Introduction to the HBV model

Matteo Giuliani, Jon Herman

November 9, 2017

Introduction to the HBV model

Matteo Giuliani, Jon Herman

November 9, 2017

Contents

1 The HBV rainfall-runoff model

The Hydrologiska Byråns Vattenbalansavdelning (HBV) model (Bergström, 1976; Bergström et al., 1995) is a lumped rainfall-runoff model developed for operational flood forecasting in Sweden. The model is composed by a cascade of four storage units, as shown in Figure 1. It requires two inputs, namely precipitation P and temperature T. The output of the model is the discharge Q_{sim}. Note that the model requires the computation of actual evapotranspiration, which is obtained from potential evapotranspiration data, estimated through the Hamon method (see Appendix A).
The HBV model relies on the five state variables reported in Table 1 and the 12 parameters reported in Table 2 (the order of them is the one used in the associated code). The ranges suggested were found in previous studies on 12 watersheds in United States (see Herman et al. (2013)). The current version of the model can be used in two ways, namely *calibration* and *simulation*, as described in the next sections.

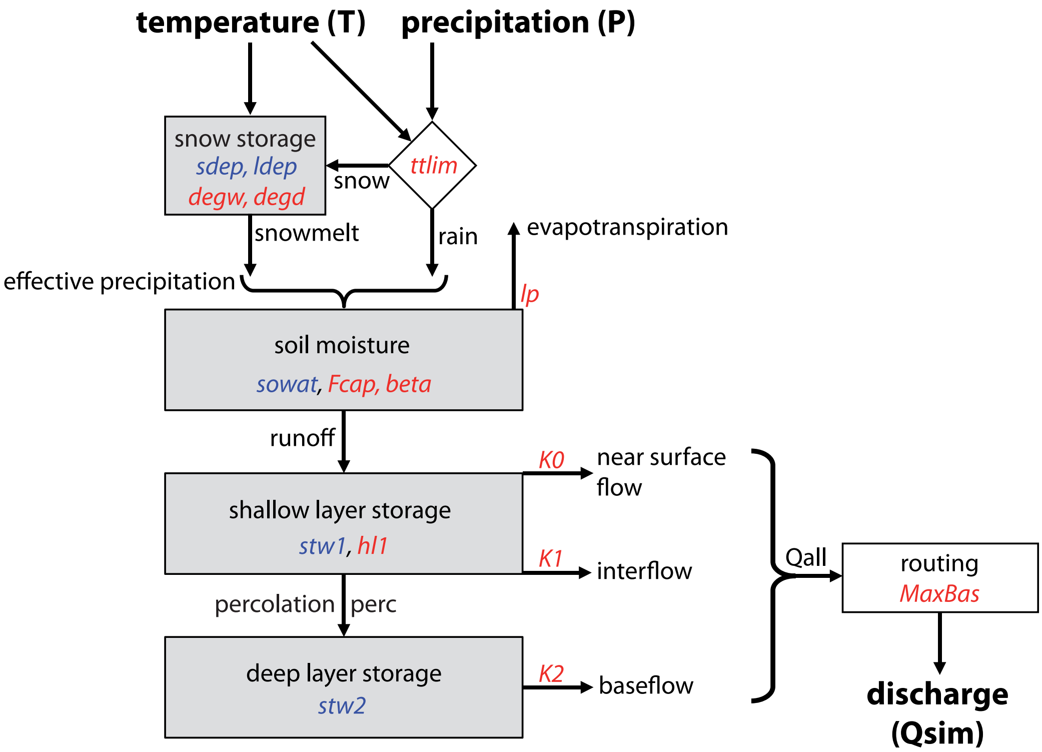

Figure 1: HBV model schematics: storage elements are gray shaded, model states and model parameters are shown in blue and red, respectively.

Table 1: HBV states.

State	Description
$sowat$	soil water storage
$sdep$	snow store
$ldep$	depth of liquid in the snow store
$stw1$	soil storage - shallow layer
$stw2$	soil storage - deep layer

Table 2: HBV parameters.

Parameter	Units	Description	Range
K_2	[day]	withdrawal rate from deep layer	(10; 20,000)
K_1	[day]	withdrawal rate for shallow layer overflow	(1; 100)
K_0	[day]	withdrawal rate from shallow layer (interflow)	(0.5; 20)
$MaxBas$	[hour]	length of hydrograph routing transformation	(24; 120)
$degd$	[mm/(day °C)]	degree day factor (snowmelt rate)	(0; 20)
$degw$	[°C]	base temperature above which melt occurs	(-3; 3)
$ttlim$	[°C]	temperature threshold below which freezing occurs	(-3; 3)
$perc$	[mm/day]	percolation rate into deep layer	(0; 100)
β	[-]	distribution of soil stores	(0; 7)
lp	[-]	limiting soil moisture at which PET takes place	(0.3; 1)
F_{cap}	[mm]	maximum soil moisture storage	(10; 2,000)
$hl1$	[mm]	maximum shallow layer storage	(0; 100)

1.1 Model calibration

The calibration on the HBV model aims to estimate the best values for the 12 parameters (see Table 2), given a time series of observations for both model input (i.e., temperature and precipitation) and model output (i.e., discharge). Given the complexity and non-linearity of the modeled processes, the HBV calibration can be a difficult operation. To guarantee a good calibration, the model can be connected to the MOEA Framework (www.moeaframework.org), a free and open source Java library for developing and experimenting with state-of-the-art multi-objective evolutionary

algorithms (MOEAs). The *CalHBV.java* file provides the link between the model to the Java library. This allows the user to run the model calibration with the algorithms available in the MOEA Framework, such as NSGAII, ε-MOEA, or GDE3.
In its current form, the calibration routine will output (or optimize) the relative variability (*alpha*), absolute value of the relative bias (*beta*) and the correlation (r) between the simulated and observed flows over the simulated time period. These objectives represent three components of the Nash Sutcliffe Efficiency (see Gupta et al., 2009). However, the output can easily be modified to include any combination of states/fluxes or error metrics from any time during the simulation.
To run the calibration, besides the compilation of the HBV model, the user has first to specify the data file in the *CalHBV.java* and then compile the java class:

$javac\ {-classpath}\ MOEAFramework - 1.16 - Executable.jar : .\ CalHBV.java$

Finally, run the generated java program:

$java\ {-classpath}\ MOEAFramework - 1.16 - Executable.jar : .\ CalHBV$

At the end of the HBV calibration, the best parameterization is printed in the terminal along with the corresponding RMSE.

1.2 Model simulation

The simulation of the HBV model aims to estimate the discharges, given a time series of model input (i.e., temperature and precipitation) observation and the values of the model parameters (e.g., stored in the file *hbv_param.txt*). To run the simulation of the HBV, the user does not need the Java files. Once the model is compiled, the user runs a simulation as follows:

$./SimHBV\ input\_file\ output\_file < hbv\_param.txt$

where $input\_file$ is the file containing the observations of temperature and precipitation (see Section 1.3) and $output\_file$ is the name of the file where the simulated flows will be saved.

1.3 Input data

The input data should be format shown in the *example_data* directory, shown in Figure 2.
The file header contains information about the watershed (i.e., name, area, and geographic coordinates used for the estimation of the evapotranspiration) along with the number of time-steps and a flag which identifies the type of temperature data, namely daily average (i.e., 1) or daily minimum and maximum (i.e., 2). The data are organized by columns as follows:

1. year
2. month
3. day
4. mean areal precipitation (mm)
5. daily streamflow discharge (mm)
6. daily average air temperature (Celsius)
7. daily maximum air temperature (Celsius)
8. daily minimum air temperature (Celsius)

Actually, the model requires only daily average temperature. If these data are available, the file will contain only 6 columns. Otherwise, it has seven columns and the average is computed within the model. Examples of both types of input file are contained in the *example_data* directory. When the model is used for simulation purposes (with no calibration), the observations of the discharges are not needed. All the data in the corresponding column (i.e., the fifth one) can be set equal to zero.

2 The code implementation

The core of the HBV model implementation is the *calc_HBV()* function, which performs the simulation of the main rainfall-runoff processes with a daily time step. This function first initializes the HBV states and fluxes to zero. It then assigns the model parameters, which are taken as input (i.e., organized in an array of doubles). Finally, it sequentially runs the five functions described in the next subsections. The structure of the *calc_HBV()* function is shown in the Pseudocode 1.

2.1 The snow function

The aim of this function is to estimate the effective precipitation for the current day t. Depending on the average temperature of the current day $T_{avg}(t)$, the corresponding precipitation can be snow (if $T_{avg}(t) < ttlim$) or rain (otherwise).
Moreover, if the temperature is greater than the parameter $degw$, the snow is melting. The snow melt is defined as

$$smelt(t) = (T_{avg}(t) - degw(t)) \times degd \qquad (1)$$

```
# Gage information

<WATERSHED_NAME>     GUADALUPE RIVER NR SPRING BRANCH, TX
<GAGE_LATITUDE>      29.860600      # Gage latitude in decimal degrees
<GAGE_LONGITUDE>    -98.382800      # Gage longitude in decimal degrees
<DRAINAGE_AREA>     1315.000000     # Drainage basin area in square miles
<TIME_STEPS>         20454          # Number of time steps of data
<TEMP_DATA>              1          # Type of temperature data (1=daily average, 2=min and max)

# Data column format

<DATA_FORMAT_START>
1  year
2  month
3  day
4  mean areal precipitation (mm)
5  daily streamflow discharge (mm)
6  daily average air temperature (Celsius)
7  daily maximum air temperature (Celsius)
8  daily minimum air temperature (Celsius)
<DATA_FORMAT_END>

# Data

<DATA_START>
   1.9480000e+03   1.0000000e+00   1.0000000e+00   0.0000000e+00   1.7300000e+00   6.3500000e+00
   1.9480000e+03   1.0000000e+00   2.0000000e+00   0.0000000e+00   1.7310000e+00   5.2888500e+00
   1.9480000e+03   1.0000000e+00   3.0000000e+00   0.0000000e+00   1.7300000e+00   1.0300000e+01
   1.9480000e+03   1.0000000e+00   4.0000000e+00   0.0000000e+00   1.7290000e+00   1.0613900e+01
   1.9480000e+03   1.0000000e+00   5.0000000e+00   0.0000000e+00   1.7290000e+00   1.0538900e+01
   1.9480000e+03   1.0000000e+00   6.0000000e+00   0.0000000e+00   1.7290000e+00   1.0716700e+01
```

Figure 2: Example of a file containing the input data formatted for the HBV model.

where $degd(t)$ is the degree day factor or snowmelt rate. The melted snow eventually contributes to the effective precipitation and is removed to the snow storage $sdep(t)$.

2.2 The soil function

The aim of this function is to estimate the soil storage of the shallow layer for the current day t. If the soil moisture storage $sowat(t)$ is greater than the maximum capacity F_{cap}, the runoff Q_{runoff} is equal to the effective precipitation plus the excess of soil moisture storage with respect to F_{cap}. The soil moisture storage is therefore equal to its maximum capacity. Conversely, if $sowat(t) < F_{cap}$, a portion of the effective precipitation goes into the soil storage and only the remaining part contributes to Q_{runoff}.
Then, the function estimates the actual evapotranspiration $AET(t)$ and compares it with the soil moisture storage. If there is enough water in the soil, the volume $AET(t)$ evaporates. If $AET(t) > sowat(t)$, all the water evaporates and $sowat(t) = 0$. Finally, the soil storage of the shallow layer $stw1(t)$ is updated:

$$stw1(t) = stw1(t-1) + Q_{runoff} \tag{2}$$

Pseudocode 1 The *calc_HBV()* function.

Inputs: an array of 12 parameters
Initialization:
Set HBV states and fluxes to 0.
Assignment of model parameters.
Iterations:
- computation of effective precipitation (**snow** function)
- set runoff depth (**soil** function)
- computation of discharge (**discharge** function)
- routing (**routing** function)
- update to the next time step (**backflow** function)

2.3 The discharge function

The aim of this function is to compute the discharge $Q_{all}(t)$, which depends on the near surface flow Q_0, the interflow Q_1, and the baseflow Q_2 (eq. 3). These three contributions depend on the soil storages $stw1(t)$ and $stw2(t)$ in the shallow and deep layer, respectively. The flow Q_0 is observed only if $stw1(t) > hl1$. The interflow Q_1 is generated only if there will be water in the shallow layer after having removed Q_0. If a volume is still present in the shallow layer (i.e., $stw1(t) - Q_0 - Q_1 > 0$), this water percolates from the shallow to the deep layer. Finally, depending on the deep layer storage $stw2(t)$, the baseflow Q_2 is estimated.

$$Q_{all}(t) = Q_0 + Q_1 + Q_2 \tag{3}$$

2.4 The routing function

The aim of this function is to transform the discharge $Q_{all}(t)$ using the parameter $MaxBas$ for routing. This transformation is performed by a triangular weighting function, which eventually estimates the final flow $Q_{sim}(t)$.

2.5 The backflow function

This function reinitializes to zero the arrays used for the routing depending on the parameter $MaxBas$.

A The Hamon potential evapotransiration

The potential evaportanspiration is computed according to the Hamon method (Hamon, 1961) and depends on the average temperature T_{avg} and the location (i.e., latitude) of the catchment according to the formulation reported in eq. 4

$$PET = k \times dL/24 \times \frac{e_s}{T_{avg} + 273.2} \tag{4}$$

where $k = 715.5$, dL is the daytime length (which depends on the day of the year and the location), and e_s the saturation vapor pressure (which depends on the average temperature).

References

Bergström, S. (1976). “Development and application of a conceptual runoff model for scandinavian catchments.” *Report no.*, SMHI Report RHO7, Norrköping.

Bergström, S., Singh, V., et al. (1995). “The hbv model..” *Computer models of watershed hydrology.*, 443–476.

Gupta, H. V., Kling, H., Yilmaz, K. K., and Martinez, G. F. (2009). “Decomposition of the mean squared error and nse performance criteria: Implications for improving hydrological modelling.” *Journal of Hydrology*, 377(1), 80–91.

Hamon, W. (1961). “Estimating potential evapotranspiration.” *Transactions of the American Society of Civil Engineers*, 128(1), 324–337.

Herman, J., Reed, P., and Wagener, T. (2013). “Time-varying sensitivity analysis clarifies the effects of watershed model formulation on model behavior.” *Water Resources Research*, 49(3), 1400–1414.