiple structures, one for each hydrological model. To specify the parameter boundaries, a structure must be created and named by your model name. This structure must contain three fields, sMin, sIni, and sMax, which designate the minimum, the initial, and the maximum parameter value respectively. These field have a $1 \times nParam$ size, where $nParam$ represents the number of parameters.

Lastly, to use data assimilation, it is necessary to indicate what state variable should be updated. This is done by editing the *reservoir_ to_ update.mat* file located in the *HOOPLA/Data/<TimeStep>/Misc* directory. This file contains multiple cell arrays, one for each model, named after the model names. To specify the reservoir that should be updated, a cell array named as your model must be created and contain the name (stored as a string variable) of the reservoirs you wish to update.

7.2 Potential evapotranspiration formula

Potential evapotranspiration calculations are performed by two functions, one which computes the prerequisite values that are necessary for the computation initialization and one that performs the calculations. Once more, you are free to name your potential evapotranspiration formula as you wish as long as you respect Matlab conditions. Here, let's consider a case where the PET formula is name myPET. The two new functions must then be defined as *myPET.m* (the formula itself) and *ini_ myPET.m* (computation of the prerequisites values).

myPET.m

Because PET formulas rely on a variety of inputs, to provide flexibility, they are embedded in a structure named *PetData* that may hold any number of fields (which foster the variables) that are necessary to the PET calculations as coded in `myPET` function. The only output of the function is E, the potential evapotranspiration value.

ini_ myPet.m

This function prepares the inputs for the *myPet.m* function. It accepts Switches (all specified options about required computations) and Data (meteorological data necessary for PET calculation). The output is *PetData*, that contains input for *myPet.m* function.

The files *myPET.m* and *ini_ myPET.m* should be placed in the *HOOPLA/Tools/PET/<my_ PET_ Name>* directory (as also shown Figure 1).

Listing your PET formula

To enable HOOPLA to use the added PET formula, the file *pet_model_names.mat* located in *HOOPLA/Data/<TimeStep>/Misc* directory should be edited. The file contains a cell array named *nameE* of size $nE \times 3$, where nE refers to the number of PET formulas available. The first column contains the name of the formula. The second and third columns hold data that are necessary to compute the PET with the corresponding formula. The second lists all the temporal data (such as temperature, solar radiation, etc.), and the third one lists the catchment constants (latitude, elevation, etc). The content of the second and third columns must be stored as a string, with the different data name separated by underscores. For instance, in the case of the Oudin formula, date, mean temperature and the catchment latitude are required to compute the PET. The first columns of *nameM* contains 'Oudin', the second, 'Date_T', and the third, 'Lat'. To make your PET formula available to HOOPLA, append the name of your model and the necessary variable names to the array as a string and save the file.

7.3 Snow accounting routine

In the current HOOPLA version, input and output of the snow accounting routine (SAR) are not very flexible: Precipitation and temperature(s) are used as inputs and the only output is the runoff depth (the sum of liquid precipitation and melt). This issue is likely to be improved in a future release of HOOPLA. Within HOOPLA, snow accounting routine functions are very similar to the hydrological models in the sense that HOOPLA asks for a function to initialize the SAR and the SAR function itself. Once more you are free to name your model as you wish as long as you respect Matlab naming restrictions. Here, let's consider a case for wich the SAR is named mySAR. The two functions that must be defined are *mySAR.m* (the model itself) and *ini_mySAR.m* (the initialization function).

mySAR.m

The input of the SAR must be the following ones:

- P: Precipitation
- T: Mean temperature
- $Tmax$: Maximum temperature
- $Tmin$: Minimum temperature
- $Date$: Date
- $SarParam$: Structure that contains all relevant fields necessary to run the model. For instance, the state variable, the calibrated parameters, etc.

Similarly, the outputs must be:

- $runoffD$: Simulated streamflow
- $SarParam$: Structure that contains all relevant fields necessary to run the model. For instance, the state variable, the calibrated parameters, etc.
- $varargout$: This is a Matlab function that allows to have flexible outputs. Varargout is triggered by the switch $Switches.exportLight$. When the switch is set to 0, varargout will be returned as output. The user may choose this variable to return and save any intermediate value within the SAR model.

ini_mySAR.m

This function is necessary for SAR initialization, i.e. to define initial state variables values (before model warm up), and to preallocate result matrices. The name of the function should be the same as the SAR name preceded by ini_.

Inputs are:

- $Switches$: Structure that contains all calibration/simulation/forecast options

- *Data*: The data relevant for the SAR
- x: the parameters vector, that contains both the hydrological model parameters and SAR parameters (the SAR parameters are assumed to be located at the end of the parameter set)

Outputs are:

- *SarResult*: Structure that contains matrices that will, in turn, contain the outputs of the `mySAR` function
- *SarParam*: Structure that contains all relevant fields necessary to run the model, typically the state variable and the hydrological model and SAR parameters.

The files *mySAR.m* and *ini_ mySAR.m* should be placed in the *HOOPLA/Tools/Snow/ <my_ SAR_ Name>* directory (as also shown Figure 1).

Listing your SAR

To enable HOOPLA to use any new SAR models, the file *snow_ model_ names.mat* located in *HOOPLA/Data/<TimeStep>/Misc* directory should be edited. The file contains a cell array named *nameS* of size $nS \times 3$, where nS refers to the number of SAR models available. The first column contains the name of the model. The second and third columns hold data that are necessary to compute the snowmelt with the corresponding SAR. The second lists all the temporal data (such as temperature, precipitation, etc.), and the third one lists the catchment constants (latitude, elevation, etc). The content of the second and third columns must be stored as a string, with the different data name separated by underscores. For instance, if CemaNeige is used, the first column contains 'CemaNeige', the second contains 'Pt_T', and the third 'Beta_gradT_QNBV_Vmin_Zz5'. To make your SAR available to HOOPLA, append the name of your model and the necessary variables names to the array as a string and save the file.

If you wish to calibrate your SAR with the Schuffle Complex Evolution or the Dynamically Dimensioned Search algorithm, you need to specify its parameter boundaries in the file *snow_ param_ boundaries.mat* located in the *HOOPLA/Data/<TimeStep>/Model_ parameters* directory. The file contains multiple structures, one for each SAR. To specify the parameter boundaries, a structure must be created and named by your SAR name. This structure must contain four fields, $sMin$, $sIni$, $sMax$, and default, which designate the minimum, the initial, the maximum parameter value, and the default parameter values, respectively. The default value is used if the user chooses not to calibrate the SAR but only the hydrological model parameters ($Switches.calibration.snowCal.on$ set to 0). These field have a $1 \times nParam$ size, where $nParam$ represents the number of SAR parameters.

8 Acknowledgments

We acknowledge the NSERC Canadian Floodnet initiative, the Chaire de recherche EDS en prévisions et actions hydrologiques, and the Natural Sciences and Engineering Research Council of Canada for funding the research that allowed the creation of this toolbox. We would like to warmly thank the Ministère du Développement durable, de l'Environnement et de la Lutte contre les changements climatiques, Direction de l'information sur le milieu atmosphérique and the Direction de l'Expertise Hydrique for providing us (and you!) with the demo meteorological and hydrological[1] data included in the toolbox. Thanks to ECMWF for the development and maintenance of the TIGGE data portal, which provides free access to meteorological ensemble forecasts to the scientific community worldwide, and allowed us to include the demo meteorological forecast to HOOPLA. Last but not least, we are grateful to the IRSTEA catchment hydrology team in Antony (France) for sharing their knowledge of the 20 lumped models used in the toolbox, the Yarpiz project [2] for the SCE code, as well as Bryan Tolson for the DDS code.

[1] The hydrological data are publicly available here.

[2] www.yarpiz.com. They provide many other great pieces of code about artificial intelligence, machine learning, engineering optimization,...

References

M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Ieee Transactions on Signal Processing*, 50(2):174–188, 2002. ISSN 1053-587X. doi: 10.1109/78.978374.

S. Bergström and A. Forsman. Development of a conceptual deterministic rainfall-runoff model. *Nordic Hydrology*, 4:147–170, 1973.

K. J. Beven, M. J. Kirkby, N. Schofield, and A. F. Tagg. Testing a physically-based flood forecasting model (topmodel) for 3 UK catchments. *Journal of Hydrology*, 69:119–143, 1984. doi: 10.1016/0022-1694(84)90159-8.

R. J. C. Burnash, R. L. Ferral, and R. A. McGuire. *A generalized streamflow simulation system - Conceptual modelling for digital computers.* 1973.

F. H. S Chiew, M. C. Peel, and A. W. Western. *Application and testing of the simple rainfall-runoff model SIMHYD*, pages 335–367. 2002.

Y. Cormary and A Guilbot. Étude des relations pluie-débit sur trois bassins versants d'investigation. In *Design of water resources projects with inadequate data: IAHS Madrid Symposium*, pages 265–279, 1973.

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

Q. Y. Duan, S. Sorooshian, and V. Gupta. Effective and efficient global optimization for conceptual rainfall-runoff models. *Water Resources Research*, 28(4):1015–1031, 1992. ISSN 0043-1397. doi: 10.1029/91wr02985.

G. Evensen. The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53:343–367, 2003. doi: 10.1007/s10236-003-0036-9.

V. Fortin and R Turcotte. Le modèle hydrologique mohyse, note de cours pour SCA7420. Report, Département des sciences de la terre et de l'atmosphère, Université du Québec à Montreal, 2007.

R. Garçon. Modèle global pluie-débit pour la prévision et la prédétermination des crues. *La Houille Blanche*, 7/8:88–95, 1999. doi: http://dx.doi.org/10.1051/lhb/1999088.

G. Girard, G Morin, and R. Charbonneau. Modèle précipitations-débits à discrétisation spatiale. *Cahiers ORSTOM, Série Hydrologie*, 9:35–52, 1972.

A. J. Jakeman, I. G. Littlewood, and P. G. Whitehead. Computation of the instantaneous unit hydrograph and identifiable component flows with application to two small upland catchments. *Journal of Hydrology*, 117:275–300, 1990. doi: 10.1016/0022-1694(90)90097-h.

N. Kharrufa. Simplified equation for evapotranspiration in arid regions. *Beiträge zur Hydrologie*, 5:1:39–47, 1985.

J. Mandel. Efficient implementation of the ensemble kalman filter. Report, UCDHSC, 2006.

T. Mathevet. *Quels modèles pluie-débit globaux au pas de temps horaire? Développements empiriques et comparaison de modèle sur un large échantillon de bassins versants.* PhD thesis, École Natl. du Génie Rural des Eaux et des For., Paris., 2005.

B. Mazenc, M. Sanchez, and D Thiery. Analyse de l'influence de la physiographie d'un bassin versant sur les paramètres d'un modèle hydrologique global et sur les débits caractéristiques à l'exutoire. *Journal of Hydrology*, 69:97–188, 1984.

R. J. Moore and R. T. Clarke. A distribution function approach to rainfall runoff modeling. *Water Resources Research*, 17:1367–1382, 1981. doi: 10.1029/WR017i005p01367.

S. A. Nielsen and E. Hansen. Numerical simulation of the rainfall-runoff process on a daily basis. *Nordic Hydrology*, 4:171–190, 1973.

P. E. O'Connell, J. E. Nash, and J. P Farrell. River flow forecasting through conceptual models, part II - the Brosna catchment at Ferbane. *Journal of Hydrology*, 10:317–329, 1970.

L. Oudin, F. Hervieu, C. Michel, C. Perrin, V. Andreassian, F. Anctil, and C. Loumagne. Which potential evapotranspiration 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. doi: 10.1016/j.jhydrol.2004.08.026.

H. L. Penman. Natural evaporation from open water, bare soil and grass. In Royal Society of London, editor, *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences.*, volume 193, pages 120–145, 1948.

C. Perrin. *Vers une amélioration d'un modèle global pluie-débit*. Thesis, 2000.

G. Seiller, F. Anctil, and C. Perrin. Multimodel evaluation of twenty lumped hydrological models under contrasted climate conditions. *Hydrology and Earth System Sciences*, 16(4):1171–1189, 2012. ISSN 1027-5606. doi: 10.5194/hess-16-1171-2012.

M. Sugawara. Automatic calibration of the tank model. *Hydrological Sciences*, 24:375–388, 1979.

A. Thiboult and F. Anctil. On the difficulty to optimally implement the ensemble kalman filter: An experiment based on many hydrological models and catchments. *Journal of Hydrology*, 529: 1147–1160, 2015.

A. Thiboult, F. Anctil, and M.H. Ramos. How does the quantification of uncertainties affect the quality and value of flood early warning systems? *Journal of Hydrology*, 551:365 – 373, 2017. ISSN 0022-1694. doi: https://doi.org/10.1016/j.jhydrol.2017.05.014.

A. Thiboult, G. Seiller, C. Poncelet, and F. Anctil. The hoopla toolbox: a hydrological prediction laboratory - manuscript submitted for publication. *Environmental Modelling & Software*, 2018.

D. Thiery. Utilisation d'un modèle global pour identifier sur un niveau piézométrique des influences multiples dues à diverses activités humaines. In *Improvement of methods of long term prediction of variations in groundwater resources and regimes due to human activity*, volume 136, pages 71–77, 1982.

C. W. Thornthwaite and J. R. Mather. The water balance. Report, Drexel Institute of Climatology, 1955.

B. A. Tolson and C. A. Shoemaker. Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. *Water Resources Research*, 43(1):n/a–n/a, 2007. ISSN 1944-7973. doi: 10.1029/2005WR004723. W01413.

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. doi: https://doi.org/10.1016/j.jhydrol.2014.04.058.

T. Wagener, D. P. Boyle, M. J. Lees, H. S. Wheater, H. V. Gupta, and S. Sorooshian. A framework for development and application of hydrological models. *Hydrology and Earth System Sciences*, 5:13–26, 2001.

P.M.M. Warmerdam, J. Kole, and J. Chormanski. Modelling rainfall-runoff processes in the hupselse beek research basin. In *IHP-V, Technical Documents in Hydrology*, pages 155–160, 1997.

R. J. Zhao, Y. L. Zuang, L. R. Fang, X. R. Liu, and Q.S. Zhang. The xinanjiang model. *IAHS Publications*, 129:351–356, 1980.