GHENT UNIVERSITY

FACULTY OF BIOSCIENCE ENGINEERING

Exercises Hydrological Modelling

Academic year: 2025 - 2026

Olivier Bonte Hans Lievens

2026-07-02

Table of contents

1 Introduction

This book contains exercises for the course Hydrological Modelling at Ghent University. The focus of this set of exercises will be on rainfall-runoff modelling with a conceptual hydrological model.

The objective of the practical exercises of this course is to illustrate some of the most important steps necessary to perform a proper modelling of a medium-scale catchment in Flanders. Only some specific steps of the process will be illustrated here across 5 exercises:

1. Delineation of the catchment (Chapter 4)
2. Implementation of a hydrological model (Chapter 5)
3. Calibration of the hydrological model (Chapter 6)
4. The analysis of uncertainty in the model output of the model (Chapter 7)
5. Assimilation of observations into the model (Chapter 8).

Needless to say that for each step, there are many options, and that the choice for e.g. a specific model, calibration algorithm, or evaluation criterium depends on the specific situation, the aim of the modelling exercise, and the personal taste of the modeller. For an extensive – yet non-exhaustive – overview of different techniques in hydrological modelling, we refer to the theoretical course notes and specialized literature.

The aim of the practicals is to:

1. Make you acquainted with the advantages and disadvantages of the selected algorithms,
2. Make you aware of the potential problems related to modelling rainfall-runoff relationships, and
3. Point you to and make you reflect on existing alternatives for the algorithms selected here.

2 Practical information

Before diving into the exercises, some practical information is provided on how to effectively set up your working environment.

2.1 The repository

All code used to create this assignment is available at https://github.com/h-cel/hydrological-modelling. This collection of files is called a repository. To get started, go the the URL above, click on the green `Code` button and select `Download ZIP` (see Figure 2.1). Unzip the downloaded file and place the resulting folder at a location of your choice. This folder contains all the files you need to complete the exercises.

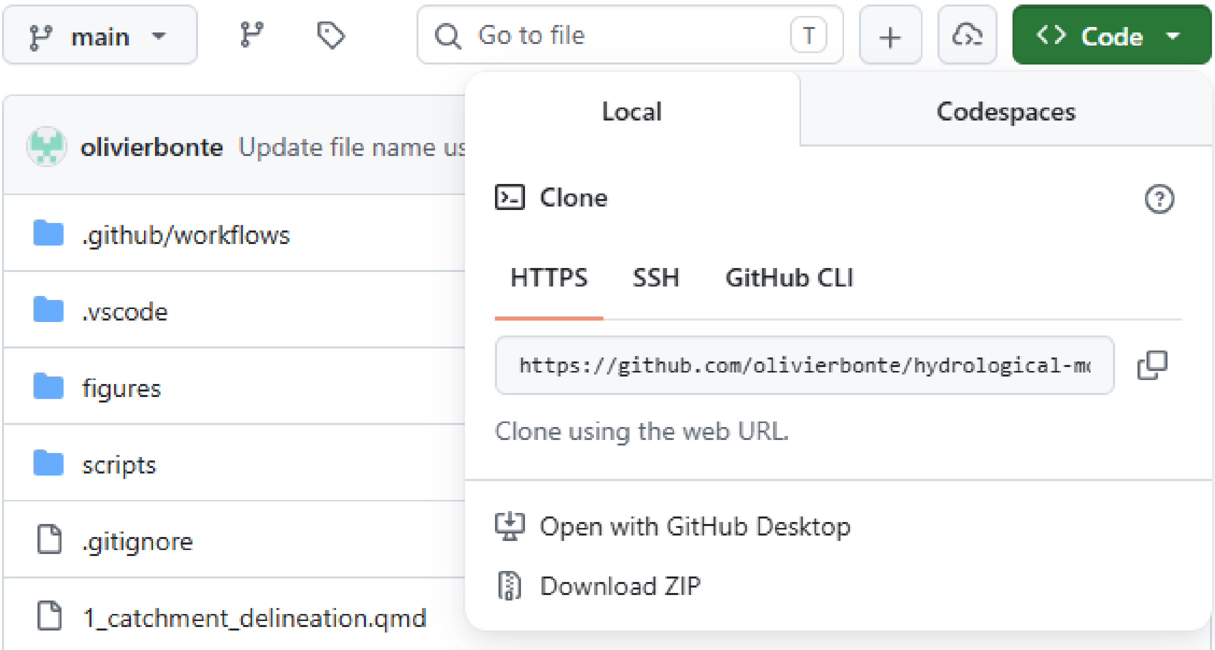

Figure 2.1: Screenshot of GitHub download button

Note that not only the code, but also the processed data is now downloaded. You can find this data in `data/processed`. More info on the data sources and processing is given in Chapter 3.

This assignment is provided to you in two formats, up to you which one you prefer to work with:

1. A website at https://h-cel.github.io/hydrological-modelling-public/
2. A PDF document, which you can download from https://h-cel.github.io/hydrological-modelling-public/Exercises-Hydrological-Modelling.pdf

2.2 Working with Quarto and Python

A general introduction to Quarto and Python is given in… LINK TO DEPARTMENT WIDE INTRO HERE LATER.

The idea of this exercise set is that you will use Quarto to combine your code and your scientific report in a single document. The goal is to write a report following the principles of reproducible research[1], where others have access to all the data and code needed to obtain the same results as the one you present.

In these practicals, you will therefore **not** write a report from scratch (in Microsoft Word, Google Docs…), but instead you'll extend the provided `.qmd` files for each exercise. The files of interested are numbered from `1_...qmd` to `6_...qmd`. Here you can add Python code[2] for your figures and write your discussion in Markdown.

> **Tip**
>
> Don't perform al your computations inside of the `.qmd` files! Instead, make a regular `.py` file inside of the `scripts` folder for each exercise. From these `.py` files, you can save the results (e.g. metrics, output data) in (for example) a `data/solutions` folder. From this folder, you can then import the necessary results in the `.qmd` files to write your report and make your figures.

2.3 Evaluation

Hand in your report of this assignment via Ufora by ADD DATE LATER. Please hand in:

- A PDF version of your report, which is rendered via Quarto.
- The full set of files (i.e. repository) including your solution as a `.zip` file.

Your report should be written **scientifically** and should contain at least the following:

- A summary of the results with reference to relevant figures and metrics, and a proper discussion. Note that all tasks are related and that the results should be discussed in reference to one another in a scientific manner. The emphasis of the report should be on this part. This part should be used to draw the attention on shortcomings of the different algorithms and make a link to potential alternatives. Use the sections `Results and discussion` in `1...qmd` to `5...qmd` for writing this part down.
- A conclusion about the modelling exercise as a whole, which is written in the `6_conclusion.qmd` file.
- To support your discussion, it is valuable to include references to relevant scientific papers. To add a reference, please include the relevant source in its bibxtex format to `references.bib`. More information on how to handle citations in Quarto can be found here. The bibtex format is explained in this guide. By means of example, one scientific article, Moore (2007), is already included.

[1]This concept is introduced in your course on Data Science in the 2nd Bachelor year of Bioscience Engineering, where you made a Quarto/Rmarkdown presentation/report for a data analysis

[2]As an example of how this is used in practice, see e.g. the source file `0_study_area_and_data.qmd` for Chapter 3

3 Study area and data

For these exercises, a hydrological dataset of the Zwalm catchment will be used. The Zwalm catchment is located in the center of Belgium and can be considered a medium-scale catchment. Its main tributary, the Zwalmbeek, is visualised in Figure 3.1. During the period 2009-2025, a dataset of daily precipitation, potential evapotranspiration and discharge is at your disposal (see Section 3.3 for more information). The average annual rainfall and potential evapotranspiration of the catchment are 892 mm and 561 mm respectively, typical values for Belgium.

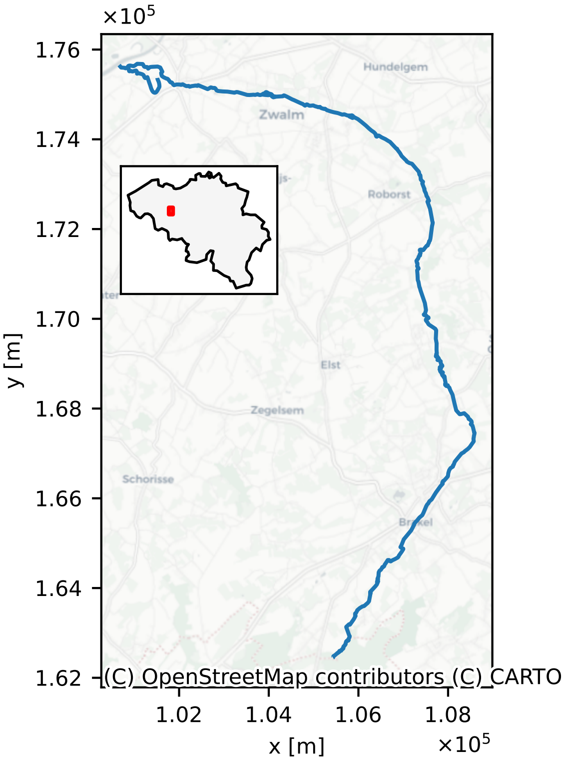

Figure 3.1: The Zwalmbeek and its location within Belgium, as denoted by the red box (CRS: EPSG:31370).

As mentioned in Chapter 2, all the processed data is available at `data/processed`. If interested, you can inspect (and execute) all downloading and processing steps yourself by following the instructions in `scripts/data_download/README.md` and `scripts/data_process/README.md`.

In the next section, more information is given on the data sources and processing steps for the different types of provided data. Note that all geographic data is provided in the Belgian Lambert 72 (BD72, EPSG:31370) coordinate reference system (CRS).

3.1 Catchment data

The data in the data/processed/catchment_info folder contains information on the river network and the catchment boundaries in the broad vicinity of the Zwalm catchment. The source of the river network is the *Vlaamse Hydrografische Atlas* (Vlaamse Milieumaatschappij 2026). The catchment boundaries come from the *Oppervlaktewaterlichamen en hun afstroomgebieden, 2022-2027* dataset (Vlaamse Milieumaatschappij 2023). In Chapter 4, you will be asked to delineate the catchment boundary of the Zwalmbeek yourself. The provided catchment boundaries can be used for comparison. Both files are provided as ESRI Shapefiles, called afstroomgebied.shp and vlaamse_hydrografische_atlas.shp respectively.

3.2 Digital terrain model

For the catchment delineation of Chapter 4, a digital terrain model (DTM) is provided in data/processed/digital_terrain_model. This DTM is derived from the *Digitaal Hoogtemodel Vlaanderen II* DTM at 1m spatial resolution (Agentschap Digitaal Vlaanderen 2014). To reduce the size of the dataset and speed up the computation, the DTM has been resampled to a spatial resolution of 10.0 m by averaging. The DTM is provided as a GeoTIFF file, called DHMVII_DTM_1m_upscaled_10m.tif.

3.3 Meteorological and discharge data

Both meteorological forcings and observed discharge are provided in the folder data/processed/forcings_discharge. All this data is retrieved from Waterinfo, a website which combines hydro(meteorological) data from both the Flanders Environment Agency and Flanders Hydraulics Research. Programmatic access to the data is possible via the Python package pywaterinfo, as demonstrated (for your information only) in scripts/data_download/waterinfo_download.py.

The combined forcings and discharge dataset is provided in the CSV file forcings_discharge.csv. An overview of the the original data sources is given in Table 3.1. Note that mm is equivalent to mm/d because of the daily time resolution. The locations of the different measurement stations are additionally shown in Figure 3.2. Some processing details:

- The variable Precipitation (P) of catchment is a combination of rainfall gauge and radar data, representative for the Zwalm catchment. The gaps in this dataset are filled with precipitation data from the Pluviometer in Maarke-Kerkem.
- Potential evapotranspiration (E_{p}) is estimated using the Penman equation at a meteorological station. Missing values are gapfilled with the daily, smoothed climatology of E_{p}. Smoothing occurs by applying a moving average with a centred window of 11 days.
- River discharge (Q) is measured near the mouth of the Zwalmbeek in the village of Nederzwalm. No gapfilling is applied to this variable, as it is only used for model evaluation, not as a model input.

Table 3.1: Metadata of the provided daily forcings and discharge dataset, retrieved from Waterinfo. Each variable has a corresponding measurement station, denoted by a unique station ID.

Variable	Units	Station	Station ID
Precipitation	mm	Maarke-Kerkem_P	P06_014
Precipitation of catchment	mm	Nederzwalm/Zwalmbeek	L06_342
Potential Evapotranspiration	mm	Waregem_ME	ME05_019
River Discharge	m^3/s	Nederzwalm/Zwalmbeek	L06_342

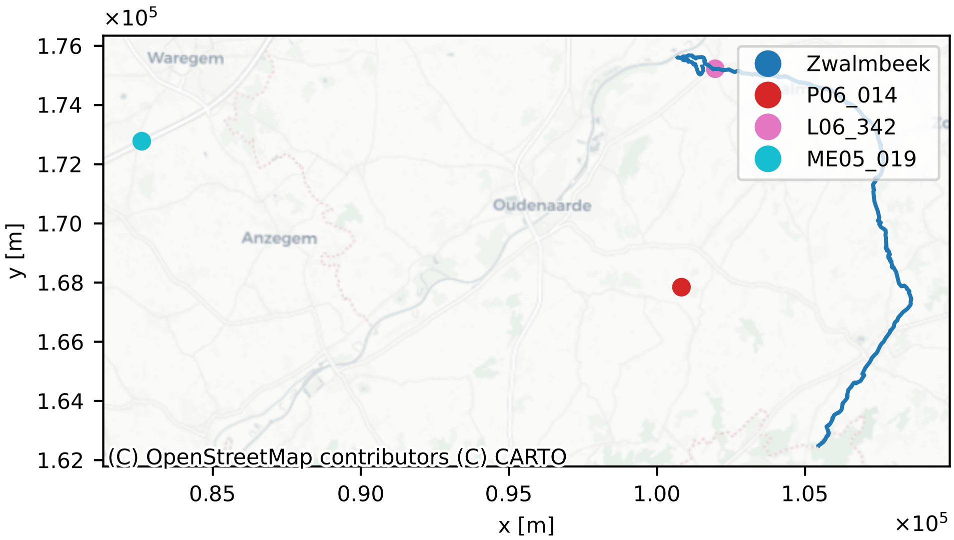

Figure 3.2: The Zwalmbeek and the measurement locations as denoted by their station ID

Consequently, following columns are present in `forcings_discharge.csv`:

```
['', 'precipitation', 'potential_evapotranspiration', 'river_discharge']
```

Note that the first column without a name is the index column, which contains the date. The corret interpretation of the date is that the data value corresponds to the daily sum (P and E_p) or daily mean (Q) of the day indicated by the date. For example, the value of P on 2020-01-01 corresponds to the total precipitation that fell during that day.

> **Tip**
>
> To read in `forcings_discharge.csv`, checkout the documentation for `pandas.read_csv()`, specifically of the `index_col` and `parse_dates` parameters, to correctly read in the dataset.

3.4 Satellite soil moisture

In Chapter 8, the goal is to improve the hydrological model predictions by assimilating observations of soil moisture. Here, we'll use satellite-derived surface soil moisture (SSM), which is provided in the folder `data/processed/satellite_soil_moisture`.

The original data source is the SSM product at 1 km spatial resolution over Europe form the Copernicus Land Monitoring Service (European Union's Copernicus Land Monitoring Service 2026). An example of the original data for one example date is shown in Figure 3.3. The SSM values are expressed in percent saturation and are representative of the top few centimeters (~5cm) of the soil.

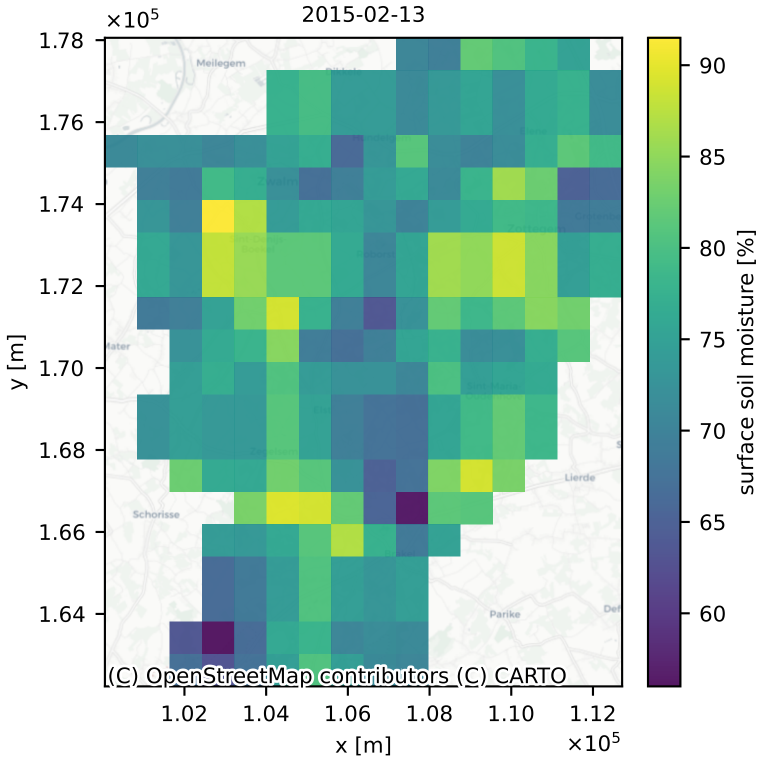

Figure 3.3: Satellite-derived surface soil moisture data at 1 km spatial resolution over the Zwalm catchment.

For ease of use, the dataset has been processed to a timeseries by taking the average SSM value over the Zwalm catchment and subsequently removing outliers. The resulting final timeseries is found is called `sattelite_soil_moisture.cv`, which has following columns:

```
['time', 'ssm', 'noise']
```

with `time` the date of acquistion, `ssm` the spatially averaged SSM and `noise` is the spatially averaged standard deviation of the SSM following the rules of propagation of uncertainty[1].

[1]For any function f, the standard deviation is given by $\sigma_f = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 \sigma_x^2 + \left(\frac{\partial f}{\partial y}\right)^2 \sigma_y^2 + \left(\frac{\partial f}{\partial z}\right)^2 \sigma_z^2 + \cdots}$ assuming no covariance between $x, y, z \cdots$ and linearly approximating f. Note that an average is a linear function, so here this is not an approximation. With the average $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ as f, $\frac{\partial f}{\partial x_i} = \frac{1}{N}$ and so $\sigma_{\bar{x}}^2 = \sum_{i=1}^{N}\left(\frac{\partial \bar{x}}{\partial x_i}\right)^2 \sigma_i^2 = \frac{1}{N^2}\sum_{i=1}^{N}\sigma_i^2$

4 Exercise 1: Delineation of the Zwalm catchment

4.1 Assignment

The delineation of a catchment based on a digital elevation model is often one of the first steps in a hydrological modelling study. The procedure is relatively straightforward, but the digital elevation model needs to be properly processed prior to the analysis. The pre-processing of the digital elevation model at least includes the removal of data gaps (i.e. spatial interpolation) and the removal of sinks (or depressions). Based on the processed DTM, the catchment can then be easily delineated using the simple principle that water always flows in the direction of the largest gradient. Different algorithms exist to do so (see course notes); here, only the gradient of every cell to its eight direct neighbours, the deterministic 8 (D8) algorithm, will be used.

> Tip 1
>
> The outlet of the Zwalm catchment is located in the vicinity of x=101673 m and y=17537 m

For this geospatial analysis, GRASS will be used, a powerful computational engine for raster, vector, and geospatial processing. It supports terrain and ecosystem modelling, hydrology, data management, and imagery processing. There are multiple ways of using GRASS. In this exercise, you can choose between 2 ways of interacting with GRASS:

- With QGIS as Graphical User Inteface (GUI), all GRASS functionality can be accessed via the Procesing Toolbox
- The Python GRASS interface allows programmatic access to GRASS functionality. For an in depth introduction, we refer to the documentation. In this exercise, we will use the `grass.tools` interface. The initial setup of the GRASS environment is provided in `scripts/1_catchment_delineation.py`, to which you can add your own code. All tools are accessible via the `tools` variable defined there. For example if you want ot use the `r.import` function, you would call `tools.r_import(input="PATH_TO_FILE",output="OUTPUT_NAME", overwrite=True)`. Note that:
  - It is recommended to set the `overwrite=True` argument for all functions.
  - It is a general convention that the `.` defined in the GRASS function name (as found in the tool manual)is replaced by `_` in the Python function name.
  - The output of a previous function (e.g. `OUTPUT_NAME` in the example above) can be used as input for a subsequent function. For example, if you want to use the output of `r.import` as input (called `elevation`) for a slope calculation with `r.slope.aspect`, you would call `tools.r_slope_aspect(elevation="OUTPUT_NAME", slope="SLOPE", overwrite=True)`.

- In scripts/helper_functions.py, two helper functions are defined: get_data_array_from_raster gives you an xarray.DataArray given the name (e.g. "OUTPUT_NAME") of a GRASS raster layer and plot_grass_raster makes a plot of a GRASS raster layer (given its name as again). Use these functions to check intermediate results.

Irrespective of the choice of interface, an overview of all GRASS functions can be found in tool manual. Specific tips for each interface will be denoted by their logo in the steps below.

1. Start by loading the DTM (see Section 3.2). Check if it is clipped to a reasonable region of interest around the Zwalm river (bounding box of approximately $x_{\text{min}} = 98000$ m, $x_{\text{max}} = 116000$ m, $y_{\text{min}} = 160000$ m, $y_{\text{max}} = 180000$ m).
   - Use the r.import function to load in raster data.
2. Spatially gap-fill the DTM (i.e. make sure there are no missing values).
   - Use the r.fillnulls function to fill in missing values.
   - Use the Fill nodata tool (from GDAL).
3. Filter the digital elevation model for sinks (or depressions) and fill them. Use the r.fill.dir function.
4. Calculate the flow direction and accumulation into each grid cell of the domain. Use the r.watershed function. Make sure that D8 is used as flow direction algorithm. Further more, ensure that the minimum size of an exterior watershed basins is set to 500. Keep both the accumulation (i.e. how many cells drain into each cell) and the flow direction map as outputs.
5. Check if the accumulation map aligns with the actual river network of the Zwalm catchment from the *Vlaamse Hydrografische Atlas* (see Section 3.1). In the attributes of the shapefile, you can filter in "NAMEN" for "Zwalmbeek - Dorenbosbeek" to only withhold the main Zwalm river.
   - Use get_data_array_from_raster to get the accumulation map as an xarray.DataArray. Use geopandas.read_file to read in the river network shapefile.
6. Find the point with the largest accumulation that is located within the Zwalm river network. Rasterise the river network to the same grid as the accumulation map to do so. This point is the outlet of the catchment.
   - Use the make_geocube function from the geocube package. Use the like argument.
   - Use the Rasterize (vector to raster) tool (from GDAL).
7. Given the outlet determined in the previous step, delineate the catchment using r.water.outlet. Vectorize the resulting raster map of the catchment.
   - Convert the output raster to an xarray.DataArray using a helper function. Subsequently, the vectorize function from the geocube package can be applied.
   - Use the Polygonize (raster to vector) tool (from GDAL).
8. Repeat step 7 for the coordinates provided in Tip 1. Compare the resulting catchment boundary with the one obtained in step 7 and with the provided catchment boundary from the *Oppervlaktewaterlichamen en hun afstroomgebieden, 2022-2027* dataset (see Section 3.1). What are the differences between these catchment boundaries? What could be the reasons for these differences?
9. Calculate the catchment area.

In your results and discussion below, make sure to include an overview figure of the Zwalm catchment including the DTM, river network and catchment boundary.

4.2 Results and discussion

5 Exercise 2: Implementation of the model

5.1 Assignment

The lumped, conceptual rainfall-runoff model implemented here is a simplified and modified version of the FLEX model as described in Fenicia et al. (2006) and Fenicia et al. (2008). The simplifications include not considering an interception reservoir and omitting lag functions for the routing. An overview of the model structure is given in Figure 5.1. The model considers three state variables, corresponding to the volume of water in three reservoirs:

1. The unsaturated soil reservoir $S(t)$ [mm]
2. The slow reacting reservoir $S_1(t)$ [mm], conceptually representing the groundwater system
3. The fast reacting reservoir $S_2(t)$ [mm]

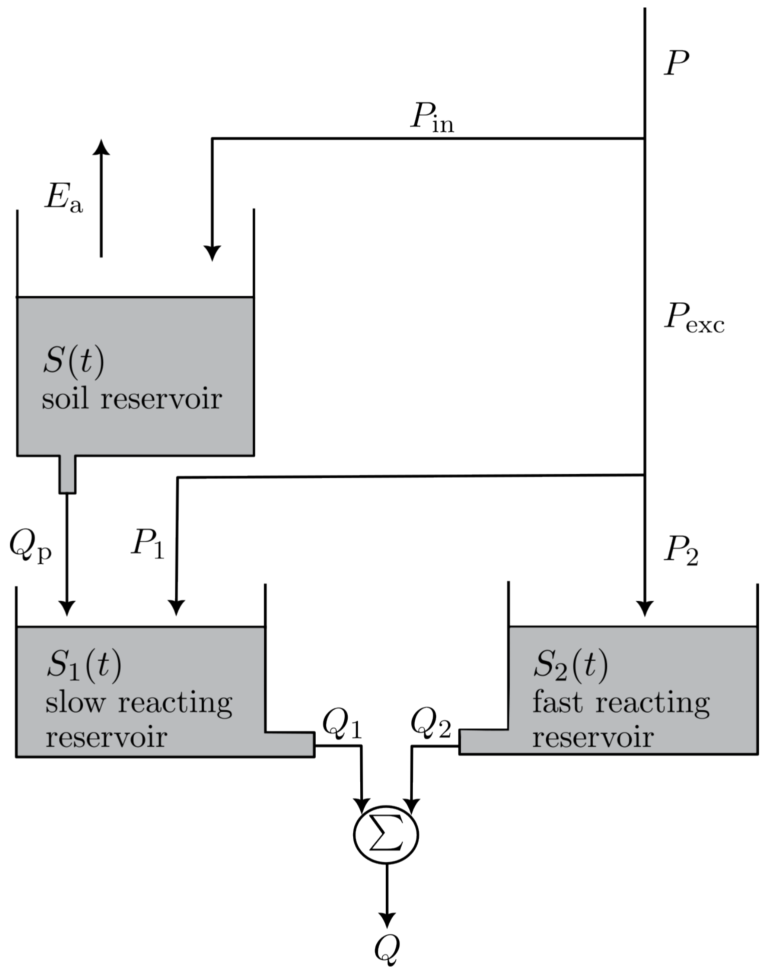

Figure 5.1: Conceptual representation of the hydrological model

The model is forced with precipitation $P(t)$ and potential evaporation $E_p(t)$ (see Section 3.3 for details on the forcing data). Note that mm is used as a unit here, as it allows to implement the model independently of the catchment area[1]

Table 5.1: Model parameters together with their units, minimum and maximum value and a set of uncalibrated values

Parameter	Units	Minimum	Maximum	Value
λ	-	0.5	1.5	0.65
S_{max}	mm	87	430	350
b	-	0.5	1	0.86
α	-	0.5	1	0.8
$Q_{p,\mathrm{max}}$	mm/d	0.75	75	34
β	-	0.05	0.1	0.06
γ	-	5	10	9.8
$S_{2,\mathrm{max}}$	mm	13	26	14
κ_2	mm/d	75	190	110
κ_1	1/d	0.26	0.52	0.32

By applying the conservation of mass to each reservoir, three ordinary differential equations (ODEs) can be derived to describe the change in storage in each reservoir over time.

$$\frac{dS(t)}{dt} = P_{\mathrm{in}}(t) - E_{\mathrm{a}}(t) - Q_{\mathrm{p}}(t) \tag{5.1}$$

$$\frac{dS_1(t)}{dt} = P_1(t) - Q_1(t) + Q_{\mathrm{p}}(t) \tag{5.2}$$

$$\frac{dS_2(t)}{dt} = P_2(t) - Q_2(t) \tag{5.3}$$

In the equations above, a number of fluxes are calculated which depend on the state variables themselves. A first flux is the actual evaporation (E_{a} [mm/d]), which can be calculated from the potential evaporation E_{p} as follows:

$$E_{\mathrm{a}}(t) = \frac{1}{\lambda}\frac{S(t)}{S_{\mathrm{max}}}E_{\mathrm{p}}(t) \tag{5.4}$$

where λ [-] is a dimensionless model parameter and S_{max} [mm] is the storage capacity of the unsaturated soil reservoir $S(t)$. The infiltration in the soil reservoir P_{in} [mm/d] is calculated using:

$$P_{\mathrm{in}}(t) = \left(1 - \frac{S(t)}{S_{\mathrm{max}}}\right)^b P(t) \tag{5.5}$$

where b is again a dimensionless model parameter. Subsequently, the excess precipitation P_{exc} [mm/d] can be estimated:

[1] $1\mathrm{mm} = 1 \cdot 10^{-3}\mathrm{m}^3/\mathrm{m}^2$

$$P_{\text{exc}}(t) = P(t) - P_{\text{in}}(t) \tag{5.6}$$

The amount of water percolating to the groundwater storage Q_{p} [mm/d] is:

$$Q_{\text{p}}(t) = Q_{\text{p,max}} \left(1 - e^{-\beta \frac{S(t)}{S_{\text{max}}}}\right) \tag{5.7}$$

with $Q_{\text{p,max}}$ [mm/d] the maximum percolation volume per unit of time and β [-] a dimensionless parameter.

P_{exc} is again partitioned between S_2 and S_1:

$$P_{\text{exc}}(t) = P_1(t) + P_2(t) \tag{5.8}$$

The part of the effective precipitation reaching S_2 is:

$$P_2(t) = \alpha \frac{S(t)}{S_{\text{max}}} P_{\text{eff}}(t)$$

with α [-] a dimensionless model parameter. P_1 is then calculated as the remaining part of the effective precipitation using Equation 5.8.

S_2 is considered as a non-linear reservoir, so that its outflow Q_2 [mm/d] can be calculated using:

$$Q_2(t) = \kappa_2 \left(\frac{S_2(t)}{S_{2,\text{max}}}\right)^{\gamma} \tag{5.9}$$

with $S_{2,\text{max}}$ [mm] the storage capacity of the fast reacting reservoir, κ_2 [mm/d] the maximum outflow rate, and γ [-] a dimensionless model parameter.

S_1 on the other hand is considered as a linear reservoir, so the outflow Q_1 [mm/d] is given by:

$$Q_1(t) = \kappa_1 S_1(t) \tag{5.10}$$

where κ_1 [1/d] the reciprocal of the residence time. The total discharge Q [mm/d] is then calculated as the sum of the outflow from the fast and slow reacting reservoir:

$$Q(t) = Q_1(t) + Q_2(t) \tag{5.11}$$

Equation 5.1, Equation 5.2, and Equation 5.3 can be denoted as a general non-linear, continuous-time state-space model:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}, \mathbf{f}, \mathbf{p})$$

with f a non-linear function, $\mathbf{x} = [S, S_1, S_2]^T$ the state vector, $\mathbf{f} = [P, E_p]^T$ the forcing vector, and $\mathbf{p} = [\lambda, S_{\text{max}}, b, \alpha, Q_{\text{p,max}}, \beta, \gamma, S_{2,\text{max}}, \kappa_2, \kappa_1]^T$ the parameter vector. For this exercise, the model will be implemented in discrete time using a simple forward Euler scheme[2]:

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + f(\mathbf{x}(t), \mathbf{f}(t), \mathbf{p})\Delta t$$

[2] For more background info on numerical methods for ODEs, see your bachelor course on differential equations

The time step Δt is set to 1 day, which is the same as the time step of the forcing data.

With all of the information above, implement the model based on Equation 5.1 to Equation 5.11. As a start, run it with the uncalibrated parameter values given in Table 5.1. Compare the simulated discharge with the observed discharge (see Section 3.3). Discuss the performance using relevant figures and metrics.

> **Tip**
>
> Some general tips and recommendations:
>
> - Implement the model in a separate file `scripts/model.py` as a function. You need to be able to reuse the model in the following exercises.
> - Make sure you can run the model one time step at a time. A for-loop can then be used to run the model for the entire time period.
> - Take into account the physical limitations of the state variables.
> - Take into account the impact of initialisation of the state variables
> - Not a single figure or metric can summarize the model performance. Nevertheless, some metrics and/or figures are (partially) linked to each other.

> **Challenge**
>
> The forward Euler scheme presented above is the simplest available numerical method to solve the ODEs. `SciPy` offers more advanced numerical time integration methods. Investigate how the choice of time integration method affects the model outcome.
> Important: this challenge is optional and only serves as an extra challenge for those who are interested.

5.2 Results and discussion

6 Exercise 3: Calibration of the model

6.1 Assignment

6.2 Results and discussion

7 Exercise 4:Uncertainty analysis of the model output

7.1 Assignment

7.2 Results and discussion

8 Exercise 5: Data assimilation

8.1 Assignment

8.2 Results and discussion

9 Conclusion

References

Agentschap Digitaal Vlaanderen. 2014. "Digitaal Hoogtemodel Vlaanderen II, DTM, Raster, 1 m." Version 2014.01. Agentschap Digitaal Vlaanderen. https://metadata.vlaanderen.be/srv/api/records/f52b1a13-86bc-4b64-8256-88cc0d1a8735.

European Union's Copernicus Land Monitoring Service. 2026. "Surface Soil Moisture 2014-Present (Raster 1 Km), Europe, Daily – Version 1." https://doi.org/10.2909/e934b15f-7d48-4c6d-a9c6-6484488aa58f.

Fenicia, Fabrizio, Hubert H. G. Savenije, Patrick Matgen, and Laurent Pfister. 2008. "Understanding Catchment Behavior Through Stepwise Model Concept Improvement." *Water Resources Research* 44 (1). https://doi.org/10.1029/2006WR005563.

Fenicia, F., H. H. G. Savenije, P. Matgen, and L. Pfister. 2006. "Is the Groundwater Reservoir Linear? Learning from Data in Hydrological Modelling." *Hydrology and Earth System Sciences* 10 (1): 139–50. https://doi.org/10.5194/hess-10-139-2006.

Moore, R. J. 2007. "The PDM Rainfall-Runoff Model." *Hydrology and Earth System Sciences* 11 (1): 483–99. https://doi.org/10.5194/hess-11-483-2007.

Vlaamse Milieumaatschappij. 2023. "Oppervlaktewaterlichamen En Hun Afstroomgebieden, 2022–2027." Versions 2022-2027. Agentschap Digitaal Vlaanderen. https://metadata.vlaanderen.be/srv/api/records/e8f55350-5c8d-4b15-ac09-1014469c3f04.

Vlaamse Milieumaatschappij. 2026. "Vlaamse Hydrografische Atlas – Waterlopen, Toestand 03/02/2026." Versions toestand 03/02/2026. Agentschap Digitaal Vlaanderen. https://metadata.vlaanderen.be/srv/api/records/d5a7a3df-a8a5-4516-92c4-d0128eea7fd7.