CrossWater model framework to calculate micropollutant loads and concentrations from diffuse sources

Andreas Moser, Eawag, 2017

# Model approach

The CrossWater model framework allows the modelling of micropollutants in a large river basin such as the Rhine. The framework calculates loads and concentrations at any selectable points within the basin. For this purpose the river basin must be subdivided into small scale subcatchments for which the input data must be supplied. At the scale of the subcatchments, the substance transfer module (Honti et al. 2016) is applied. Two modelling options are available, the first is a straightforward approach of load aggregation where the output of the substance transfer model are aggregated for the upstream areas of the selected outlets. With the second option the model AQUASIM (Reichert 1994) is set up with input from the substance transfer model and run.



# Input data

The parameters and the paths to the input data such as the other parameters must be included in the config.ini file.

## Preprocessing

Input data consists of three text files with time series of discharge, precipitation and temperature for every subcatchment. Three dBase files are required with the physical properties of the subcatchment such as application area and rate of the substances. They can be directly taken from the ArcGIS shapefiles (dbf files). In addition, a few parameters describe in what mode and for which substance the model is run.

### Discharge data

**discharge\_path:** The model requires hourly discharge data for every subcatchment. The data is stored as delimiter- separated values in a text file with the time steps in the rownames and the subcatchment ID’s in the columnnames. The separators are semicolons and the unit of the values is in m3/s.

**local\_discharge\_path:** The locally produced discharge is the amount of water with origin in the corresponding catchment. For headwaters this is equal to the discharge at the outlet and for other catchments this needs to be estimated or modelled.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | “ID1” | “ID2” | “ID3” | … | “IDn” |
| 2010-01-01 00:00:00 | Q [m3/s] | … |  |  |  |
| 2010-01-01 01:00:00 | … |  |  |  |  |
| 2010-01-01 02:00:00 |  |  |  |  |  |
| … |  |  |  |  |  |
| YYYY-MM-DD hh:mm:ss |  |  |  |  |  |

### Precipitation data

**precipitation\_path:** When the model is run for biocide concentrations, hourly precipitation data must be present in the a text file with the same format as the discharge data. The unit must be in mm.

### Temperature data

**temperature\_path:** Temperature data are daily values for stored as semicolon delimited text file

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | “ID1” | “ID2” | “ID3” | … | “IDn” |
| 2010-01-01 | T [°C] | … |  |  |  |
| 2010-01-02 | … |  |  |  |  |
| 2010-01-03 |  |  |  |  |  |
| … |  |  |  |  |  |
| YYYY-MM-DD |  |  |  |  |  |

### Catchments

**catchment\_path:** The total catchment area is divided in small subcatchments each having a unique identification number and being connected with a tree structure. Every catchment is connected to one catchment downstream, but may have zero up to several catchments upstream. Only the outlet of the total catchment area has no connection downstream. The input data of the catchment properties must be in the dBase format and have at least the four attributes described hereafter, but can also contain other unused attributes:

|  |  |  |
| --- | --- | --- |
| Attribute | Unit | Description |
| WSO1\_ID | [-] | Unique identification number. |
| NEXTDOWNID | [-] | Identification number of catchment downstream |
| STRAHLER | [-] | Stream order number after Strahler. |
| AREA | [m2] | Total area of catchment. |

### Application area

**landcover\_path:** The application area is the surface, where the substance is applied on. For the pesticides this is the agricultural land use, where the surface of crops is listed per subcatchment. In the case of biocides the application area is either the facades or the area of the roofs. The format of the tables must be dBase. In the following the input tables and their attributes are described.

Land use data for pesticides:

|  |  |  |
| --- | --- | --- |
| Attribute | Unit | Description |
| WSO1\_ID | [-] | Unique identification number. |
| barl, swhe, dwhe, lfall, flow, lmaiz, oats, ocro, ocer, ofar, oind, roof, oveg, pota, puls, lrape, ryem, soya, sugb, sunf, ltext, toba, toma, pari, nurs, ltwin, loliv, lfrui, gras, citr | [km2] | Total area of described crop within subcatchment. *Names of crops may also be called differently, but the unit must be in km2.* |

Building data for biocides:

|  |  |  |
| --- | --- | --- |
| Attribute | Unit | Description |
| WSO1\_ID | [-] | Unique identification number. |
| FOOTPRINT | [km2] | Total area of building footprints per subcatchment. |
| FACADE | [km2] | Total area of building facades per subcatchment. |

### Application rate

**micropollutant\_path:** The application rate of a substance is the mass used per unit area. It is derived from national sales figures or surveys on farms and therefore varies regionally. The dbf table must contain the attributes about the catchment identification number and the rate for the substance.

|  |  |  |
| --- | --- | --- |
| Attribute | Unit | Description |
| WSO1\_ID | [-] | Unique identification number. |
| APPL\_RATES | [g/m2] | Application rate of the active substance. |

### Other parameters and folder paths

**selected areas:** List of crops or biocide application area where the micropollutant is applied on. Must correspond to the names in the land use or housing dbf table (see application areas) and contain at least one entry.

**max\_ids (*optional*):** Maximum number of catchments that are processed, counting along the columns of the discharge table. If left empty all the catchments of the input data is processed.

**batch\_size (*optional*):** Parameter defining batch size of input files to be read at the time. Needed if computer memory is too low to read entire precipitation and discharge files at once.

**strahler\_limit (*optional*):** only catchments with Strahler number up to this limit are processed. If left empty all catchments are processed.

**hdf\_input\_path:** path where HDF5 tables are stored

**timesteps\_per\_day:** number of time steps of input data per day

## Substance transfer module (catchment\_model)

### Layout template

**layout\_xml\_path:** Layout templates are specific for every substance. Parameters that are not subject to change between with the catchments are defined in the template. Other Parameters are marked with a place holder in curly brackets:

* Id: identification number of catchment
* A\_appl: Application area
* R\_appl: application rate
* A\_tot: total catchment area

### Parameter set for catchment model

**param\_txt\_path:** Parameter set for the substance transfer model for a specific substance. Ideally from calibrating with studies where mass of applied substance and measured concentrations at the catchment outlet are reported.

k[substance]: value

Init\_avail[substance]: value

T\_objective[substance]: value

Transfer\_au[substance]: value

Transfer\_ua[substance]: value

Eta[*substance*]: value

sigma[*Conc\_{id}*]: value

C\_back[substance]: value

(In the brackets of each parameter is the name of the substance having no meaning or function for in the framework. Except for the sigma parameter where “Conc\_{id}” must be written which is adapted for in the catchment model for each subcatchment.)

### HDF5 input data

**input\_catchment\_path:** The HDF5 input data is generated during the preprocessing of the model. The files are structured in groups and datasets. Every catchment has its own group containing two dataset. The first dataset is called inputs and stores the time series of the discharge, precipitation and temperature. The second dataset are the parameters for total catchment area, application area and application rate. The time steps are stored in a separate dataset on the same hierarchical level as the catchments.

Programs such as ViTables allow to view the HDF5 groups and datasets.

### HDF5 output data

**output\_catchment\_path:** All results are in a single table with timestep, catchment, concentration, discharge, local\_discharge and load as attributes.

**output\_steps\_path:** In the output\_steps\_path the results are grouped per timestep.

### Other parameters and folder paths

**number\_of\_workers:** The process can be run in parallel. The jobs, meaning the process of one catchment modelling, is distributed to the number of workers. This number can also be larger than the number of CPU’s of your machine. Usually an efficient choice of workers is the number of CPU’s or the double.

**debug:** If debug is ‘true’ (‘yes’, ‘y’) the temporary files (layouts, C++ input/output files) are not deleted and can be found in the temporary folder structured by the workers.

**tmp\_path:** Folder path of the temporary files.

**ratio\_herbicide\_area\_cal (*optional*):** Parameter indicating the ratio of agricultural area to total area in the catchment where the herbicide parameters have been calibrated. The background concentration of the substance transfer model for herbicides reflects the concentration in the soil groundwater which is an important contributor of water and substance at base flow conditions. The concentration in the soil groundwater is assumed to be dependent on the intensity of agricultural land use in the area. With this parameter the actually applied background concentration for each catchment is linearly adapted assuming that a catchment without agricultural land use has a background concentration of 0 a catchment with this ratio has the calibrated background concentration.

## Routing module (routing\_model)

**input\_steps\_path:** Folder path for the HDF5 output file grouped by time steps generated in catchment\_model. The table is grouped by time steps with catchment, concentration, discharge, local\_discharge and load as attributes.

**buffered\_input:** If buffered\_input is ‘true’ (‘yes’, ‘y’), the input file with the buffered lakes is used as input for aggregation by outlets and the generation of the Aquasim file.

### Lake buffering

**lakes:** List of WSO1\_ID’s designating the lake outlets where the substance output in terms of concentration is assumed to be constant due to complete mixing within the lake. The resulting constant concentration is calculated by the division of the total yearly load with the total yearly discharge.

**input\_steps\_buffered\_path:** Output path of lake buffering, but can also be replaced with the input\_steps\_path when lake buffering is not run.

### Aggregation by outlets

**outlet\_list:** List of WSO1\_ID’s of catchments that are considered as outlets and for which the load and the local discharge are added up in the upstream area.

**output\_aggreg\_steps\_path:** Folder path to the HDF5 table where the aggregated results (discharge, load\_aggregated, local\_discharge\_aggregated, outlet) are stored grouped by time steps.

**output\_aggreg\_path:** Folder path to the HDF5 table where the aggregated results (t, discharge, load\_aggrgated, loacal\_discharge\_aggregated) are stored for each catchment in the outlet list.

**csv\_folder:** Folder path where the text-file for each outlet catchment in the outlet list is saved.

### Routing with Aquasim

**riversegments\_path:** dbf file with the river segments of the river network that are modelled as compartments in Aquasim.

|  |  |  |
| --- | --- | --- |
| Attribute | Unit | Description |
| WSO1\_ID | [-] | Unique identification number. For the last river segment the ID must be [-9999] |
| X | [m] | Length coordinate along the river (increasing towards downstream) |
| ELEV | [m] | Elevation of the river at the upstream end |
| WIDTH | [m] | Width of the river segment |
| Kst | [m^(1/3)/s] | Strickler coefficient |
| STRAHLER | [-] | Strahler number |
| MQ | [m3/s] | Average discharge or estimated discharge (serves as Initial condition in Aquasim) |

**nr\_compartments:** number of compartments the modelled river should be divided in.

**output\_aqu\_steps\_path:** HDF5 table with the prepared input data for Aquasim. A table for the lateral and another for the upstream input are available for each time step with the attributes of compartment, discharge and load\_aggregated.   
Furthermore *table\_upstream* and *table\_lateral* describe the relations between the subcatchments and the compartments. The table *links*

**output\_aqu\_compartment\_path:** Path to HDF5 table with lateral/upstream input data, initial conditions and parameterization for each Aquasim compartment.

**output\_aqu\_csv\_path:** Write compartment names and catchments to file specified as argument.

**output\_aqu\_path:** Path for Aquasim file.

Example of a config\_file.ini

# Calculations and model run

To run any of the functions, the repository must be cloned and the PYTHONPATH has to be set.

Command line:

set PYTHONPATH = [crosswater\_path]

## Preprocessing

The preprocessing imports the input data stored in text files and dBASE database files into a HDF5 file which makes it usable and faster accessible for the further steps.

Command line:

python [crosswater\_path]\crosswater\preprocessing\hdf\_input.py config\_file.ini

## Catchment model (substance transfer model)

The substance transfer model (Honti et al. 2016, Moser et al. 2017) is an executable program written in C++. Within this module the program is applied for all subcatchments available in the HDF input data. For efficient computing the simulations are implemented with thread-based parallelism achieving a high CPU utilization. The output is written to a HDF5 table structured by the catchments and another one structured by the time steps.

Command line:

python [crosswater\_path]\crosswater\catchment\_model\run.py config\_file.ini

## Routing model

### Dampening of concentration at lake river outflow

A special situation is given by the presence of large in the river network. Because of the long water residence time in these water bodies, the concentration dynamics in the lake outlet is strongly dampened and differs substantially from other river sections. To account for these different dynamics, the input into each of these lakes is transformed separately with this module. Complete mixing is assumed into one year of discharge and used the resulting concentration as a constant value in the river water flow out of the respective lakes. The load varies accordingly with discharge from the lake.

Concentrations of all subcatchments upstream of the designated lake outlets are averaged over the entire time period and the corresponding time varying load is calculated. Since catchments without pollutant loads contribute to the discharge at the lake outlet, the concentration still shows small variations.

Command line:

python [crosswater\_path]\crosswater\catchment\_model\buffer\_lakes.py config\_file.ini

### Upstream aggregation of loads resulting concentration

For the designated outlets, the loads of all subcatchments upstream are accumulated. The time series of the outlets are written to a HDF5 table and to csv files. This method neglects any transport time or fate of the substance in the river system. If the buffered input is selected, the substance input upstream of the lakes is assumed to be completely mixed resulting in a constant concentration at the lake outlet.

Command line:

python [crosswater\_path]\crosswater\catchment\_model\run\_upstreamaggregation.py config\_file.ini

### Generate AQUASIM file for defined river network

A ready-to-run AQUASIM system file containing the model definitions is created that can be loaded into AQUASIM or run directly with the batch version. For more information how to run AQUASIM refer to the user manual.

The provided river network is divided into the desired number of compartments. Compartments are river reaches that are connected with advective links and may have diffusive links alongside. One compartment exists of one or more river segments and does not contain any junctions. First the river network is first divided at the river junctions, if the river network consists of only one river (for example the Rhine) or the desired number of compartments is higher, the river reaches are further divided at the inlets of the largest tributaries.

The links between compartments are defined as well as the upstream input and the lateral input to the compartments. A compartment can be linked only to a single downstream but may have several compartment upstream. The upstream input describe water and substance flow at the upstream end of a compartment. The lateral input specifies water and substance inflow along a compartment. Discharge and substance loads are aggregated for the accordingly connected subcatchments (see Figure 1).

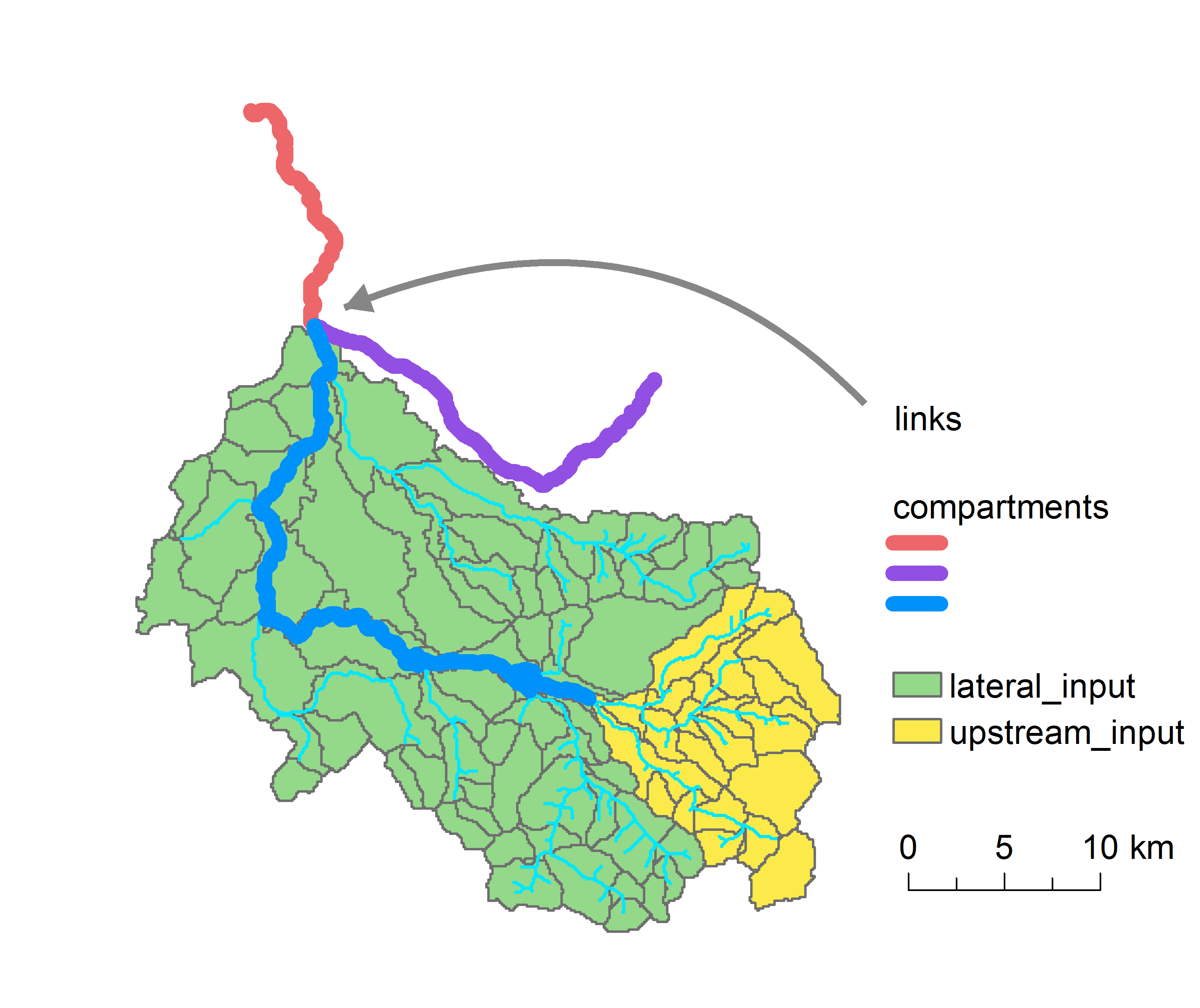


Figure 1: Compartments, links, upstream and lateral input from subcatchments.

The compartments are parameterized with a length coordinate x, the river bed elevation, the river section width and Strickler coefficient Kst. The parameterization is adapted from the input of the river network by discretizing the compartments in the number of river segments it is composed of. Since AQUASIM exhibits difficulties with flat riverbed slopes or even depressions in the riverbed elevations, the slope is rectified by interpolation. Furthermore the compartment length, the riverbed elevation at the start and end of the compartment, and the first length coordinate are identified.

Initial conditions for discharge and runoff depth are allocated to each compartment. These are the mean discharge from the river network input and the runoff depth is derived from the Strahler order.

Finally the model is formulated and the AQUASIM system file is generated. Basis for this are:

* VarSys: Variables (program, constant, real list, state and formula variables)
* ProcSys: Processes (degradation)
* CompSys: River section comparments
* LinkSys: Links (advective links between compartments)
* CalsSys: Definitions for calculations

Changes and adaptions may be conducted directly in the python code or after generation of the system file in the program. More information about the AQUASIM system can be found in the user manual.

Technical AUQUASIM documentation:

Reichert, P., AQUASIM 2.0 - User Manual, Swiss Federal Institute for Environmental Science and Technology (EAWAG), CH-8600 Dübendorf, Switzerland, 214 p., 1998.

Command line:

python [crosswater\_path]\crosswater\catchment\_model\run\_routing.py config\_file.ini

### Run AQUASIM

Run batch version of AQUASIM

# Output

## Preprocessing

The HDF5 output table contains the input and variables for the substance transfer module. For each subcatchment a group entry named “catch\_[WSO1\_ID]” exists containing two tables “parameters” and “inputs”. In “parameters” the value and the supposed unit of the total area (A\_tot), the application area (A\_appl) and the application rate (R\_appl) is stored. In “inputs” time series of temperature, precipitation, discharge, and local discharge of the corresponding subcatchment are available.

Besides the groups are the tables “steps\_per\_day” indication the number of time steps per day in the time series and “time\_steps” in the python numpy data type int64.

The storage path and name is defined in the configuration file with *hdf\_input\_path*.

## Catchment model (substance transfer model)

All results of the substance transfer model are in a first step stored in one large table which is the only table in *output\_catchments\_path*. In the HDF5 file *output\_steps\_path* the results are grouped (“step\_[time\_step]”) by the time steps having each a table “value” with the results of all subcatchments. The following columns are the results of the substance transfer module:

* **concentration**: the concentration within the local discharge or runoff [ng/l]
* **discharge:** total discharge in the subcatchment (equals local\_discharge if subcatchment is a headwater [m3/s]
* **local\_discharge:** the local runoff in the subcatchment or the water originating from precipitation in the subcatchment and finally ending up at the outlet [m3/s]
* **load:** the pollutant mass released at the subcatchment during the time step (calculated by multiplying “concentration” with “local\_discharge”) with the units [g/timestep]

Temporary output of substance transfer module (*tmp\_path*) are the generated text files for the executable program from Honti et al. 2016 (*input\_[WSO\_ID], layout\_[WSO1\_ID], param\_[WSO1\_ID], out\_[WSO1\_ID]*)

## Routing module

### Lake buffering

The output of this module is written to a HDF5 table (*input\_steps\_buffered\_path*) having the same structure as the output from the substance transfer model. All subcatchments above the designated lakes have a yearly averaged concentration output and accordingly to the local runoff calculated load leaving the subcatchment.

### Upstream load aggregation

For each subcatchment where the upstream load aggregation is modelled, a comma delimited text file is issued with the following columns:

* **date**: time in whole numbers starting with 0
* **discharge**: discharge of the river at the considered subcatchment and should be used to calculate the concentration [m3/s]
* **load**: modelled load in [g/timestep]
* **discharge\_aggregated:** the sum of the “local\_discharge” of all upstream subcatchments [m3/s]

The same results are also saved in HDF5 tables.

### AQUASIM system file

The AQUASIM file ready for simulation is written to a text file. This file can be loaded to AQUASIM and the simulations can directly be started. To For further information about the

convedit

### AQUASIM run

Run batch version of AQUASIM.

# Example data

References

Honti, M., Schuwirth, N., Rieckermann, J., & Stamm, C. (2016). Can integrative catchment management mitigate future water quality issues caused by climate change and socio-economic development?. *Hydrol. Earth Syst. Sci.*, 1-28.

Reichert, P. (1994). AQUASIM–a tool for simulation and data analysis of aquatic systems. *Water Science and Technology*, *30*(2), 21-30.