International Institute for Applied Systems Analysis
IIASA www.iiasa.ac.at

Address: IIASA, Schlossplatz 1, A-2361 Laxenburg, Austria

Email: repository@iiasa.ac.at

Report

Community Water Model
CWatM Manual

Peter Burek	burek@iiasa.ac.at
Mikhail Smilovic	smilovic@iiasa.ac.at
Luca Guillaumot	guillaumot@iiasa.ac.at
Jens de Bruijn	debruijn@iiasa.ac.at
Peter Greve	greve@iiasa.ac.at
Yusuke Satoh	satoh.yusuke@nies.go.jp
Adam Islaam	islaam@iiasa.ac.at
Alejandra Virgen-Urcelay	a.urcelay@oceans.ubc.ca
Ting Tang	tangt@iiasa.ac.at
Taher Kahil	kahil@iiasa.ac.at
Yoshihide Wada	wada@iiasa.ac.at

October 2020
Version 1 for CWatM 1.05

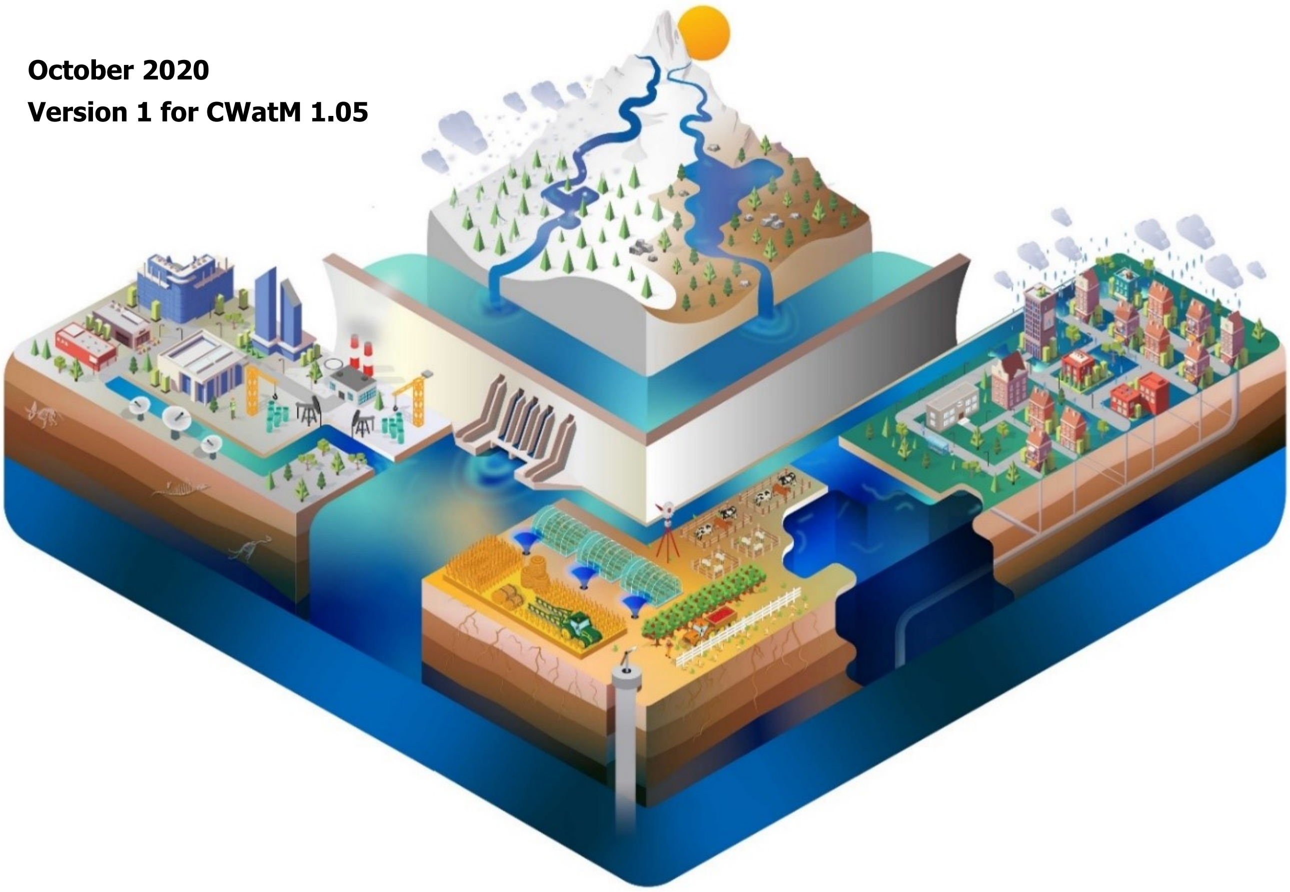


Table of contents

This report represents the web page: https://cwatm.iiasa.ac.at

Done with Sphinx for Python: https://www.sphinx-doc.org

ZVR 524808900

Disclaimer:

The authors gratefully acknowledge funding from IIASA and the National Member Organizations that support the institute (The Austrian Academy of Sciences; The Brazilian Federal Agency for Support and Evaluation of Graduate Education (CAPES); The National Natural Science Foundation of China (NSFC); The Academy of Scientific Research and Technology (ASRT), Egypt; The Finnish Committee for IIASA; The Association for the Advancement of IIASA, Germany; The Technology Information, Forecasting and Assessment Council (TIFAC), India; The Indonesian National Committee for IIASA; The Iran National Science Foundation (INSF); The Israel Committee for IIASA; The Japan Committee for IIASA; The National Research Foundation of Korea (NRF); The Mexican National Committee for IIASA; The Research Council of Norway (RCN); The Russian Academy of Sciences (RAS); Ministry of Education, Science, Research and Sport, Slovakia; The National Research Foundation (NRF), South Africa; The Swedish Research Council for Environment, Agricultural Sciences and Spatial Planning (FORMAS); The Ukrainian Academy of Sciences; The Research Councils of the UK; The National Academy of Sciences (NAS), USA; The Vietnam Academy of Science and Technology (VAST).

The authors gratefully acknowledge funding from Austrian Research Promotion Agency (FFG) under the FUSE project funded by the Belmont Forum (Grant Agreement: 730254), EUCP (European Climate Prediction System) project funded by the European Union under Horizon 2020 (Grant Agreement: 776613), and CO-MICC project which is part of ERA4CS, an ERA-NET initiated by JPI Climate with co-funding by the European Union and the Austrian Federal Ministry of Science, Research and Economy (BMWFW)

Abstract

With a growing population and economic development, it is expected that water demands will increase significantly in the future, especially in developing regions. At the same time, climate change is expected to alter spatial patterns of precipitation and temperature and will have regional to localized impacts on water availability. Thus, it is important to assess water demand, water supply and environmental needs over time to identify the populations and locations that will be most affected by these changes linked to water scarcity, droughts and floods. The Community Water Model will be designed for this purpose in that they include an accounting of how future water demands will evolve in response to socioeconomic change and how water availability will change in response to climate.

CWatM will represent one of the new key elements of the WAT program going forward and increasing the innovative niche of work. We will use and develop the model to work at both global and regional (basin) level. The configuration of the model is open source and community-driven to promote our work amongst the wider water community and is flexible enough to introduce further planned developments such as water quality and hydro-economy.

Our vision for short to medium term work of the group is to introduce water quality (i.e., salinization in deltas and eutrophication associated with mega cities) into the community model and to consider how to include a qualitative/quantitative measure of transboundary river and groundwater governance into a scenario and modelling framework.

About the authors

Peter Burek is Senior Research Scholar at the Water (WAT) Program, International Institute for Applied Systems Analysis (IIASA). (Contact: burek@iiasa.ac.at)

Mikhail Smilovic is Research Scholar at the Water (WAT) Program, International Institute for Applied Systems Analysis (IIASA). (Contact: smilovic@iiasa.ac.at)

Luca Guillaumot is Research Scholar at the Water (WAT) Program, International Institute for Applied Systems Analysis (IIASA). (Contact: guillaumot@iiasa.ac.at)

Jens de Bruijn is Research Scholar at the Water (WAT) Program, International Institute for Applied Systems Analysis (IIASA). (Contact: debruijn@iiasa.ac.at)

Peter Greve is Research Scholar at the Water (WAT) Program, International Institute for Applied Systems Analysis (IIASA). (Contact: greve@iiasa.ac.at)

Yusuke Satoh is former Research Scholar at the Water (WAT) Program, and now at National institute for Environment, Japan. (Contact: satoh.yusuke@nies.go.jp)

Adam Islaam is Graphic Designer at IIASA Communications, International Institute for Applied Systems Analysis (IIASA). (Contact: islaam@iiasa.ac.at)

Alejandra Virgen-Urcelay had an internship at the Water (WAT) Program, International Institute for Applied Systems Analysis (IIASA).

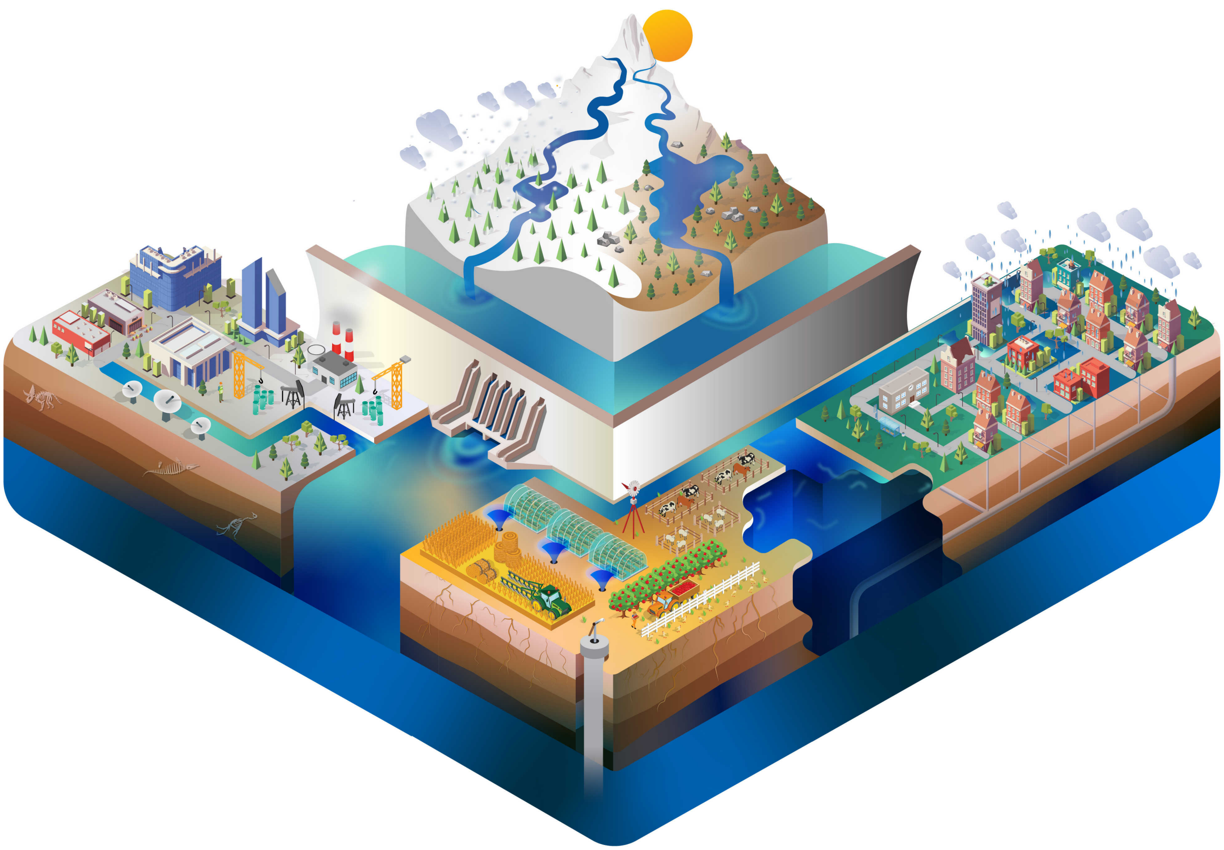
Ting Tang is Research Scholar at the Water (WAT) Program, International Institute for Applied Systems Analysis (IIASA). (Contact: tangt@iiasa.ac.at)

Taher Kahil is Research Scholar at the Water (WAT) Program, International Institute for Applied Systems Analysis (IIASA). (Contact: kahil@iiasa.ac.at)

Yoshihide Wada is Program Director at the Water (WAT) Program, International Institute for Applied Systems Analysis (IIASA). (Contact: wada@iiasa.ac.at)

Acknowledgments

The authors acknowledge the Global Environment Facility (GEF) for funding the development of this research and the CWatM model development as a part of the Integrated Solutions for Water, Energy, and Land (ISWEL) project (GEF Contract Agreement: 6993) and the support of the United Nations Industrial Development Organization (UNIDO). The authors also acknowledge the continuous support of the Asian Development Bank (ADB), the Austrian Development Agency (ADA), and the Austrian Federal Ministry of Sustainability and Tourism to the Water Futures and Solutions (WFaS) initiative at Water Program of IIASA. This study and the model development were also conducted as part of the Belmont Forum Sustainable Urbanisation Global Initiative (SUGI)/Food–Water–Energy Nexus theme for which coordination was supported by the US National Science Foundation under grant ICER/EAR-1829999 to Stanford University. This study is also partly supported by financial support from the Austrian Research Promotion Agency (FFG) under the FUSE project funded by the Belmont Forum (Grant Agreement: 730254), EUCP (European Climate Prediction System) project funded by the European Union under Horizon 2020 (Grant Agreement: 776613), and CO-MICC project which is part of ERA4CS, an ERA-NET initiated by JPI Climate with co-funding by the European Union and the Austrian Federal Ministry of Science, Research and Economy (BMWFW).
Furthermore the authors acknowledge the Sphinx project https://www.sphinx-doc.org/en/master for creating online documentations.



CWatM Documentation

Oct 12, 2020

CHAPTER

ONE

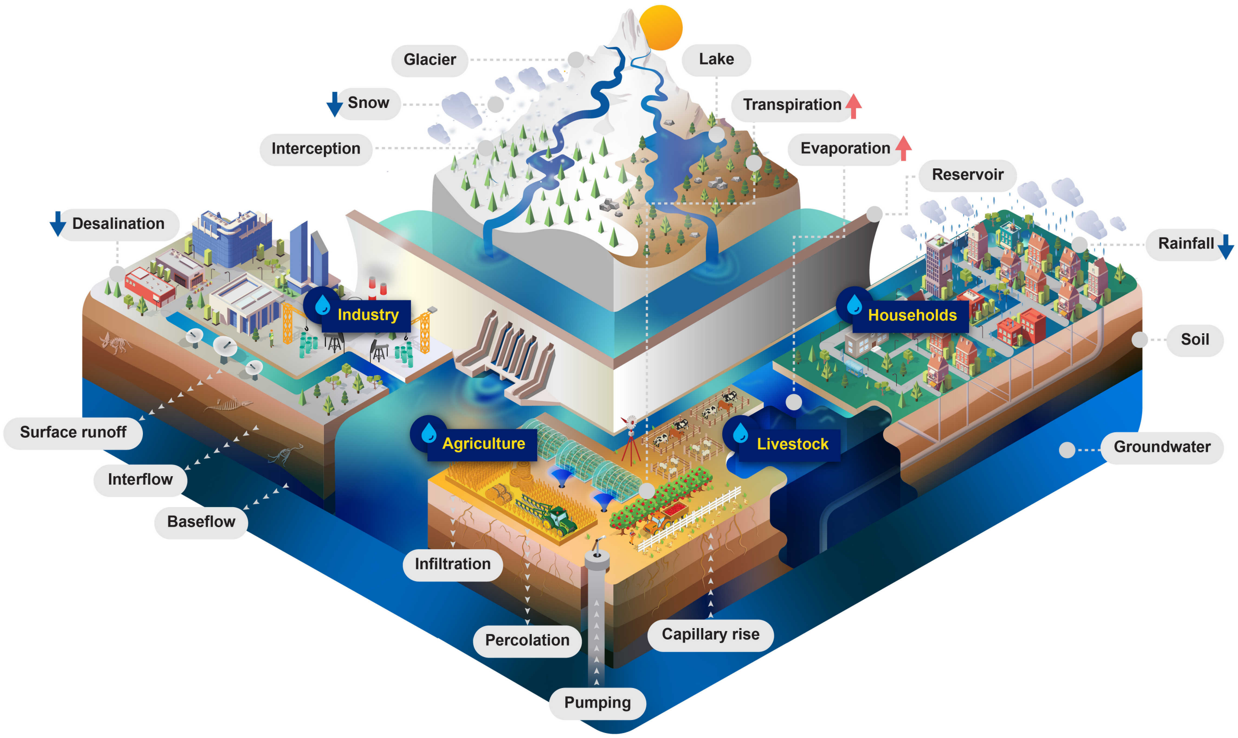
INTRODUCTION

1.1 Community Water Model - CWatM

With a growing population and economic development, it is expected that water demands will increase significantly in the future, especially in developing regions. At the same time, climate change is expected to alter spatial patterns of precipitation and temperature and will have regional to localized impacts on water availability. Thus, it is important to assess water demand, water supply and environmental needs over time to identify the populations and locations that will be most affected by these changes linked to water scarcity, droughts and floods. The Community Water Model will be designed for this purpose in that they include an accounting of how future water demands will evolve in response to socioeconomic change and how water availability will change in response to climate.

CWatM will represent one of the new key elements of the WAT program going forward and increasing the innovative niche of the work. We will use and develop the model to work at both global and regional (basin) level. The configuration of the model is open source and community-driven to promote our work amongst the wider water community and is flexible enough to introduce further planned developments such as water quality and hydro-economy

Our vision for short to medium term work of the group is to introduce water quality (i.e., salinization in deltas and eutrophication associated with mega cities) into the community model and to consider how to include a qualitative/ quantitative measure of transboundary river and groundwater governance into a scenario and modelling framework.



CWatM - Water related processes included in the model design

[1] http://www.iiasa.ac.at/cwatm

CHAPTER

TWO

MODEL DESIGN

Contents

2.1 Background

2.1.1 Water Futures and Solutions Initiatives (WFAS)

Water Futures and Solutions Initiatives[2] is using a multi-model approach for global climatic, hydro-socioeconomic modeling in order to assess possible futures. We use three leading global hydrological models H08, WaterGAP and PCR-GLOBWB for estimating water demand and supply. This approach is used for a better understanding of the uncertainty and limitations of modeling. It provides a degree of confidence in the results an is in-line with the ISI-MIPS[3] approach of multi-modeling

[2] http://www.iiasa.ac.at/web/home/research/water-futures.html
[3] https://www.isimip.org/

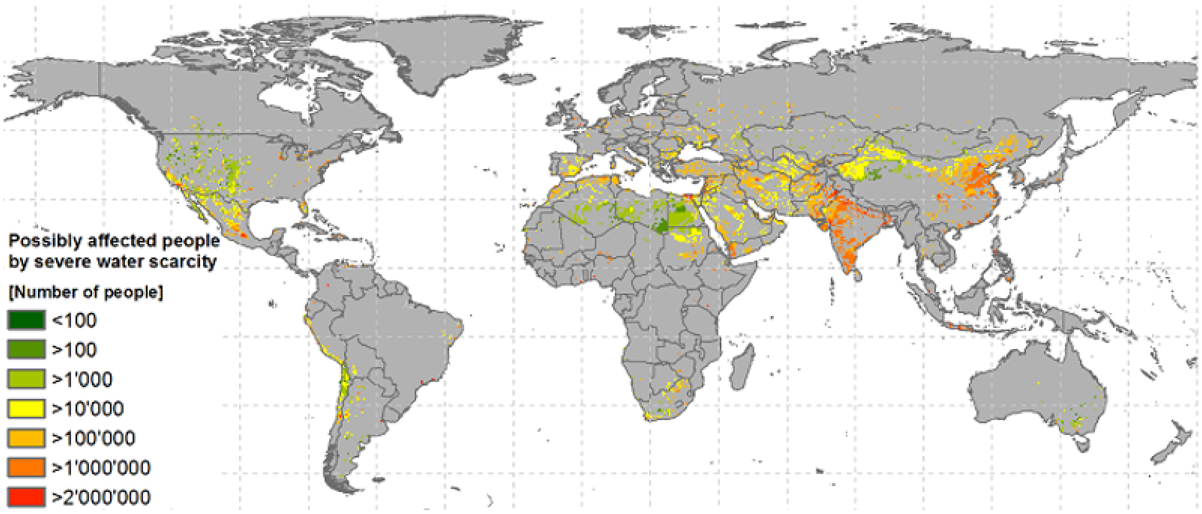


Figure 1: Potential population under severe water scarcity in 2050 - Middle of the Road Scenario - WFAS fast-track analysis[4]

2.1.2 Nexus Integration - Water Energy Food Environment

In the framework of the Integrated Solution project[5]
the Community Water Model (CWatM) will be coupled with the existing IIASA models MESSAGE[6] and GLOBIOM[7] in order to do enhanced water assessments and an improved analysis feedback on water, energy, food and environmental aspects

[4] http://www.iiasa.ac.at/web/scientificUpdate/2015/program/wat/WFaS-fast-track-analysis.html
[5] http://www.iiasa.ac.at/web/home/research/researchProjects/Nexus_Solutions.html
[6] http://www.iiasa.ac.at/web/home/research/modelsData/MESSAGE/MESSAGE.en.html
[7] http://www.globiom.org/

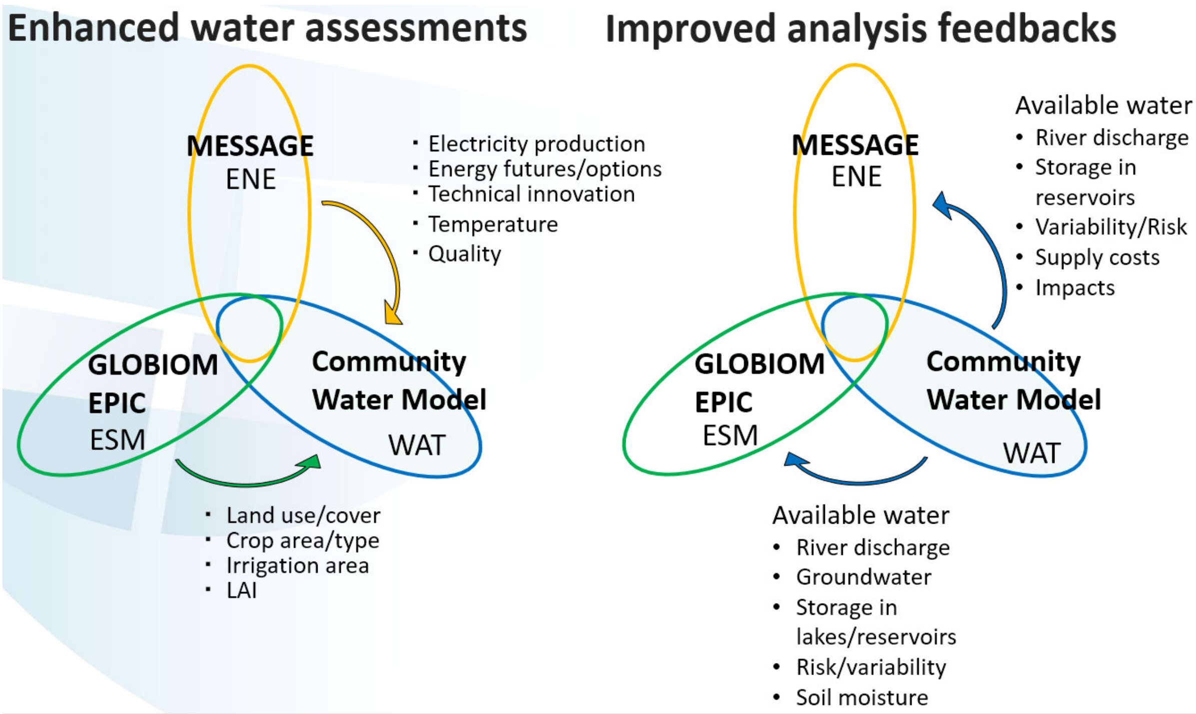


Figure 2: IIASA model interactions

2.2 Model design and processes

2.2.1 Design

The Community Water Model (CWatM) will be designed for the purpose to assess water availability, water demand and environmental needs. It includes an accounting of how future water demands will evolve in response to socioeconomic change and how water availability will change in response to climate.

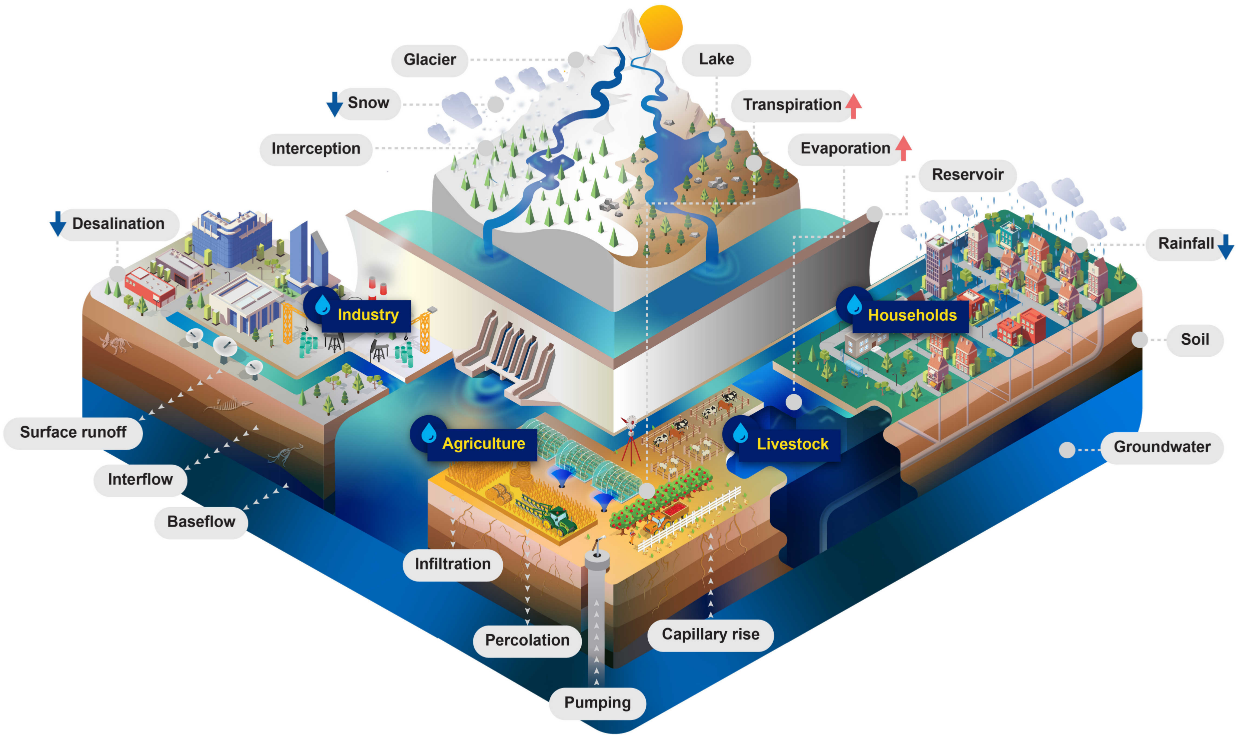


Figure 3: CWatM - Water related processes included in the model design

2.2.2 Processes

Calculation of potential Evaporation

Using Penman-Montheith equations based on FAO 56

Calculation of rain, snow, snowmelt

Using day-degree approach with up to 10 vertical layers Including snow- and glacier melt.

Land cover

using a fraction of 6 different land cover types

- Forest
- Grassland
- Irrigated land
- Paddy irrigated land
- Sealed areas (urban)
- Water

Water demand

- including water demand from industry and domestic land use via precalculated monhly spatial maps
- including agricultural water use from calculation of plant water demand and livestock water demand
- Return flows (water withdrawn but not consumed and returned to the water circle)

Vegetation

Vegetation taken into account for calculating

- Albedo
- Transpiration (including rooting depth, crop phenology, and potential evapotranspiration)
- Interception

Soil

Three soil layers for each land cover type including processes:

- Frost interupting soil processes
- Infiltration
- Preferential flow
- Capillary rise
- Surface runoff
- Interflow
- Percolation into groundwater

Groundwater

Groundwater storage is simulated as linear groundwater reservoir

Lakes & Reservoirs

- Lakes are simulated with weir function from Poleni for rectangular weir.
- Reservoirs are simulated as outflow function between three storage limits (conservative, normal,flood) and three outflow functions (minimum, normal, non-damaging)

Routing

Routing is calculated using the kinematic wave approach

2.3 Features of the Model

2.3.1 Community Model

Feature	Description
Community driven	Open-source but lead by IIASA GitHub repository[8]
Well documented	Documentation, automatic source code documentation GitHub Docu[9]
Easy handling	Use of a setting file with all necessary information for the user *Complete settings file* (page 28) and *Output Meta NetCDF information* (page 42)
Multi-platform	Python 3.x under Windows, Mac, Linux, Unix - to be used on different platforms (PC, clusters, super-computers)
Modular	Processes in Python subprograms, easy to adapt to the requirements of options/ solutions Modular structure[10]

2.3.2 Water Model

Feature	Description
Flexible	different resolution, different processes for different needs, links to other models, across sectors and across scales
Adjustable	to be tailored to the needs at IIASA i.e. collaboration with other programs/models, including solutions and option as part of the model
Multi-disciplinary	including economics, environmental needs, social science perspectives
Sensitive	Sensitive to option / solution
Fast	Global to regional modeling – a mixture between conceptional and physical modeling – as complex as necessary but not more
Comparable	Part of the ISI-MIP community

2.3.3 Demo of first results

Here are some first demonstration of the model run:

Demo of the model (page 107)

[8] https://github.com/CWatM
[9] https://cwatm.github.io
[10] https://github.com/CWatM/CWatM

CHAPTER

THREE

PUBLICATION

Contents

3.1 Please cite when using CWatM

Burek, P., Satoh, Y., Kahil, T., Tang, T., Greve, P., Smilovic, M., Guillaumot, L., Zhao, F., and Wada, Y.: Development of the Community Water Model (CWatM v1.04) - a high-resolution hydrological model for global and regional assessment of integrated water resources management, Geosci. Model Dev., 13, 3267–3298, https://doi.org/10.5194/gmd-13-3267-2020, 2020.

Link to paper in Geoscientific Model Development[11]

3.2 Publication using CWatM

1. Vinca, A., S. Parkinson, E. Byers, P. Burek, Z. Khan, V. Krey, F. A. Diuana, Y. Wang, A. Köberle, I. Staffel, S. Pfenninger, A. Muhammad, A. Rowe, R. Schaeffer, N. Rao, Y. Wada, N. Djilali and K. Riahi (2019). The Nexus Solutions Tool (NEST): An open platform for optimizing multi-scale energy-water-land system transformations.
2. Wang, M., M. Strokal, P. Burek, C. Kroeze, L. Ma and A. B. G. Janssen (2019). "Excess nutrient loads to Lake Taihu: Opportunities for nutrient reduction." Science of the Total Environment 664: 865-873.
3. Wang, M., T. Tang, P. Burek, P. Havlík, T. Krisztin, C. Kroeze, D. Leclère, M. Strokal, Y. Wada, Y. Wang and S. Langan (2019). "Increasing nitrogen export to sea: A scenario analysis for the Indus River." Science of The Total Environment: 133629.

[11] https://gmd.copernicus.org/articles/13/3267/2020

4. He, X., Feng1, K., Li1, X., Craft, A.B., Wada, Y., Burek, P., Wood, E.F., Sheffield, J. Solar and wind energy enhances drought resilience and groundwater sustainability. Nature Communication. Under review
5. Burek, P., Greve, P., Wada, Y. Using the Budyko framework for calibrating a global hydrological model. In preparation
6. He, X., Poledna, S., Burek, P. Kahil, T, Wada, Y. et al. Investigation of drought adaptation options using an integrated hydrological and agent-based model. In preparation

3.3 Presentation CWatM

1. Burek P (2018). Global Hydrological Model Community Water Model (CWatM). In: Indus Basin Knowledge Forum (IBKF), 31 May-2 June 2018, Laxenburg, Austria.
2. Burek P , Kahil T, Parkinson S, Satoh Y, & Wada Y (2018). Integrated modeling for assessing water-energy-land nexus - Application of a hydrological and hydro-economic modeling framework for the Zambezi basin. In: Japan Geoscience Union Meeting 2018, 20-24 May, 2018, Chiba, Japan.
3. Burek P , Satoh Y, Greve P, Kahil T, Byers E , Langan S , & Wada Y (2018). Improving Water Resources Management on Global and Region Scales – Evaluating Strategies for Water Futures with the IIASA's Community Water Model. In: American Geo-Sciences Union Fall Meeting, 11-15 December, New Orleans, USA.
4. Palazzo A , van Dijk M, Willaarts B, Magnuszewski P, Mayor Rodriguez B, Burek P , Kahil T, Tang T , et al. (2018). Integrated solutions for water, energy, and land nexus management the Zambezi Basin: stakeholder engagement and modeling. In: 3rd Zambezi Basin Stakeholders' forum: Water-Energy-Food-Ecosystems (WEFE) Nexus for Socio-Economic Benefits in the Zambezi River Basin, 8-9 October 2018, Lilongwe, Malawi.
5. Tang T , Strokal M, Wada Y , Burek P , Kroeze C, van Vliet M, & Langan S (2018). Sources and export of nutrients in the Zambezi River basin: status and future trend. In: International Conference Water Science for Impact, 16-18 October 2018, Wageningen, Netherlands.
6. Burek P , Satoh Y, Greve P, Kahil T, & Wada Y (2017). The CommunityWater Model (CWatM) / Development of a community driven global water model. In: European Geosciences Union (EGU) General Assembly 2017, 23–28 April 2017, Vienna, Austria.
7. Burek P, Satoh Y, Greve P, Kahil T, Wada Y (2017). The Community Water Model (CWatM) / Development of a community driven global water model. In: European Geosciences Union (EGU) General Assembly 2017, 23–28 April 2017, Vienna, Austria - Poster[12]

3.4 Additional selected publications

1. Burek, P., Y. Satoh, G. Fischer, M. T. Kahil, A. Scherzer, S. Tramberend, L. F. Nava, Y. Wada, S. Eisner, M. Flörke, N. Hanasaki, P. Magnuszewski, B. Cosgrove, D. Wiberg and A. P. D. W. Bill Cosgrove (2016). Water Futures and Solution - Fast Track Initiative (Final Report). IIASA, Laxenburg, Austria.
2. Greve, P., L. Gudmundsson, B. Orlowsky and S. I. Seneviratne (2016). "A two-parameter Budyko function to represent conditions under which evapotranspiration exceeds precipitation." Hydrology and Earth System Sciences 20(6): 2195-2205.
3. Kahil, M. T., S. Parkinson, Y. Satoh, P. Greve, P. Burek, T. I. E. Veldkamp, R. Burtscher, E. Byers, N. Djilali, G. Fischer, V. Krey, S. Langan, K. Riahi, S. Tramberend and Y. Wada (2018). "A Continental-Scale Hydroeconomic Model for Integrating Water-Energy-Land Nexus Solutions." Water Resources Research 54(10): 7511-7533.

[12] http://pure.iiasa.ac.at/14536/1/Cwat_poster.pdf

4. Satoh, Y., T. Kahil, E. Byers, P. Burek, G. Fischer, S. Tramberend, P. Greve, M. Flörke, S. Eisner, N. Hanasaki, P. Magnuszewski, L. F. Nava, W. Cosgrove, S. Langan and Y. Wada (2017). "Multi-model and multi-scenario assessments of Asian water futures: The Water Futures and Solutions (WFaS) initiative." Earth's Future 5(7): 823-852.
5. Tang, T., M. Strokal, M. T. H. van Vliet, P. Seuntjens, P. Burek, C. Kroeze, S. Langan and Y. Wada (2019). "Bridging global, basin and local-scale water quality modeling towards enhancing water quality management worldwide." Current Opinion in Environmental Sustainability 36: 39-48.
6. Tramberend, S., R. Burtscher, P. Burek, T. Kahil, G. Fischer, J. Mochizuki, Y. Wada, R. Kimwaga, P. Nyenje, R. Ondiek, N. Prossie, C. Hyandye, C. Sibomana and S. Langan (2019). East Africa Future Water Scenarios to 2050. IIASA Research Report. Laxenburg, Austria, IIASA.
7. Vinca, A., S. Parkinson, E. Byers, P. Burek, Z. Khan, V. Krey, F. A. Diuana, Y. Wang, A. Köberle, I. Staffel, S. Pfenninger, A. Muhammad, A. Rowe, R. Schaeffer, N. Rao, Y. Wada, N. Djilali and K. Riahi (2019). The Nexus Solutions Tool (NEST): An open platform for optimizing multi-scale energy-water-land system transformations.
8. Wada, Y., M. Flörke, N. Hanasaki, S. Eisner, G. Fischer, S. Tramberend, Y. Satoh, M. T. H. van Vliet, P. Yillia, C. Ringler, P. Burek and D. Wiberg (2016). "Modeling global water use for the 21st century: the Water Futures and Solutions (WFaS) initiative and its approaches." Geoscientific Model Development 9(1): 175-222.
9. Wada, Y., T. Gleeson and L. Esnault (2014). "Wedge approach to water stress." Nature Geosci 7(9): 615-617.

CHAPTER

FOUR

SETUP OF THE MODEL

Contents

4.1 Setup python version

4.1.1 Python

Requirements are a 64-bit Python 3.7.x or 3.8.x version[13]

Warning: a 32-bit version is not able to handle the data requirements!

Warning: CWatM is tested for Python 3.7 and 3.8 and will for sure not work with Python versions lower than 3.6. We recommend using Python 3.7 or 3.8

[13] https://www.python.org/downloads/

4.1.2 External libraries

These external libraries are needed:

- NumPy[14]
- SciPy[15]
- netCDF4[16]
- GDAL[17]
- FloPy[18]
- pytest[19]
- pytest-html[20]

The seven libraries can be installed with conda, pip or downloaded at Unofficial Windows Binaries for Python Extension Packages[21]

> **Warning:** Installing GDAL via pip causes sometimes problems. We recommend downloading the library from Unofficial Windows Binaries for Python Extension Packages[22] as GDAL-3.0.4-cp37-cp37m-win_amd64.whl (or a later version depending on your Python version) and installing them as:
>
> pip install C:/Users/XXXXX/Downloads/GDAL-3.0.4-cp37-cp37m-win_amd64.whl

4.1.3 Installing

Finally the model can be installed with pip.

```
pip install git+git://github.com/CWatM/CWatM
```

or directly downloaded via **'clone or download'** from: https://github.com/CWatM/CWatM

and installing them in a folder.

4.1.4 C++ libraries

For the computational time demanding parts e.g. routing, CWatM comes with a C++ library. A pre-compiled version is included for Windows and Linux. Normally, you don't have to do anything and the pre-compiled version should just work.

Pre-compiled C++ libraries

Windows and CYGWIN_NT-6.1

[14] http://www.numpy.org
[15] https://www.scipy.org
[16] https://pypi.python.org/pypi/netCDF4
[17] http://www.gdal.org
[18] https://www.usgs.gov/software/flopy-python-package-creating-running-and-post-processing-modflow-based-models
[19] https://docs.pytest.org/en/latest/
[20] https://pypi.org/project/pytest-html/
[21] http://www.lfd.uci.edu/~gohlke/pythonlibs
[22] http://www.lfd.uci.edu/~gohlke/pythonlibs

a compiled version is provided and CWatM is detecting automatically which system is running and which compiled version is needed

Linux

For Cygwin linux a compiled version *t5cyg.so* is provided in *../cwatm/hydrological_modules/routing_reservoirs/* for version CYGWIN_NT-6.1.
If you use another cygwin version please compile it by yourself and name it *t5_linux.so*

For Linux Ubuntu a compiled version is provided as *t5_linux.so*. The file is in *../cwatm/hydrological_modules/routing_reservoirs/*

Note: If you use another Linux version or the compiled version is not working or you have a compiler which produce faster executables please compile a version on your own.

Compiling a version

C++ sourcecode is in *../cwatm/hydrological_modules/routing_reservoirs/t5.cpp*

Note: A compiled version is already provided for Windows and Linux.

Windows

A compiled version is provided, but maybe you have a faster compiler than the "Minimalist GNU for Windows" or "Microsoft Visual Studio 14.0" we used.

To compile with g++:

```
..\g++ -c -fPIC -Ofast t5.cpp -o t5.o
..\g++ -shared -Ofast -Wl,-soname,t5.so -o t5.so  t5.o
```

To compile with Microsoft Visual Studio 14.0:

```
call "C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\bin\amd64/vcvars64.bat"
cl /LD /O2 t5.cpp
```

Note:

We used Visual Studio, because it seems to be computational faster

the libray used with Windows is named *t5.dll*, if you generate a libray *t5.so* the filename in **../cwatm/management_modules/globals.py** has to be changed!

Linux

To compile with g++:

```
..\g++ -c -fPIC -Ofast t5.cpp -o t5_linux.o
..\g++ -shared -Ofast -Wl,-soname,t5_linux.so -o t5_linux.so  t5_linux.o

    or
```

(continues on next page)

(continued from previous page)

```
    ..\g++ -c -Ofast t5.cpp -o t5_linux.o
..\g++ -shared -Ofast -Wl,-soname,t5_linux.so -o t5_linux.so  t5_linux.o
```

Warning: Please rename your compiled version to t5_linux.so! At the moment the file t5_linux.so is compiled with Ubuntu Linux

4.1.5 Error and exeption handling

We try to make our program behave properly when encountering unexpected conditions. Therefore we caption a number of possible wrong inputs.

If you get an output with an error number please look at *Error handling* (page 60)

4.1.6 Test the Python version

Run from the command line:

```
cwatm
or
python cwatm.py if you installed CWatM not with pip
```

The output should be:

```
Running under platform:  Windows  **(or Linux etc)**
CWatM - Community Water Model
Authors: ...
Version: ...
Date: ...
```

Warning: If python is not set in the environment path, the full path of python has to be used

Warning: Please use the right version of CWatM with the right version of Python (either 3.7 or 3.8)

4.1.7 Run the Python version

Run from the command line:

```
cwatm settingsfile flags
```

example:

```
cwatm settings1.ini
```

or with more information and an overview of computational runtime:

```
cwatm settings1.ini -l -t
```

Warning: If python is not set in the environment path, the full path of python has to be used

Warning: The model needs a settings file as an argument. See: *Settings file* (page 21)

Flags

Flags can be used to change the runtime output on the screen

example:

```
-q --quiet       output progression given as .
-v --veryquiet   no output progression is given
-l --loud        output progression given as time step, date and discharge
-c --check       input maps and stack maps are checked, output for each input map BUT␣
↪no model run
-h --noheader    .tss file have no header and start immediately with the time series
-t --printtime   the computation time for hydrological modules are printed
    -w --warranty    copyright and warranty information
```

4.2 Windows executable Python version

A CWatM executable cwatm.exe can be used instead of the Python version

- ADVANTAGE: You can run it without installing or knowledge of Python
- DISADVANTAGE 1: You cannot see the source code or change it
- DISADVANTAGE 2: We do not update this version as often as the Python version
- It is done with cx_freeze library
- It includes all Python libraries

Note:

A cwatmexe.zip (around 300 MB with all Python libraries) is stored on:
Source code on Github repository of CWATM[23]
Executable cwatmexe.zip on Github repository of CWATM[24]

Note:

We recommend using the Python 3.7.x version,
but if you not experienced in Python or have problems installing CWATM, please use the executable version.
Either start it in DOS box (command cmd), or use the batch file cwatmbat.bat to start it

[23] https://github.com/CWatM/CWatM
[24] https://github.com/CWatM/CWatM/blob/version091/cwatmexe.zip

---

Todo: We will put a whole example of 30 deg Rhine basin with all necessary data in another zip file. Just for an easier start.

4.3 NetCDF meta data

The format for spatial data for output data is netCDF. In the meta data file information can be added e.g. a description of the parameter

Note: It is not necessary to change this file! This is an option to put additional information into output maps

4.4 Test the data

The model is only as good as the data!
To give out a list of data and to check the data the model can run a check.

example:

```
cwatm settings1.ini -c
or
cwatm settings1.ini -c > checkdata.txt
```

A list is created with:

```
Name:      Name of the variable
Filename:  filename or if the value if it is a fixed value
nonMV:     non missing value in 2D map
MV:        missing value in 2D map
lon-lat:   longitude x latitude of 2D map
CompressV: 2D is compressed to 1D?
MV-comp:   missing value in 1D
Zero-comp: Number of 0 in 1D
NonZero:   Number of non 0 in 1D
min:       minimum in 1D (or 2D)
mean:      mean in 1D (or 2D)
max:       maximum in 1D (or 2D)
```

example:

```
Name                          File/Value                                   nonMV   ␣
↪     MV     lon-lat    Compress    MV-comp   Zero-comp     NonZero          min          mean␣
↪         max
MaskMap                       put5min_netcdf/areamaps/rhine5min.map         5236   ␣
↪      0       68x77       False           0        2404        2832        0.00         0.54␣
↪      1.00
Ldd                           _5min/input5min_netcdf/routing/ldd.nc          5236   ␣
↪      0       68x77       False           0           0        5236        1.00         5.34␣
↪      9.00
```

(continues on next page)

(continued from previous page)

```
Mask+Ldd                                                             2832    ␣
↪     0      68x77       True          0       2832          0        0.00        0.00␣
↪      0.00
CellArea                      n_netcdf/landsurface/topo/cellarea.nc          2832    ␣
↪     0      68x77       True          0          0       2832    5.31E+07    5.63E+07␣
↪   5.94E+07
precipitation_coversion       86.4                                                  -    ␣
↪     -          -          -          -                86.40
evaporation_coversion         1.00                                                  -    ␣
↪     -          -          -          -                 1.00
crop_correct                  1.534                                                 -    ␣
↪     -          -          -          -                 1.53
NumberSnowLayers              7                                                     -    ␣
↪     -          -          -          -                 7.00
GlacierTransportZone          3                                                     -    ␣
↪     -          -          -          -                 3.00
ElevationStD                  min_netcdf/landsurface/topo/elvstd.nc          2832    ␣
↪     0      68x77       True          0          0       2832        0.04       78.67␣
↪     672.68
...
...
```

4.5 Settings file

The settings file is controlling the CWatM run

```
######## ##           ##  ####  ######  ##     ##
##       ##           ## ##  ##   ##   ####  ####
##        ##         ##  ##  ##   ##   ## #### ##
##        ##   ##   ## ########  ##  ##   ##   ##
##         ## #### ##  ##    ##  ##  ##        ##
##         ####  #### ##      ## ## ##          ##
##########  ##    ##  ##      ## ## ##          ##

# Community Water Model Version 0.99
```

4.5.1 Components of the settings file

General flags

General flags are set in the first paragraph For example: If Temperature data are in unit ° Celsius ot Kelvin

```
[OPTIONS]
#-------------------------------------------------------
# OPTION - to switch on/off
#-------------------------------------------------------

# Data otions
# if temperature is stored in Kelvin instead Celsius
TemperatureInKelvin = True
# if lat/lon the area has to be user defined = precalculated
gridSizeUserDefined = True
```

(continues on next page)

(continued from previous page)

```

#-------------------------------------------------
# Evaporation: calculate pot. evaporation (True) or use precalculated pot.evaporation map stacks (False)
calc_evaporation = False

#-------------------------------------------------
# Irrigation and water demand

# if irrigation is included, otherwise paddy and non paddy is put into 'grassland'
includeIrrigation = True
# if water demand from irrigation, industry and domestic is included
includeWaterDemand = False
# Water allocation
# if water demand and availability is calculated for region to compare demand vs. avail
usingAllocSegments = False
# limit abstraction to available groundwater (True) include fossil groundwater (False)
limitAbstraction = False

# Environmental Flow
calc_environflow = False
use_environflow = False

#-------------------------------------------------
# Soil
# use preferential flow, that bypasses the soil matrix and drains directly to the groundwater (not for irrPaddy)
preferentialFlow = False
# Capillar rise
CapillarRise = True

#-------------------------------------------------
# Routing

# if runoff concentration to the edge of a cell is included
includeRunoffConcentration = True
# Waterbodies like lakes and reservoirs
includeWaterBodies = True
# kinematic wave routing, if False no routing is calculated
includeRouting = True

#-------------------------------------------------
# Inflow from outside of the modelled area
inflow = False

# --- Reporting & Debugging ------------------- ---------------------------------------------
# Reporting options
writeNetcdfStack = True
reportMap = True
reportTss = True
# Checking water balance (for debugging)
calcWaterBalance = False
sumWaterBalance = False
# use additional PCRaster GIS commands
PCRaster = False
```

(continues on next page)

(continued from previous page)

```
#-------------------------------------------------------
# DEFINITIONS OF PARAMETERS
```

NetCDF meta data

The format for spatial data for input and output data is netCDF. For output data the basic information are given in the settingsfile

```
[NETCDF_ATTRIBUTES]
institution = IIASA
title = Global Water Model - WATCH WDFEI
metaNetcdfFile = $(FILE_PATHS:PathRoot)/source/metaNetcdf.xml
```

For each output file the specific information about units, variable name, displayed variable name is given in the metaNetcdf.xml. See: *Output Meta NetCDF information* (page 42)

Path of data, output

Note: Further on the pathes can be used as placeholders

```
#-------------------------------------------------------
[FILE_PATHS]
#-------------------------------------------------------
PathRoot = E:/CWATM_rhine

PathOut  = $(PathRoot)/output
PathMaps  = $(PathRoot)/cwatm_input
PathMeteo  = $(PathRoot)/climate
```

Defining the modeling area

In general the input data are stored and used at global scale. The modeling area can be defined by:

- a mask map e.g.: $(FILE_PATHS:PathRoot)/source/rhine30min.tif
- coordinates e.g.: 14 12 0.5 5.0 52.0
- lowest point of a catchment e.g.: 6.25 51.75

Note:

The mask map can be a .tif, PCraster or a netCDF format

The coordinates have the format: Number of Cols, Number of rows, cellsize, upper left corner X, upper left corner Y

The point location (lon lat) will be used to create the catchment upstream of this point

---

Warning: If you use a mask map, make sure you do not use blanks in the file path or name!

```
# AREA AND OUTLETS
#-------------------------------------------------------
[MASK_OUTLET]

# Area mask
# A pcraster map, tif or netcdf map e.g.  $(FILE_PATHS:PathRoot)/data/areamaps/area_indus.map
# or a retancle: Number of Cols, Number of rows, cellsize, upper left corner X, upper left corner Y
MaskMap = $(FILE_PATHS:PathRoot)/source/rhine30min.tif
#MaskMap = 14 12 0.5 5.0 52.0


#-------------------------------------------------------
# Station data
# either a map e.g. $(FILE_PATHS:PathRoot)/data/areamaps/area3.map
# or a location coordinates (X,Y) e.g. 5.75 52.25 9.25 49.75 )
# Lobith/Rhine
Gauges = 6.25 51.75

# if .tif file for gauges, this is a flag if the file is global or local
# e.g. Gauges = $(FILE_PATHS:PathRoot)/data/areamaps/gaugesRhine.tif
GaugesLocal = True
```

Defining the time

The start and end time have to be defined. Spin-up time is the time for warming up (results will be stored after the spin-up time)

Note: The time can be given as date: dd/mm/yyyy or as relative date: number (but then CalendarDayStart has to be defined)

Note: Spin-up time can be given as date or number

```
#-------------------------------------------------------
[TIME-RELATED_CONSTANTS]
#-------------------------------------------------------

# StepStart has to be a date e.g. 01/06/1990
# SpinUp or StepEnd either date or numbers
# SpinUp: from this date output is generated (up to this day: warm up)

StepStart = 1/1/1990
SpinUp =  1/01/1995
StepEnd =  31/12/2010
```

Initial conditions

Initial conditions can be stored and be loaded in order to initialise a warm start of the model

Note: Initial conditions are store as one netCDF file with all necessary variables

```
#-------------------------------------------------------
[INITIAL CONDITIONS]
#-------------------------------------------------------

# for a warm start initial variables a loaded
# e.g for a start on 01/01/2010 load variable from 31/12/2009
load_initial = False
initLoad = $(FILE_PATHS:PathRoot)/init/Rhine_19891231.nc

# saving variables from this run, to initiate a warm start next run
# StepInit = saving date, can be more than one: 10/01/1973 20/01/1973
save_initial = False
initSave = $(FILE_PATHS:PathRoot)/init/Rhine
StepInit = 31/12/1989 31/12/2010
```

StepInit indicate the date(s) when initial conditions are saved:

```
StepInit = 31/12/1989
StepInit = 31/12/1989 31/12/2010
StepInit = 31/12/1989 5y
here: second value in StepInit is indicating a repetition of year(y), month(m) or day(d),
e.g. 2y for every 2 years or 6m for every 6 month
```

Output

Output can be spatial/time as netCDF4 map stacks

and/or time series at specified points

Note: For additional information see *Model Output* (page 47)

Output can be as maps and time series:

- per day [Daily]
- total month [MonthTot], average month [MonthAvg], end of month [MonthEnd]
- total year [AnnualTot], average year [AnnualAvg], end of year [AnnualEnd]
- total sum [TotalTot], total average [TotalAvg]

For each of the following sections output can be defined for different variables:

- Meteo
- Snow
- Soil for different land cover (forest, grassland, irrigated land, paddy irrigated)
- Water demand

- Groundwater
- River routing
- Lakes and reservoirs

Or output can be defined in the section *[output]*

An output directory can be defined and for each sort of output the variable(s) can be set:

*OUT_* defines that this variable(s) are output
*MAP_* or *TSS_* defines if it is a spatial map or a time series of point(s)
*AreaSum_* or *AreaAvg_* after *TSS_* defines if the catchment sum of average upstream of the point is calculated
Daily or *MonthAvg* or .. is specifying the time
The variable is given after the equal sign e.g. * = discharge*
If more than one variable should be used for output, split with **,**
E.g. OUT_MAP_Daily = discharge -> daily spatial map of discharge

As example output for precipitation, temperature and discharge is shown here:

```
# OUTPUT maps and timeseries
OUT_Dir = $(FILE_PATHS:PathOut)
OUT_MAP_Daily =
OUT_MAP_MonthEnd =
OUT_MAP_MonthTot = Precipitation, Tavg
OUT_MAP_MonthAvg =

OUT_TSS_MonthTot = Precipitation, Tavg   # monthly total precipitation and average temperature
OUT_TSS_Daily = discharge                # daily discharge
OUT_TSS_MonthEnd = discharge
OUT_TSS_AnnualEnd = discharge

OUT_TSS_AreaSum_Daily = Precipitation    # daily sum of precipitation for the upstream catchment
OUT_TSS_AreaAvg_MonthAvg = runoff        # monthly average sum of runoff for the upstream catchment
```

Note: For each variable the meta data information can be defined in *Output Meta NetCDF information* (page 42)

Reading information

Information will be read in from values in the settings file Here the value definitions for [SNOW] is shown:

```
#-------------------------------------------------------
[SNOW]
#-------------------------------------------------------

# Number of vertical Snow layers
NumberSnowLayers = 7
# up to which layer the ice melt is calculated with the middle temperature
GlacierTransportZone = 3
```

(continues on next page)

(continued from previous page)

```

# Temperature lapse rate with altitude [deg C / m]
TemperatureLapseRate = 0.0065
# Multiplier applied to precipitation that falls as snow
SnowFactor = 1.0
# Range [m C-1 d-1] of the seasonal variation, SnowMeltCoef is the average value
SnowSeasonAdj = 0.001
# Average temperature at which snow melts
TempMelt =1.0
# Average temperature below which precipitation is snow
TempSnow = 1.0
# Snow melt coefficient: default: 4.0
# SRM: 0.0045 m/C/day ( = 4.50 mm/C/day), Kwadijk: 18 mm/C/month (= 0.59 mm/C/day)
# See also Martinec et al., 1998.

# use in CALIBRATION -> copied to CALIBRATION
#SnowMeltCoef = 0.004
IceMeltCoef  = 0.007

#------------------------------------------------------------
# INITIAL CONDITIONS - Initial snow depth in snow zone 1-7 [mm]  - SnowCoverIni

[FROST]
# Snow water equivalent, (based on snow density of 450 kg/m3) (e.g. Tarboton and Luce,
↪ 1996)
SnowWaterEquivalent = 0.45
# Daily decay coefficient, (Handbook of Hydrology, p. 7.28)
Afrost = 0.97
# Snow depth reduction coefficient, [cm-1], (HH, p. 7.28)
Kfrost = 0.57
# Degree Days Frost Threshold (stops infiltration, percolation and capillary rise)
# Molnau and Bissel found a value 56-85 for NW USA.
FrostIndexThreshold = 56
```

Note: TemperatureLapseRate = 0.0065 | for the variable TemperatureLapseRate the value of 0.0065 is set

Variables can also be defined by spatial maps or map stacks

```
tanslope = $(PathTopo)\tanslope.map
forest_coverFractionNC    = $(PathForest)\coverFractionInputForest366days.nc
```

Note: suffix can be .map, but if there is no PCraster map it will look automatically for netCDF .nc

Warning: in most cases values can be replaced by map

4.5.2 Sections of information

- Snow
- Frost
- General information on land cover types
- Soil
- **Information for each of the six land cover types**
  - Forest
  - Grassland
  - Paddy irrigated area
  - Irrigated area
  - Sealed area
  - Water covered area
- Interflow
- Groundwater
- Water demand
- Runoff concentration
- Routing
- Lakes and reservoirs
- Inflow

4.5.3 Complete settings file

Example of a settings file:

```
# ----------------------------------------------

######## ##          ##  ####  ######  ##    ##
##       ##          ## ##  ##   ##   ####  ####
##        ##        ##  ##  ##   ##   ## #### ##
##        ##   ##   ## ########  ##  ##   ##   ##
##         ## #### ##  ##    ##  ##  ##        ##
##         ####  #### ##      ## ## ##          ##
##########  ##    ##  ##      ## ## ##          ##

# Community Water Model Version 0.99
# SETTINGS FILE
# ----------------------------------------------


[OPTIONS]
#----------------------------------------------------
# OPTION - to switch on/off
#----------------------------------------------------

# Data otions
```

(continues on next page)

(continued from previous page)

```
# if temperature is stored in Kelvin instead Celsius
TemperatureInKelvin = True
# if lat/lon the area has to be user defined = precalculated
gridSizeUserDefined = True

#-----------------------------------------------
# Evaporation: calculate pot. evaporation (True) or use precalculated pot.evaporation map stacks (False)
calc_evaporation = False

#-----------------------------------------------
# Irrigation and water demand

# if irrigation is included, otherwise paddy and non paddy is put into 'grassland'
includeIrrigation = True
# if water demand from irrigation, industry and domestic is included
includeWaterDemand = False
# Water allocation
# if water demand and availability is calculated for region to compare demand vs. avail
usingAllocSegments = False
# limit abstraction to available groundwater (True) include fossil groundwater (False)
limitAbstraction = False

# Environmental Flow
calc_environflow = False
use_environflow = False

#-----------------------------------------------
# Soil
# use preferential flow, that bypasses the soil matrix and drains directly to the groundwater (not for irrPaddy)
preferentialFlow = False
# Capillar rise
CapillarRise = True

#-----------------------------------------------
# Routing

# if runoff concentration to the edge of a cell is included
includeRunoffConcentration = True
# Waterbodies like lakes and reservoirs
includeWaterBodies = True
# kinematic wave routing, if False no routing is calculated
includeRouting = True

#-----------------------------------------------
# Inflow from outside of the modelled area
inflow = False

# --- Reporting & Debugging ------------------- ---------------------------------------------
# Reporting options
writeNetcdfStack = True
reportMap = True
reportTss = True
# Checking water balance (for debugging)
```

(continues on next page)

(continued from previous page)

```
calcWaterBalance = False
sumWaterBalance = False
# use additional PCRaster GIS commands
PCRaster = False






#-------------------------------------------------------
# DEFINITIONS OF PARAMETERS
#-------------------------------------------------------

#-------------------------------------------------------
[FILE_PATHS]
#-------------------------------------------------------
PathRoot = E:/CWATM_rhine

PathOut = $(PathRoot)/output
PathMaps = $(PathRoot)/cwatm_input
PathMeteo = $(PathRoot)/climate




#-------------------------------------------------------
[NETCDF_ATTRIBUTES]
institution = IIASA
title = Global Water Model - WATCH WDFEI
metaNetcdfFile = $(FILE_PATHS:PathRoot)/source/metaNetcdf.xml

#-------------------------------------------------------
# AREA AND OUTLETS
#-------------------------------------------------------
[MASK_OUTLET]

# Area mask
# A pcraster map, tif or netcdf map e.g.  $(FILE_PATHS:PathRoot)/data/areamaps/area_
↪indus.map
# or a retancle: Number of Cols, Number of rows, cellsize, upper left corner X, upper␣
↪left corner Y
MaskMap = $(FILE_PATHS:PathRoot)/source/rhine30min.tif
#MaskMap = 14 12 0.5 5.0 52.0


#-------------------------------------------------------
# Station data
# either a map e.g. $(FILE_PATHS:PathRoot)/data/areamaps/area3.map
# or a location coordinates (X,Y) e.g. 5.75 52.25 9.25 49.75 )
# Lobith/Rhine
Gauges = 6.25 51.75

# if .tif file for gauges, this is a flag if the file is global or local
# e.g. Gauges = $(FILE_PATHS:PathRoot)/data/areamaps/gaugesRhine.tif
GaugesLocal = True

```

(continues on next page)

(continued from previous page)

```
#-------------------------------------------------------
[TIME-RELATED_CONSTANTS]
#-------------------------------------------------------

# StepStart has to be a date e.g. 01/06/1990
# SpinUp or StepEnd either date or numbers
# SpinUp: from this date output is generated (up to this day: warm up)

StepStart = 1/1/1990
SpinUp =  1/01/1995
StepEnd =  31/12/2010




#-------------------------------------------------------
[INITITIAL CONDITIONS]
#-------------------------------------------------------

# for a warm start initial variables a loaded
# e.g for a start on 01/01/2010 load variable from 31/12/2009
load_initial = False
initLoad = $(FILE_PATHS:PathRoot)/init/Rhine_19891231.nc

# saving variables from this run, to initiate a warm start next run
# StepInit = saving date, can be more than one: 10/01/1973 20/01/1973
save_initial = False
initSave = $(FILE_PATHS:PathRoot)/init/Rhine
StepInit = 31/12/1989 31/12/2010

#-------------------------------------------------------
# CALIBARTION PARAMETERS
#-------------------------------------------------------
[CALIBRATION]

# These are parameter which are used for calibration
# could be any parameter, but for an easier overview, tehey are collected here
# in the calibration template a placeholder (e.g. %arnoBeta) instead of value

# Snow
SnowMeltCoef = 0.0027
# Cropf factor correction
crop_correct =  1.11
#Soil
soildepth_factor = 1.28
#Soil preferentialFlowConstant = 4.0, arnoBeta_add = 0.1
preferentialFlowConstant = 4.5
arnoBeta_add = 0.19
# interflow part of recharge factor = 1.0
factor_interflow = 2.8
# groundwater recessionCoeff_factor = 1.0
recessionCoeff_factor = 5.278
# runoff concentration factor runoffConc_factor = 1.0
runoffConc_factor = 0.1
#Routing manningsN Factor to Manning's roughness = 1.0 [0.1-10.]
manningsN = 1.86
# reservoir  normal storage limit (fraction of total storage, [-]) [0.15 - 0.85]␣
↪default 0.5
```

(continues on next page)

(continued from previous page)

```
normalStorageLimit = 0.44
# lake parameter - factor to alpha: parameter of of channel width and weir␣
↪coefficient  [0.33 - 3.] dafault 1.
lakeAFactor = 0.33
# lake parameter - factor for wind evaporation
lakeEvaFactor = 1.52
#-------------------------------------------------------
# TOPOGRAPHY MAPS
#-------------------------------------------------------
[TOPOP]
# local drain direction map (1-9)
Ldd = $(FILE_PATHS:PathMaps)/routing/ldd.map

# Elevation standard deviation [m], i.e. altitude difference elevation within pixel.
# Used for sub-pixel modelling of snow accumulation and melt
ElevationStD = $(FILE_PATHS:PathMaps)/landsurface/topo/elvstd.map

# Area of pixel [m2] (for lat/lon every cell has a different area)
CellArea = $(FILE_PATHS:PathMaps)/routing/cellarea.map

#-------------------------------------------------------
# INPUT METEOROLOGICAL TIMESERIES AS MAPS
#-------------------------------------------------------
[METEO]
# precipitation [kg m-2 s-1]
#PrecipitationMaps = $(FILE_PATHS:PathMeteo)/pr*
PrecipitationMaps = $(FILE_PATHS:PathMeteo)/30min/pr_rhine*
# average daily temperature [K]
#TavgMaps = $(FILE_PATHS:PathMeteo)/tavg*
TavgMaps = $(FILE_PATHS:PathMeteo)/30min/tavg_rhine*

# ----------------------------------------
# This is used if calc_evaporation = False

# daily reference evaporation (free water)
E0Maps = $(FILE_PATHS:PathMeteo)/30min/EWRef_rhine.nc
#E0Maps = $(FILE_PATHS:PathMeteo)/EWRef_daily*
# daily reference evapotranspiration (crop)
ETMaps = $(FILE_PATHS:PathMeteo)/30min/ETRef_rhine.nc
#ETMaps = $(FILE_PATHS:PathMeteo)/ETRef_daily*

# --------------------------------
# from kg m-2s-1 to m : 86.4
precipitation_coversion = 86.4

# from MM to m : 0.001
#precipitation_coversion = 0.001

evaporation_coversion = 1.00

# OUTPUT maps and timeseries
#OUT_Dir = $(FILE_PATHS:PathOut)
#OUT_MAP_Daily = Precipitation, prec1

#-------------------------------------------------------
# CALCULATE EVAPORATION - PENMAN - MONTEITH
#-------------------------------------------------------
```

(continues on next page)

(continued from previous page)

```
[EVAPORATION]

# This is used if calc_evaporation = True
# use albedo maps
albedo = True
albedoMaps = $(FILE_PATHS:PathMaps)/landsurface/albedo/albedo.nc

# if not albedo maps use fixed albedo
# Albedo of bare soil surface (Supit et. al.)
AlbedoSoil = 0.15
# Albedo of water surface (Supit et. al.)
AlbedoWater = 0.05
# Albedo of vegetation canopy (FAO,1998)
AlbedoCanopy = 0.23

# use specific humidity (TRUE) QAir,  or relative humidity (FALSE) - rhs
useHuss = False

# map stacks Temperature [K}]
TminMaps = $(FILE_PATHS:PathMeteo)/tmin*
TmaxMaps = $(FILE_PATHS:PathMeteo)/tmax*
# Instantaneous surface pressure[Pa]
PSurfMaps = $(FILE_PATHS:PathMeteo)/ps*
# 2 m istantaneous specific humidity[kg /kg] (QAir) or relative humidity [%] (rhs)
RhsMaps = $(FILE_PATHS:PathMeteo)/hurs*
# wind speed maps at 10m [m/s]
WindMaps = $(FILE_PATHS:PathMeteo)/wind*
# radiation surface downwelling shortwave maps  [W/m2]
RSDSMaps = $(FILE_PATHS:PathMeteo)/rsds*
# radiation surface downwelling longwave maps [W/m2] [W/m2]
RSDLMaps = $(FILE_PATHS:PathMeteo)/rlds*

# OUTPUT maps and timeseries
#OUT_Dir = $(FILE_PATHS:PathOut)
#OUT_MAP_Daily = EWRef, ETRef, temp, prec

#-------------------------------------------------------
[SNOW]
#-------------------------------------------------------

# Number of vertical Snow layers
NumberSnowLayers = 7
# up to which layer the ice melt is calculated with the middle temperature
GlacierTransportZone = 3

# Temperature lapse rate with altitude [deg C / m]
TemperatureLapseRate = 0.0065
# Multiplier applied to precipitation that falls as snow
SnowFactor = 1.0
# Range [m C-1 d-1] of the seasonal variation, SnowMeltCoef is the average value
SnowSeasonAdj = 0.001
# Average temperature at which snow melts
TempMelt =1.0
# Average temperature below which precipitation is snow
TempSnow = 1.0
# Snow melt coefficient: default: 4.0
# SRM: 0.0045 m/C/day ( = 4.50 mm/C/day), Kwadijk: 18 mm/C/month (= 0.59 mm/C/day)
```

(continues on next page)

(continued from previous page)

```
# See also Martinec et al., 1998.

# use in CALIBRATION -> copied to CALIBRATION
#SnowMeltCoef = 0.004
IceMeltCoef  = 0.007

#-----------------------------------------------------------
# INITIAL CONDITIONS - Initial snow depth in snow zone 1-7 [mm]  - SnowCoverIni

[FROST]
# Snow water equivalent, (based on snow density of 450 kg/m3) (e.g. Tarboton and Luce,
↪ 1996)
SnowWaterEquivalent = 0.45
# Daily decay coefficient, (Handbook of Hydrology, p. 7.28)
Afrost = 0.97
# Snow depth reduction coefficient, [cm-1], (HH, p. 7.28)
Kfrost = 0.57
# Degree Days Frost Threshold (stops infiltration, percolation and capillary rise)
# Molnau and Bissel found a value 56-85 for NW USA.
FrostIndexThreshold = 56

#-----------------------------------------------------------
# INITIAL CONDITIONS: FrostIndexIni

[VEGETATION]
cropgroupnumber = $(FILE_PATHS:PathMaps)/others/cropgrp.nc
# soil water depletion fraction, Van Diepen et al., 1988: WOFOST 6.0, p.86, Doorenbos
↪et. al 1978

#-------------------------------------------------------
[SOIL]
#-------------------------------------------------------

PathTopo = $(FILE_PATHS:PathMaps)/landsurface/topo
PathSoil = $(FILE_PATHS:PathMaps)/landsurface/soil
PathSoil1 =  $(FILE_PATHS:PathMaps)/others

# Topography - tangent slope, slope length
tanslope = $(PathTopo)/tanslope.map
slopeLength = $(PathTopo)/slopeLength.map

# maps of relative elevation above flood plains
relativeElevation = $(PathTopo)/dzRel_hydro1k.nc

# Soil hydraulic properties

# soil (Hypres pedotransfer function - http://esdac.jrc.ec.europa.eu/ESDB_Archive/
↪ESDBv2/popup/hy_param.htm)
KSat1 = $(PathSoil1)/ksat1.map
KSat2 = $(PathSoil1)/ksat2.map
KSat3 = $(PathSoil1)/ksat3.map
# Alpha: an Genuchten's shape parameter
alpha1 = $(PathSoil1)/alpha1.map
alpha2 = $(PathSoil1)/alpha2.map
alpha3 = $(PathSoil1)/alpha3.map
#Lambda: an Genuchten's shape parameter = n-1-> n = lamda+1, m = 1 - (1/n)
lambda1 = $(PathSoil1)/lambda1.map
```

(continues on next page)

(continued from previous page)

```
lambda2 = $(PathSoil1)/lambda2.map
lambda3 = $(PathSoil1)/lambda3.map
# thetas  is the volumetric water content  saturated
thetas1 = $(PathSoil1)/thetas1.map
thetas2 = $(PathSoil1)/thetas2.map
thetas3 = $(PathSoil1)/thetas3.map
# thetar is the volumetric water content  residual
thetar1 = $(PathSoil1)/thetar1.map
thetar2 = $(PathSoil1)/thetar2.map
thetar3 = $(PathSoil1)/thetar3.map

percolationImp = $(PathSoil)/percolationImp.map

maxGWCapRise    = 5.0

minCropKC        = 0.2
minTopWaterLayer = 0.0

# Soil depth
StorDepth1 = $(PathSoil)/storageDepth1.map
StorDepth2 = $(PathSoil)/storageDepth2.map

# preferential flow (between 1.0 and 8.0)
# used in CALIBRATION -> copied to CALIBRATION
#preferentialFlowConstant = 4.0

#-------------------------------------------------------
[LANDCOVER]
PathLandcover = $(FILE_PATHS:PathMaps)/landsurface

coverTypes = forest, grassland, irrPaddy, irrNonPaddy, sealed, water
coverTypesShort = f, g, i, n, s, w
fractionLandcover = $(PathLandcover)/fractionLandcover.nc

# Landcover can vary from year to year
dynamicLandcover = True
# if landcover cannot vary, which year should be taken as fixed year
fixLandcoverYear = 1961

#-------------------------------------------------------

[__forest]
PathForest = $(FILE_PATHS:PathMaps)/landcover/forest
PathSoil1 =  $(FILE_PATHS:PathMaps)/others

# Parameters for the Arno's scheme
# arnoBeta is defined by orographic,+ land cover add + calibration add, the soil␣
↪water capacity distribution is based on this
# range [0.01 - 1.2]
forest_arnoBeta = 0.2

#forest_soil
forest_KSat1 = $(PathSoil1)/forest_ksat1.map
forest_KSat2 = $(PathSoil1)/forest_ksat2.map
forest_KSat3 = $(PathSoil1)/ksat3.map
forest_alpha1 = $(PathSoil1)/forest_alpha1.map
forest_alpha2 = $(PathSoil1)/forest_alpha2.map
```

(continues on next page)

(continued from previous page)

```
forest_alpha3 = $(PathSoil1)/alpha3.map
forest_lambda1 = $(PathSoil1)/forest_lambda1.map
forest_lambda2 = $(PathSoil1)/forest_lambda2.map
forest_lambda3 = $(PathSoil1)/lambda3.map
forest_thetas1 = $(PathSoil1)/forest_thetas1.map
forest_thetas2 = $(PathSoil1)/forest_thetas2.map
forest_thetas3 = $(PathSoil1)/thetas3.map
forest_thetar1 = $(PathSoil1)/forest_thetar1.map
forest_thetar2 = $(PathSoil1)/forest_thetar2.map
forest_thetar3 = $(PathSoil1)/thetar3.map

# other paramater values
forest_minInterceptCap  = 0.001
forest_cropDeplFactor   = 0.0

forest_fracVegCover  = $(PathForest)/fracVegCover.map
forest_rootFraction1 = $(PathForest)/rootFraction1.map
forest_rootFraction2 = $(PathForest)/rootFraction2.map
#forest_maxRootDepth  = 2.0
forest_maxRootDepth  = $(PathForest)/maxRootDepth.map
forest_minSoilDepthFrac = $(PathForest)/minSoilDepthFrac.map


forest_cropCoefficientNC = $(PathForest)/CropCoefficientForest_10days.nc
forest_interceptCapNC    = $(PathForest)/interceptCapForest10days.nc

# initial conditions: forest_interceptStor, forest_w1, forest_w2, forest_w3,


[__grassland]
PathGrassland = $(FILE_PATHS:PathMaps)/landcover/grassland

# Parameters for the Arno's scheme:
grassland_arnoBeta = 0.0
# arnoBeta is defined by orographic,+ land cover add + calibration add, the soil water capacity distribution is based on this
# range [0.01 - 1.2]

# other paramater values

grassland_minInterceptCap  = 0.001
grassland_cropDeplFactor   = 0.0

grassland_fracVegCover  = $(PathGrassland)/fracVegCover.map
grassland_rootFraction1 = $(PathGrassland)/rootFraction1.map
grassland_rootFraction2 = $(PathGrassland)/rootFraction2.map
grassland_maxRootDepth  = $(PathGrassland)/maxRootDepth.map
grassland_minSoilDepthFrac = $(PathGrassland)/minSoilDepthFrac.map


grassland_cropCoefficientNC = $(PathGrassland)/CropCoefficientGrassland_10days.nc
grassland_interceptCapNC    = $(PathGrassland)/interceptCapGrassland10days.nc

# initial conditions: grassland_interceptSto, grassland_w1, grassland_w2, grassland_w3


[__irrPaddy]
```

(continues on next page)

(continued from previous page)

```
PathIrrPaddy = $(FILE_PATHS:PathMaps)/landcover/irrPaddy

# Parameters for the Arno's scheme:
irrPaddy_arnoBeta = 0.2
# arnoBeta is defined by orographic,+ land cover add + calibration add, the soil water capacity distribution is based on this
# range [0.01 - 1.2]

# other paramater values

irrPaddy_minInterceptCap  = 0.001
irrPaddy_cropDeplFactor   = 0.0

irrPaddy_fracVegCover  = $(PathIrrPaddy)/fracVegCover.map
irrPaddy_rootFraction1 = $(PathIrrPaddy)/rootFraction1.map
irrPaddy_rootFraction2 = $(PathIrrPaddy)/rootFraction2.map
irrPaddy_maxRootDepth  = $(PathIrrPaddy)/maxRootDepth.map
irrPaddy_minSoilDepthFrac = $(PathIrrPaddy)/minSoilDepthFrac.map

irrPaddy_cropCoefficientNC = $(PathIrrPaddy)/CropCoefficientirrPaddy_10days.nc

# maximum flooding depth for paddy
irrPaddy_maxtopwater = 0.05



# initial conditions: irrPaddy_interceptStor, irrPaddy_w1, irrPaddy_w2, irrPaddy_w3



[__irrNonPaddy]
PathIrrNonPaddy = $(FILE_PATHS:PathMaps)/landcover/irrNonPaddy

# Parameters for the Arno's scheme:
irrNonPaddy_arnoBeta = 0.2
# arnoBeta is defined by orographic,+ land cover add + calibration add, the soil water capacity distribution is based on this
# range [0.01 - 1.2]

# other paramater values


irrNonPaddy_minInterceptCap  = 0.001
irrNonPaddy_cropDeplFactor   = 0.0

irrNonPaddy_fracVegCover  = $(PathIrrNonPaddy)/fracVegCover.map
irrNonPaddy_rootFraction1 = $(PathIrrNonPaddy)/rootFraction1.map
irrNonPaddy_rootFraction2 = $(PathIrrNonPaddy)/rootFraction2.map
irrNonPaddy_maxRootDepth  = $(PathIrrNonPaddy)/maxRootDepth.map
irrNonPaddy_minSoilDepthFrac = $(PathIrrNonPaddy)/minSoilDepthFrac.map


irrNonPaddy_cropCoefficientNC = $(PathIrrNonPaddy)/CropCoefficientirrNonPaddy_10days.nc


# initial conditions: irrNonPaddy_interceptStor, irrNonPaddy_w1, irrNonPaddy_w2, irrNonPaddy_w3
```

(continues on next page)

(continued from previous page)

```


[__sealed]
PathSealed = $(FILE_PATHS:PathMaps)/landcover/sealed

sealed_minInterceptCap  = 0.001

# initial conditions: sealed_interceptStor


[__open_water]
PathWater = $(FILE_PATHS:PathMaps)/landcover/water

water_minInterceptCap  = 0.0

#-------------------------------------------------------
[GROUNDWATER]
#-------------------------------------------------------

PathGroundwater = $(FILE_PATHS:PathMaps)/groundwater

recessionCoeff = $(PathGroundwater)/recessionCoeff.map
# baseflow = recessionCoeff * storage groundwater
specificYield = $(PathGroundwater)/specificYield.map
kSatAquifer = $(PathGroundwater)/kSatAquifer.map
# both not used at the moment in groundwater modul, but already loaded

#--------------------------------------------------------------
# INITIAL CONDITIONS: storGroundwater


#-------------------------------------------------------
[WATERDEMAND]
#-------------------------------------------------------

PathWaterdemand = $(FILE_PATHS:PathMaps)/landsurface/waterDemand
# For water demand vs. availability: areas have to be aggregated
# Allocation map
allocSegments = $(PathWaterdemand)/catchx.nc

domesticWaterDemandFile = $(PathWaterdemand)/domesticWaterDemand.nc
industryWaterDemandFile = $(PathWaterdemand)/industryWaterDemand.nc

irrNonPaddy_efficiency = $(FILE_PATHS:PathMaps)/landsurface/waterDemand/efficiency.nc
irrPaddy_efficiency = $(FILE_PATHS:PathMaps)/landsurface/waterDemand/efficiency.nc

#irrNonPaddy_efficiency = 0.8
#irrPaddy_efficiency = 0.8
irrigation_returnfraction = 0.5

# ---------------------------------------------------------
# Estimate of fractions of groundwater and surface water abstractions
# Either a fixed fraction for surface water abstration
# based on fraction of average baseflow and upstream average discharge
# if swAbstractionFrac < 0: fraction is taken from baseflow / discharge
# if swAbstractionFrac > 0 this value is taken as a fixed value
swAbstractionFrac = 0.9
```

(continues on next page)

(continued from previous page)

```
averageDischarge = $(FILE_PATHS:PathOut)/discharge_totalavg_rhine30min.nc
# in [m3/s]
averageBaseflow  = $(FILE_PATHS:PathOut)/baseflow_totalavg_rhine30min.nc
# in [m3/s]
baseflowInM = True
# if baseflow is in [m] instead of [m3/s] it will be converted


#-------------------------------------------------------
# RUNOFF CONCENTRATION
#-------------------------------------------------------
[RUNOFF_CONCENTRATION]

# using triagular weigthning method
# the bigger the factor, more lag time
forest_runoff_peaktime = 1.0
grassland_runoff_peaktime = 0.5
irrPaddy_runoff_peaktime = 0.5
irrNonPaddy_runoff_peaktime = 0.5
sealed_runoff_peaktime = 0.15
water_runoff_peaktime = 0.01

interflow_runoff_peaktime =1.0
baseflow_runoff_peaktime = 2.0

# initial conditions:
# here only 1 layer is shown, but there are up to 10: runoff_concIni


#-------------------------------------------------------
# ROUTING MAPS and PARAMETERSD
#-------------------------------------------------------
[ROUTING]

PathRouting = $(FILE_PATHS:PathMaps)/routing

# Number of substep per day
# should be 10 for 0.5 deg but 24 for 0.1 deg

NoRoutingSteps = 10
#kinematic wave parameter: 0.6 is for broad sheet flow
chanBeta = 0.6

# Channel gradient (fraction, dy/dx)
chanGrad = $(PathRouting)/kinematic/changrad.nc
# Minimum channel gradient (for kin. wave: slope cannot be 0)
chanGradMin = 0.0001

#Channel Manning's n
chanMan = $(PathRouting)/kinematic/chanman.nc
#Channel length [meters]
chanLength = $(PathRouting)/kinematic/chanleng.nc
#Channel bottom width [meters]
chanWidth = $(PathRouting)/kinematic/chanbw.nc
#Bankfull channel depth [meters]
chanDepth = $(PathRouting)/kinematic/chanbnkf.nc

```

(continues on next page)

(continued from previous page)

```
# initial conditions: channelStorageIni, riverbedExchangeIni, dischargeIni

#-------------------------------------------------------
# LAKES AND RESERVOIRS
#-------------------------------------------------------
[LAKES_RESERVOIRS]

PathLakesRes = $(FILE_PATHS:PathMaps)/routing/lakesreservoirs

# Use reservoirs and lakes (otherwise use only lakes Lake ID=1 and 3 => natural␣
↪conditions)
useResAndLakes = True
# Reservoirs do have a year of implementation
dynamicLakesRes = True
# if Reservoirs does not have a year of implemtation, which year should be taken as␣
↪fixed year
fixLakesResYear = 1950


#----------------------------
#Big lakes and Reservoirs

# ID of every lake, reservoir from HydroLakes database
waterBodyID = $(PathLakesRes)/lakesResID.nc
# 1 for lake, 2 for reservoir, 3 for lake and reservoir
waterBodyTyp = $(PathLakesRes)/lakesResType.nc
# Avergae discharge from HydroLakes Database
waterBodyDis = $(PathLakesRes)/lakesResDis.nc

# Lakes surface area from HydroLakes Database
waterBodyArea = $(PathLakesRes)/lakesResArea.nc
# a factor to scale the outlet of a lake
#lakeAFactor = 1.0  -> calibration

#----------------------------
# Small lakes and reservoirs

useSmallLakes = True

smallLakesRes = $(PathLakesRes)/smallLakesRes.nc
smallwaterBodyDis = $(PathLakesRes)/smalllakesresDis.nc

# averageRunoff in [m] (if not given smallwaterBodyDis is taken instead)
#averageRunoff  = $(FILE_PATHS:PathOut)/runoff_totalavg_cali.nc

# for water demand
#min storage in [m3] (if not give it is calculated)
#minStorage  = $(FILE_PATHS:PathOut)/minsmalllakeStorage_cali.nc


# initial conditions: lakeInflowIni, lakeStorageIni, outLakeIni, lakeOutflowIni,␣
↪reservoirStorageIni

#------------------
# Reservoirs
# reservoir volume from HydroLakes database
waterBodyVolRes = $(PathLakesRes)/lakesResVolRes.nc
```

(continues on next page)

(continued from previous page)

```
# reservoir starting year from HydroLakes database
waterBodyYear = $(PathLakesRes)/lakesResYear.nc

#  Conservative, normal and flood storage limit (fraction of total storage, [-])
conservativeStorageLimit = 0.1
#normalStorageLimit = 0.5   # --> put into calibration
floodStorageLimit = 0.9
# adjusting the balance between normal and flood storage
# [0 ..1]  0: NormalstorageLimit      1: (= closer to flood) results in keeping the␣
↪normal qoutflow longer constant
adjust_Normal_Flood = 0.5

# Minimum, Normal and Non-damaging reservoir outflow  (fraction of average discharge,␣
↪[-])
MinOutflowQ = 0.2
NormalOutflowQ = 1.0
NonDamagingOutflowQ = 4.0

# initial conditions: lakeInflowIni, lakeStorageIni, outLakeIni, lakeOutflowIni,␣
↪reservoirStorageIni


#-------------------------------------------------------
[INFLOW]
#-------------------------------------------------------

# if option inflow = true
# the inflow from outside is added at inflowpoints
In_Dir = $(FILE_PATHS:PathRoot)/in

# nominal map with locations of (measured)inflow hydrographs [cu m / s]
InflowPoints = $(In_Dir)/in.map
#InflowPoints = 8.25 49.75 7.75 50.25

# if InflowPoints is a map, this flag is to identify if it is global (False) or local␣
↪(True)
# observed or simulated input hydrographs as time series [cu m / s]
# Note: that identifiers in time series have to correspond to InflowPoints
# can be several timeseries in one file or different files e.g. main.tss mosel.tss
#QInTS = main1.tss mosel1.tss
QInTS = mm.tss



#-------------------------------------------------------
[ENVIRONMENTALFLOW]
#-------------------------------------------------------

# Either calculate without run with predone discharge (set calc_ef_after = False)
calc_ef_after = True
# Or calculate after run (set calc_ef_after = False) and defining the file to be used
EFDis = $(FILE_PATHS:PathOut)/discharge_rhine.nc

# if predone discharge, do the maps need to be cut to fit to the mask?
cut_ef_map = False

EnvironmentalFlowFile = $(FILE_PATHS:PathOut)/MQ90_12month.nc
```

(continues on next page)

(continued from previous page)

```

# MAF: Mean, Q90: percentile 90, MMF: monthly average, MQ90: monthly Q90 9averagwed over al Jan, Feb..
# EF_VMF: Environmental flow - variable monthly flow, EF_VMF_LIH - EF- variable monthly flow, high intermediate, low class
OUT_Dir = $(FILE_PATHS:PathOut)
#OUT_MAP_Once = MAF, Q90
#OUT_MAP_12month = MMF, MQ90, EF_VMF, EF_VMF_LIH
#OUT_MAP_12month = MQ90, EF_VMF



#++++++++++++++++++++++++++++++++++++++++++++++++++++++++
#++++++++++++++++++++++++++++++++++++++++++++++++++++++++


[OUTPUT]

# OUTPUT maps and timeseries
OUT_Dir = $(FILE_PATHS:PathOut)

OUT_TSS_Daily = discharge
#OUT_TSS_MonthAvg = discharge
#OUT_TSS_AnnualAvg = discharge

#OUT_Map_Daily = discharge
#OUT_Map_MonthAvg = discharge, precipitation, runoff
#OUT_Map_AnnualAvg = discharge
#OUT_MAP_TotalAvg = discharge, baseflow




```

4.6 NetCDF meta data

4.6.1 Output Meta NetCDF information

The metaNetcdf.xml includes information on the output netCDF files e.g. description of the parameter, unit ..

Example of a metaNetcdf.xml file:

```
<CWATM>
# METADATA for NETCDF OUTPUT DATA

# varname: name of the variable in the CWAT code
# unit: unit of the varibale
# long name# standard name

# Discharge maps
<metanetcdf varname="discharge" unit="m3/s"  standard_name="Discharge" long_name="Discharge in cubic meter per second"  title="1st Demo CWATM" author="PB" />

# others
```

(continues on next page)

(continued from previous page)

```
<metanetcdf varname="soilmoisture" unit="mm"  standard_name="soil moisture" long_name=
↪"Soil moisture"  title ="1st Demo CWATM" author="PB" />

# Initial condition Files
<metanetcdf varname="initcondition"  purpose ="Initial Conditions CWATM" author="PB" /
↪>
<metanetcdf varname="SnowCover1" unit="mm"  standard_name="SnowCover1" long_name=
↪"Snow cover top layer"  />
<metanetcdf varname="SnowCover2" unit="mm"  standard_name="SnowCover2" long_name=
↪"Snow cover middle layer"  />
<metanetcdf varname="SnowCover3" unit="mm"  standard_name="SnowCover3" long_name=
↪"Snow cover lower layer"  />
<metanetcdf varname="FrostIndex" unit="degree/days"  standard_name="FrostIndex" long_
↪name="Frost index based on Molnau, Bissel (1983)"  />
</CWATM>
```

4.6.2 Name and location of the NetCDF meta data file

In the settings file the name and location of the metadata file is given.

```
#-------------------------------------------------------
[NETCDF_ATTRIBUTES]
institution = IIASA
title = Global Water Model - WATCH WDFEI
metaNetcdfFile = $(FILE_PATHS:PathRoot)/cwatm/metaNetcdf.xml
```

4.7 Initialisation

CWatM needs to have estimates of the initial state of the internal storage variables, e.g. the amount of water stored in snow, soil, groundwater etc.
There are two possibilities:

1. The initial state of the internal storage variables are unknown and a **first** guess has to be used e.g. all storage variables are half filled.
2. The initial state is known from a previous run, where the variables are stored at a certain time step. This is called **warm start**

The **warm start** is usful for:

- using a long pre-run to find the steady-state storage of the groundwater storage and use it as initial value
- using the stored variables to shorten the warm-up period
- using the stored variables to restart every day with the values from the previous day (forecasting mode)

4.7.1 Example of soil moisture

The next figure shows the impact of different initial condition on the soil moisture of the lower soil. In one of the simulations the soil is initially almost completely saturated. In another simulation the soil is completely dry and the third simulation starts with initial conditions in between the two extremes.

In the beginning the effect of different initial condition can be seen clearly. But after one year the three curves converge. The **memory** of the lower soil goes back for about one year.

For all the initial condition apart from groundwater, lakes and reservoirs the memory is about 12 month.

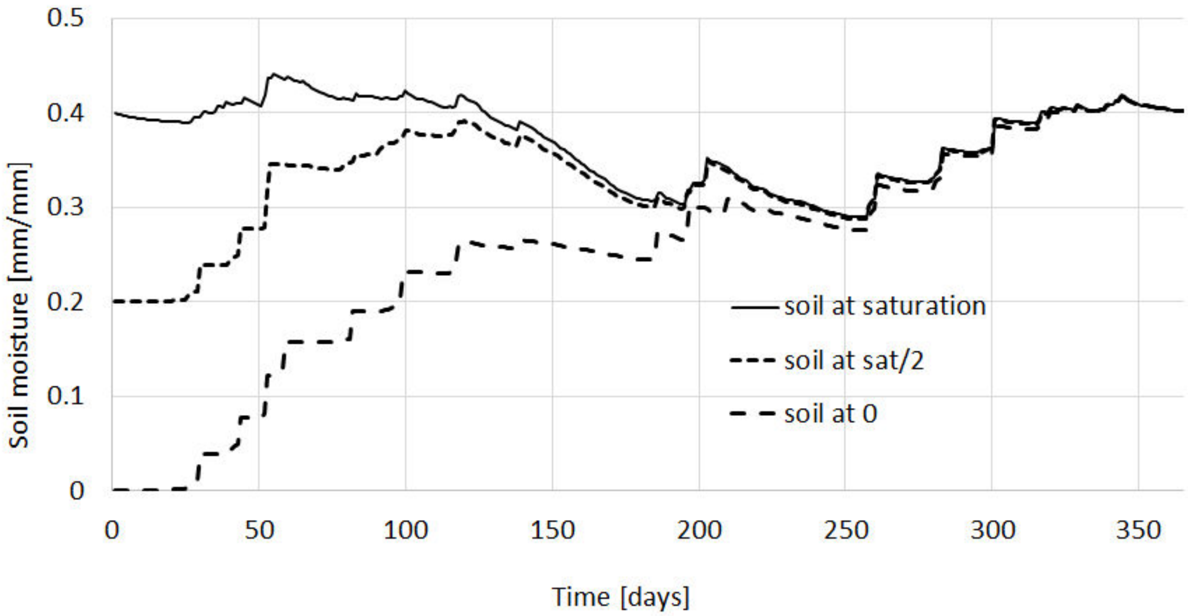


Figure: Simulation of soil moisture in the lower soil with different initial conditions

For the groundwater zone a longer warm-up period is needed, because of the slow response of groundwater. Here a rather fast reacting groundwater storage is shown with the three curves coverge after two years. We propose a warm-up of several decades. The longer the better.

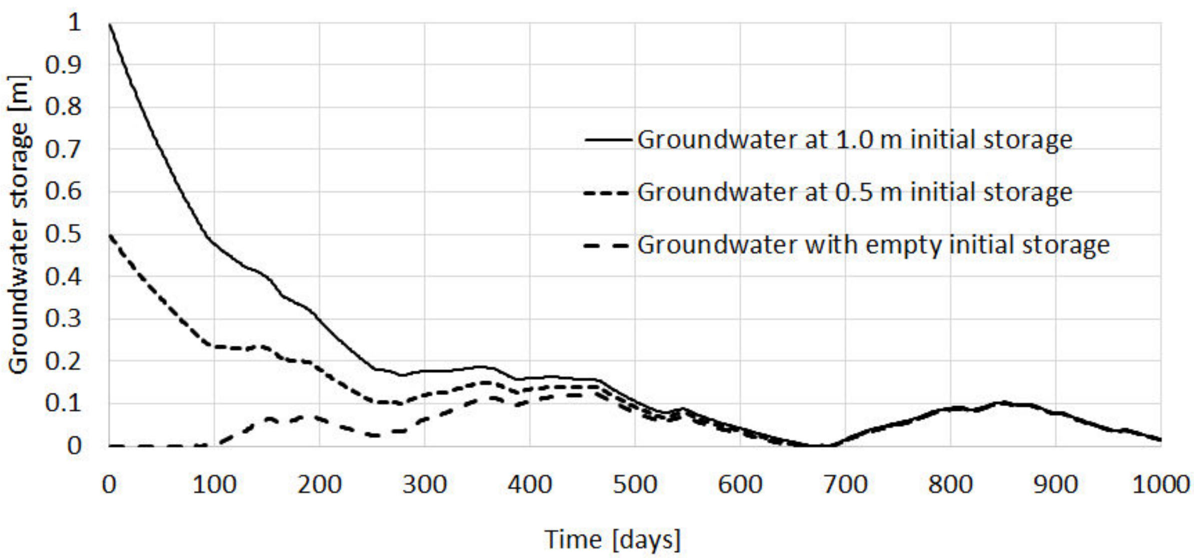


Figure: Simulation of groundwater storage with different initial conditions

4.7.2 Cold start

For a **cold start** the values of the storage variables are unknown and set to a "first" guess. A list of variables and their default value for a **cold start** is given below in: *Initial conditions* (page 47)

Set up a cold start in the settingsfile

In the settings file the option: **load_initial** has to be set on **False**

```
#-------------------------------------------------------
[INITITIAL CONDITIONS]
#-------------------------------------------------------

# for a warm start initial variables a loaded
# e.g for a start on 01/01/2010 load variable from 31/12/2009
load_initial = False
initLoad = $(FILE_PATHS:PathRoot)/init/Rhine_19891231.nc
```

Note: It is possible to exclude the warming up period of your model run for further analysis of results by setting the **SpinUp** option

```
[TIME-RELATED_CONSTANTS]
SpinUp =  01/01/1995
```

4.7.3 Storing initial variables

In the settings file the option **save_initial** has to be set to **True**

The name of the initial netCDF4 file has to be put in **initsave**

and one or more dates have to be specified in StepInit

```
# saving variables from this run, to initiate a warm start next run
# StepInit = saving date, can be more than one: 10/01/1973 20/01/1973
save_initial = False
initSave = $(FILE_PATHS:PathRoot)/init/Rhine
StepInit = 31/12/1989 31/12/2010
```

4.7.4 Warm start

CWatM can write internal variables to a netCDF file for choosen timesteps. These netCDF files can be used as the initial conditions for a suceeding simulation.

This is useful for establishing a steady-state after a long-term run and then using this steady-state for succeding simu-lations or for an every day run (forecasting mode)

Warning: If the parameters are changed after a run (especially the groundwater, lakes and reservoir parameters) the stored initial values do not represent the conditions of the storage variables. Stored initial conditions should **not** be used as initial values for a model run with another set of parameters. If you do this during calibration, you will not be able to reproduce the calibration results!

Set up a cold start in the settingsfile

In the settings file the option: **load_initial** has to be set on **True** And define the name of the netcdf4 file in **initLoad**

Note: Use the initial values of the previous day here. E.g. if you run the model from 01/01/2006 use the inital condition from 31/12/2005

```
#-------------------------------------------------------
[INITITIAL CONDITIONS]
#-------------------------------------------------------

# for a warm start initial variables a loaded
# e.g for a start on 01/01/2010 load variable from 31/12/2009
load_initial = False
initLoad = $(FILE_PATHS:PathRoot)/init/Rhine_19891231.nc
```

4.7.5 Initial conditions

No.	Variable	Description	Default value	Number of maps
1	SnowCover	Snow cover for up to 7 zones	0	7
2	FrostIndex	Degree days frost threshold	0	1
3	Forest state	Interception storage	0	1
		Top water layer	0	1
		Soil storage for 3 soil layers	0	3
4	Grassland state	Interception storage	0	1
		Top water layer	0	1
		Soil storage for 3 soil layers	0	3
5	Paddy irrigation state	Interception storage	0	1
		Top water layer	0	1
		Soil storage for 3 soil layers	0	3
6	Irrigation state	Interception storage	0	1
		Top water layer	0	1
		Soil storage for 3 soil layers	0	3
7	Sealed area state	Interception storage	0	1
8	Groundwater	Groundwater storage	0	1
9	Runoff concentration	10 layers of runoff concentration	0	10
10	Routing	Channel storage	0.2 * total cross section	1
	Routing	Riverbed exchange	0	1
	Routing	Discharge	depending on ini channel stor.	1
11	Lakes and Reservoirs	Lake inflow	from HydroLakes database	1
		Lake outflow	same as lake inflow	1
		Lake&Res outflow to other lakes&res	same as lake inflow	1
		Lake storage	based on inflow and lake area	1
		Reservoir storage	0.5 * max. reservoir storage	1
		Small lake storage	based on inflow and lake area	1
		Small lake inflow	from HydroLakes database	1
		Small lake outflow	same as small lake inflow	1

4.8 Model Output

An advantage of CWatM is the full flexibility of the output variables.

- All parameters and variables can be used for output as maps or time series.
- Even if the model is run at daily timestep, output can be daily, monthly, annual, at the end of a run
- all variables maps are stored as netcdf and the meta data information can be added

4.8.1 Time depending and non depending output maps

Output maps will be produced as spatial maps, stack of spatial maps (over time)
Format: netCDF[25]

The netCDF maps can be read with:

Windows

- Panoply[26]

Linux

- ncview[27]
- cdo[28]

4.8.2 Or time series at specified points

Timeseries are procuded as ASCII files, which can be read with every text editor
or with PCRaster Aquila[29]

The specific point(s) where timeseries are provided are defined in the settings file as *Gauges*:
Can be several points in the format lon lat lon lat ..

```
# Station data
# either a map e.g. $(FILE_PATHS:PathRoot)/data/areamaps/area3.map
# or a location coordinates (X,Y) e.g. 5.75 52.25 9.25 49.75 )
# Lobith/Rhine
Gauges = 6.25 51.75 7.75 49.75

# if .tif file for gauges, this is a flag if the file is global or local
# e.g. Gauges = $(FILE_PATHS:PathRoot)/data/areamaps/gaugesRhine.tif
GaugesLocal = True
```

4.8.3 Output variables

Output can be every global defined variable in the model Variable are e.g. Precipitation, runoff, baseflow

but also not so common variables as:

- reservoirStorage (amount of water in the reservoirs in [m3])
- nonIrrReturnFlowFraction (returnflow from domenstic and industrial water use [m3])
- actualET[1] (actual evapotranspiration from grassland [m/day])
- ...

[25] http://www.unidata.ucar.edu/software/netcdf/
[26] http://www.giss.nasa.gov/tools/panoply
[27] http://meteora.ucsd.edu/~pierce/ncview_home_page.html
[28] https://www.unidata.ucar.edu/software/netcdf/workshops/2012/third_party/CDO.html
[29] http://pcraster.geo.uu.nl/projects/developments/aguila/

4.8.4 Daily, monthly - at the end or average

- per day
- total month, average month, end of month
- total year, average year, end of year
- total average, total at the end

available prefixes are: 'daily', 'monthtot','monthavg', 'monthend','annualtot','annualavg','annualend','totaltot','totalavg'

for example

```
[OUTPUT]
# OUTPUT maps and timeseries
OUT_Dir = $(FILE_PATHS:PathOut)
OUT_MAP_Daily = discharge, runoff
OUT_MAP_MonthAvg = Precipitation
OUT_MAP_TotalEnd = lakeStorage
OUT_MAP_TotalAvg = Tavg

OUT_TSS_Daily = discharge
OUT_TSS_MonthTot = runoff
OUT_TSS_AnnualAvg = Precipitation
OUT_TSS_AnnualTot = runoff
```

Note: For each variable the meta data information can be defined in *Output Meta NetCDF information* (page 42)

Note: For information how to adjust the output in the settings file see *Output* (page 25)

4.8.5 Time series as point infomation or catchment sum or average

As standard time series can include values of the specific cell as defined in the settings file as *Gauges* But time series can also show the area sum or area average of the upstream catchment from the specific cell

for example

```
[OUTPUT]
# OUTPUT maps and timeseries
# Standard values of a specific cell
OUT_TSS_Daily = discharge
OUT_TSS_AnnualAvg = Precipitation
# Area sum of upstream catchment
OUT_TSS_AreaSum_MonthTot = Precipitation, runoff
# Area sum of upstream catchment
OUT_TSS_AreaAvg_MonthTot = Precipitation
```

4.8.6 Most important output variables - a selection

```
#Variable name     : Description
discharge          : river discharge
```

(continues on next page)

(continued from previous page)

```
runoff              : runoff
Precipitation       : rainfall + snow
Tavg                : average temperature
ETRef: potential    : evaporation from reference soil
sum_gwRecharge      : total groundwater recharge
totalET             : total actual evapotranspiration
baseflow            : baseflow from groundwater
... (to be continued)
```

4.8.7 Output variables - starting a list

A list of variables can be produced by using:
grep -d recurse 'self.var.' *.py
Every self.var.variable can be used as output variable
For a description of the variable please take a look at the python module itself.
.
As output variable please use without self.var.

We started a list of possible output variables. Please note that this list is under construction. We still need to fill in all descriptions and all units. You find this list at *List of output variables* (page 80)

CHAPTER

FIVE

TUTORIAL

Contents

5.1 Requirements

5.1.1 Requirements

Python version

NEW from 2019 on: Requirements are a 64 bit Python 3.7.x version[30]

Warning: a 32 bit version is not able to handle the data requirements!

Warning: From 2019 on we are changing to Python37. We do not provide further support for Python 2.7

Warning: CWatM is tested for Python 3.7, 3.8 and will for sure not work with Python versions lower than 3.6. We recommend using Python 3.7, 3.8

Libraries

These external libraries are needed:

- Numpy[31]
- Scipy[32]
- netCDF4[33]
- GDAL[34]
- Flopy[35]

Warning: Installing GDAL via pip causes sometimes problems. We recommend downloading the library from Unofficial Windows Binaries for Python Extension Packages[36] as GDAL-3.0.4-cp37-cp37m-win_amd64.whl (or a later version depending on your Python version) and installing them as:

pip install C:/Users/XXXXX/Downloads/GDAL-3.0.4-cp37-cp37m-win_amd64.whl

Windows

The five libraries can be installed with pip or downloaded at Unofficial Windows Binaries for Python Extension Packages[37]

[30] https://www.python.org/downloads/release/python-372/
[31] http://www.numpy.org
[32] https://www.scipy.org
[33] https://pypi.python.org/pypi/netCDF4
[34] http://www.gdal.org
[35] https://www.usgs.gov/software/flopy-python-package-creating-running-and-post-processing-modflow-based-models
[36] http://www.lfd.uci.edu/~gohlke/pythonlibs
[37] http://www.lfd.uci.edu/~gohlke/pythonlibs

Windows executeable Python version

The A cwatmexe.zip with all Python libraries and a test case (River Rhine)) is stored on:
Source code on Github repository of CWATM[38]
Executable cwatmexe.zip on Github repository of CWATM[39]

5.2 Test the executable model version

only Windows

If you familiar with Python just go to the next chapter.

```
cwatm
|-- README.md
|
|
└-- cwatmexe
|    └- lib
|    └---cwatm.exe
|    └---metaNetcdf.xml
|    └--- libraries etc.
|
|
└--rhine_basin
|    └---climate_rhine
|    └---cwatm_input_rhine
|    └---init
|    └---output
|    └---run_python_rhine30.bat
|    └---settings_rhine30.ini
|
|
└-- run_test1.bat
└-- run_test2_rhine30min.bat
└-- settings_rhine_test.ini
└-- tutorial.html
```

Either start cwatm.exe in a DOS box (cmd windows command), or use a batch file e.g. run_test1.bat

5.2.1 Test 1

In the root directory cwatm

Please try:

```
run run_test1.bat or type .\cwatmexe\cwatm.exe
```

The output should be like See: *Test the Python model version* (page 54)

5.2.2 Test 2

Please try:

[38] https://github.com/CWatM/CWatM
[39] https://github.com/CWatM/CWatM/blob/version091/cwatmexe.zip

```
run run_test2_rhine30min.bat or type .\cwatmexe\cwatm.exe settings_rhine30_test.ini -l
```

The output should be like See: *Error and exeption handling* (page 57)

5.3 Test the Python model version

Windows and Linux (and maybe Mac, but not tested)

Please try:

```
python <modelpath>/run_cwatm.py  (for the Python3.7 version)
or:
<modelpath>/cwatm  (for the .exe version)
```

The output should be:

```
Running under platform:  Windows  **(or Linux etc)**
CWatM - Community Water Model
Authors: ...
Version: ...
Date: ...
Arguments list:
settings.ini     settings file
-q --quiet       output progression given as .
-v --veryquiet   no output progression is given
-l --loud        output progression given as time st
-c --check       input maps and stack maps are check
-h --noheader    .tss file have no header and start
-t --printtime   the computation time for hydrologic
-w --warranty    copyright and warranty information
```

Warning: If python is not set in the environment path, the full path of python has to be used

5.3.1 Error because you did not run it with Python

if the model is causing an error with look like this:

```
File "run_cwatm.py", line 116
print("%-6s %10s %11s\n" %("Step","Date","Discharge"), end=' ')
SyntaxError: invalid syntax
```

You run the model without the python command in front. Please use: python cwatm.py (You may have to adjust the path to your python version and to cwatm.py).

5.3.2 Error because python is not added to the PATH

If executing python return an error like this

> 'python' is not recognized as an internal or external command

You need either need to add Python to the PATH Environmental Variable or you need to start Python with full path.

> c:/path_to_python/python

5.3.3 Error because the python libraries are installed incorrectly

If the model is causing an error at this stage, please check the python libraries:

```
python
import numpy
import scipy.ndimage
import gdal
import netCDF4
```

5.4 Running the model 1

Warning: The model needs a settings file as an argument. See: *Settings file* (page 21)

python <modelpath>/cwatm.py settingsfile flags

example:

```
python cwatm.py settings_rhine.ini -l
```

The flag -l show the output on screen as date and discharge

At this point you should receive this eror message:

```
======================== CWATM FILE ERROR ===========================
Cannot find option file: d:/work/CWATM/source/metaNetcdf.xml In  "metaNetcdfFile"
searching: "d:/work/CWATM/source/metaNetcdf.xml"
path: d:/work/CWATM/source does not exists
```

5.5 Downloading and installing the spatial dataset

The spatial dataset contains:

- static data ie. data that does not change over time (a model assumption) e.g. soil data
- time dependend (inter annual) data that change periodical during a year e.g. crop coefficient of vegetation
- time dependend (intra annual) data that change by month or year e.g. fraction of landcover

These data are stored as global dataset:

- cwat_input.zip for the 30' global version
- cwat_input5min.zip for the 5' global version

As climate data different forcings can be used e.g:

- PGMFD v.2 (Princeton), GSWP3, etc.
- precipitation from e.g. MSWEP http://www.gloh2o.org/
- WATCH+WFDEI https://www.isimip.org/gettingstarted/details/5/

and as projection e.g.:

- ISI-MIP dataset https://www.isimip.org/gettingstarted/#input-data-bias-correction

For the tutorial we cut out Rhine basin and included the WATCH+WFDEI precipitation, average temperature and the calculated potential evaporation .
A 30' and a 5' version can be found on FTP in rhine/climate

Reference:

Weedon, G.P., S.S. Gomes, P.P. Viterbo, W.J. Shuttleworth, E.E. Blyth, H.H. Österle, J.C. Adam, N.N. Bellouin, O.O. Boucher, and M.M. Best, 2011: Creation of the WATCH Forcing Data and Its Use to Assess Global and Regional Reference Crop Evaporation over Land during the Twentieth Century. J. Hydrometeor., 12, 823–848, doi: 10.1175/2011JHM1369.1

Weedon, G. P., G. Balsamo, N. Bellouin, S. Gomes, M. J. Best, and P. Viterbo (2014), The WFDEI meteorological forcing data set: WATCH Forcing Data methodology applied to ERA-Interim reanalysis data, Water Resour. Res., 50, 7505–7514, doi:10.1002/2014WR015638.

Note:

Please copy and unpack the spatial dataset (either 30' or 5')in a folder
Please copy the the climate dataset 30min_meteo_rhine.zip or 5min_meteo_rhine.zip in a seperate folder
Please create a folder called output

Note:

For testing purpose there is a file rhine_basin.zip on GitHub
it has all the necessary data to run the River Rhine on 30 arcmin from 1990-2010

5.6 Changing the Settings file

to run the model the pathes to data have to be set correctly: The information of pathes are stored in the settings file around line 80-100

[FILE_PATHS]:

```
PathRoot = E:/
PathOut = $(PathRoot)/output
PathMaps = E:/cwatm_input
PathMeteo = E:/climate
#-------------------------------------
[NETCDF_ATTRIBUTES]
institution = IIASA
title = Global Water Model - WATCH WDFEI
metaNetcdfFile = $(FILE_PATHS:PathRoot)/CWATM/source/metaNetcdf.xml
```

Note: Please change the pathes according to your file system

5.7 Error and exception handling

We try to make our program behave properly when encountering unexpected conditions. Therefore we caption a number of possible wrong inputs.

If you get an output with an error number please look at *Error handling* (page 60)

5.8 Running the model 2

If you type now:

```
python cwatm.py settings_rhine.ini -l
```

You should see:

```
E:\CWATM_rhine\source>python cwatm.py settings_rhine30min.ini -l
CWATM - Community Water Model  Version: 0.991  Date:  16/09/2017
International Institute of Applied Systems Analysis (IIASA)
Running under platform:  Windows
-----------------------------------------------------------
CWATM Simulation Information and Setting
The simulation output as specified in the settings file: settings_rhine30min.ini
can be found in E:/CWATM_rhine/output
Step         Date   Discharge
1      01/01/1961        4.20
2      02/01/1961        4.23
...
```

If you don't see this. Something went wrong and you might see this instead:

```
E:\CWATM_rhine\source>python cwatm.py settings_rhine30min.ini -l
CWATM - Community Water Model  Version: 0.991  Date:  16/09/2017
International Institute of Applied Systems Analysis (IIASA)
Running under platform:  Windows
-----------------------------------------------------------
ERROR 4: `E:/CWATM_rhine/cwatm_input/routing/ldd.map' does not exist in the file␣
↪system,
and is not recognised as a supported dataset name.
management_modules.messages.CWATMFileError:
======================== CWATM FILE ERROR ===========================
In  "Ldd"
searching: "E:/CWATM_rhine/cwatm_input/routing/ldd.map"
path: E:/CWATM_rhine/cwatm_input/routing does not exists
```

The model tries to help you on finding the error.
In this case it is looking for the river network map ldd.map or ldd.nc or ldd.tif
but it cannot find the file and not even the path to the file.

Here you might change:

```
[FILE_PATHS]
PathRoot = E:/CWATM_rhine
PathMaps = $(PathRoot)/cwatm_input
```

or:

```
[TOPOP]
# local drain direction map (1-9)
Ldd = $(FILE_PATHS:PathMaps)/routing/ldd.map
```

But many other error can occure too! Have fun.

P.s. some error we captured and we give a hint. Please look at *Error handling* (page 60)

5.9 Changing parameters of the model

Note: An overview of possibilities is given in see *Settings file* (page 21)

5.10 Changing the Output

5.10.1 Output variables

Output can be every global defined variable in the model Variable are e.g. Precipitation, runoff, baseflow

but also not so common variables as:

- reservoirStorage (amount of water in the reservoirs in [m3])
- nonIrrReturnFlowFraction (returnflow from domenstic and industrial water use [m3])
- actualET[1] (actual evapotranspiration from grassland [m/day])
- ...

5.10.2 Daily, monthly - at the end or average

- per day
- total month, average month, end of month
- total year, average year, end of year
- total average, total at the end

available prefixes are: 'daily', 'monthtot','monthavg', 'monthend','annualtot','annualavg','annualend','totaltot','totalavg'

for example

```
[OUTPUT]
# OUTPUT maps and timeseries
OUT_Dir = $(FILE_PATHS:PathOut)
OUT_MAP_Daily = discharge, runoff
OUT_MAP_MonthAvg = Precipitation
```

(continues on next page)

(continued from previous page)

```
OUT_MAP_TotalEnd = lakeStorage
OUT_MAP_TotalAvg = Tavg

OUT_TSS_Daily = discharge
OUT_TSS_AnnualAvg = Precipitation
```

Note: For each variable the meta data information can be defined in *Output Meta NetCDF information* (page 42)

Note: For information how to adjust the output in the settings file see *Output* (page 25)

5.10.3 Time series as point infomation or catchment sum or average

As standard time series can include values of the specific cell as defined in the settings file as *Gauges* But time series can also show the area sum or area average of the upstream catchment from the specific cell

for example

```
[OUTPUT]
# OUTPUT maps and timeseries
# Standard values of a specific cell
OUT_TSS_Daily = discharge
OUT_TSS_AnnualAvg = Precipitation
# Area sum of upstream catchment
OUT_TSS_AreaSum_MonthTot = Precipitation, runoff
# Area sum of upstream catchment
OUT_TSS_AreaAvg_MonthTot = Precipitation
```

5.10.4 Most important output variables - a selection

```
#Variable name     : Description
discharge          : river discharge
runoff             : runoff
Precipitation      : rainfall + snow
Tavg               : average temperature
ETRef: potential   : evaporation from reference soil
sum_gwRecharge     : total groundwater recharge
totalET            : total actual evapotranspiration
baseflow           : baseflow from groundwater
... (to be continued)
```

5.10.5 A list of all variables

We started a list of possible output variables. Please note that this list is under construction. We still need to fill in all descriptions and all units. You find this list at *List of output variables* (page 80)

CHAPTER

SIX

ERROR HANDLING

We try to make our program behave properly when encountering unexpected conditions. The problematic situations that a program can encounter fall into two categories.

- Programmer mistakes: If someone forgets to pass a required argument to a function
- Genuine problems: If the program asks the user to enter a name and it gets back an empty string, that is something the programmer can not prevent.

This part deals with genuine problems. Please look for your error number

Error No	Description	Settingsfile	Measure	Module	Procedure
E101 (page 67)	Gauges in settingsfile is not a coordinate e.g. Gauges = bad bad	Gauges	Put in pairs of coordinates or a map with coordinates	datahandling	valuecell
E102 (page 67)	One of the gauges is outside the map extend of the mask map	Gauges	Make sure that all gauges are inside the mask map area	datahandling	valuecell
E103 (page 68)	Maskmap is not a valid mask map nor valid coordinates nor valid point e.g. MaskMap = 1 2 3 4 5 6	MaskMap	Put in a pair of coordinates, a defined rectangle (5 numbers) or a filename	datahandling	loadsetclone
E104 (page 68)	MaskMap point does not have a valid value in the river network (LDD)	MaskMap	You put in a coordinate as MaskMap, but at this coordinate there is no valid river network value	datahandling	maskfrompoint
E105 (page 68)	The map you are loading has a different shape (different cols,rows) than the other maps	Any map	Make sure your map has the same resolution, rows, cols than the river network map	datahandling	compressArray

Continued on next page

Table 1 – continued from previous page

Error No	Description	Settingsfile	Measure	Module	Procedure
E106 (page 68)	The map you are loading has missing cell values (NaNs) where the river network and the mask map has valid values	Any map	Check the map and include cell values where the river network map has valid (non NaNs) values	datahandling	compressArray
E107 (page 68)	The netCDF map has different attributes (resolution, rows,cols) than your standard map	Any netCDF map	Make sure your map has the same resolution, rows, cols as the river network map	datahandling	mapattrNetCDF
E108 (page 69)	The tif map has different attributes (resolution, rows,cols) than your standard map	Any tif map	Make sure your map has the same resolution, rows, cols as the river network map	datahandling	mapattrTiff
E109 (page 69)	The meteo map (e.g. temperature, precipitation, evaporation..) has different attributes (resolution, rows,cols) than your standard map	Meteo netCDF map	Make sure your meteo input maps have the same resolution, rows, cols as the river network map. If it is the ET maps, it might be from another run with different mask. Please look at the option: calc_evaporation	datahandling	readmeteodata
E110 (page 69)	The netcdf map with a time variable (e.g. waterdemand, land cover, lakes..) has different attributes (resolution, rows,cols) than your standard map	All time depending netCDF maps appart from meteo maps	Make sure your netCDF input maps have the same resolution, rows, cols as the river network map	datahandling	readnetcdf2

Continued on next page

Table 1 – continued from previous page

Error No	Description	Settingsfile	Measure	Module	Procedure
E111 (page 69)	The netcdf maps without time is turned upside dowm	Any netCDF map without time	Make sure your map is in right order e.g. latitude coordinate are turned	datahandling	readnetcdfWithoutTime
E112 (page 70)	The initial map (load_initial) stack is turned upside down	Initload	Make sure your initial map (in initial_load) is in right order e.g. latitude coordinate are turned	datahandling	readnetcdfInitial
E113 (page 70)	The initial map stack (initial_load) has different attributes (resolution, rows,cols) than your standard map	Initload	Make sure your initial maps (in initial_load)has the same attributes. Maybe your initial maps are from a different run with a different mask?	datahandling	readnetcdfInitial
E114 (page 70)	Problem reading initial maps	Initload	Initial maps has maybe not the same shape as the mask map. Maybe put load_initial = False	datahandling	readnetcdfInitial
E115 (page 70)	Variable with a True or False value can not be read	Any flag (True or False)	Make sure the variable has either: varname = False or varname = True. Ttrue or faalse is not working!	datahandling	returnBool
E116 (page 70)	One of the variable names in [Option] is written wrong	Any variable in [Option]	A keyword in option is maybe written wrong e.g. CaaapillarRise instead CapillarRise	datahandling	checkOption
E117 (page 71)	One of the variable names is written wrong	Any variable after [Option]	A keyword in option is maybe written wrong e.g. MaaaskMap instead MaskMap. Pay attention: in Linux words are case sensitive!	datahandling	cbinding

Continued on next page

Table 1 – continued from previous page

Error No	Description	Settingsfile	Measure	Module	Procedure
E118 (page 71)	Timing in the section TIME-RELATED_CONSTANTS is wrong	StepStart, SpinUp, StepEnd	Please check the variables StepStart, SpinUp, StartEnd	timestep	ctbinding
E119 (page 71)	'Either date in StepStart is not a date or in SpinUp or StepEnd it is neither a number or a date	StepStart, SpinUp, StepEnd	Please check the variables StepStart, SpinUp, StartEnd - A date is missing	timestep	Calendar
E120 (page 71)	First date in StepInit is neither a number or a date	StepInit	Check StepInit in INITITIAL CONDITIONS. It is not a date or a number	timestep	Calendar
E121 (page 71)	Second value in StepInit is not a number or date nor indicating a repetition of year(y), month(m) or day(d) e.g. 2y for every 2 years or 6m for every 6 month	StepInit	Check StepInit. The second part is neither a date or d,m,y	timestep	datetosaveInit
E122 (page 72)	Third value in StepInit is not an integer after 'y' or 'm' or 'd'	StepInit	Check StepInit. The third part after d,m,y is not an integer	timestep	datetosaveInit
E123 (page 72)	StepStart has to be a valid date	StepStart	Check StepStart. It has to be a date e.g. 01/01/2009	timestep	checkifDate
E124 (page 72)	StepEnd is earlier than StepStart	StepStart, StepEnd	Check StepStart and StepEnd. StepEnd has to be later than StepStart	timestep	checkifDate
E125 (page 72)	Spin Date is smaller/bigger than the first/last time step date	StepStart, SpinUp, StepEnd	Check that SpinDate is in between StepStart and StepEnd	timestep	checkifDate
E127 (page 72)	Coordinates are not pairs	InflowPoints	Check that location in InflowPoints comes in pairs	inflow	initial

Continued on next page

Table 1 – continued from previous page

Error No	Description	Settingsfile	Measure	Module	Procedure
E128 (page 73)	OUT_MAP_Daily = discharge may be not defined in [OUTPUT]	calc_ef_afterRun, Out_MAP_Daily	Make sure that you define Out_MAP_Daily = discharge. Is seems daily discharge is not stored	'Environflow	initial
E129 (page 73)	Output points in Gauges are not pairs e.g. Gauges = 17.5 55.6 18.5	Gauges	Check that output-points in Gauges are of coordinate e.g. Gauges = 6.25 51.75	output	initial
E130 (page 73)	Out_TSS or Out_MAP is not one of these: daily, monthend, monthtot, monthavg, annualend, annualtot, annualavg	Out_TSS, Out_MAP	Please check that the wording after Out_TSS or Out_MAP is correct. Only a few keywords are valid e.g. Out_MAP_Daily, Out_TSS_monthavg	output	initial
E131 (page 73)	Second keyword after TSS or MAP is wrong	Out_TSS, Out_MAP	Use TSS for point value, AreaSum for sum of area, AreaAvg for average of area e.g. OUT_TSS_AreaSum_Daily	output	initial
E132 (page 73)	Variable is not defined in list of variables	Out_TSS, Out_MAP	Please correct the writing of the variable name, check also case sensitive e.g. Precipitation instead precipitation	output	dynamic
E201 (page 74)	Cannot load the maskmap as a file e.g. MaskMap = notexisting.tif	MaskMap	Make sure the file you put in MaskMap is existing and is a map e.g. *.nc, *.map, *.tif	datahandling	loadsetclone
E202 (page 74)	Your map is upside down	Any map	Make sure your map is in right order e.g. latitude coordinate are turned	datahandling	loadmap

Continued on next page

Table 1 – continued from previous page

Error No	Description	Settingsfile	Measure	Module	Procedure
E203 (page 74)	Cannot find the map, filename does not exists at this location	Any map	the datafile cannot be found, the location is wrong or the file is missing	datahandling	loadmap
E204 (page 74)	Trying to read the precipitation map metadata	Precipitation map	Make sure the precipitation netcdf are existing and correct	datahandling	metaNetCDF
E205 (page 74)	Read error while reading netcdf map	any netcdf map	Make sure netcdf map is existing and correct	datahandling	readCoordNetCDF
E206 (page 75)	read error while reading meteo maps	any meteo maps	Make sure the meteo input maps are in the right location and chave correct attributes	datahandling	checkMeteo_Wordclim
E207 (page 75)	read error while reading the wordclim map for downscaling meteo maps	Wordlclim maps	Make sure the wordclim maps are in the right location and chave correct attributes	datahandling	checkMeteo_Wordclim
E208 (page 75)	Cannot find meteomaps	any meteomap	Make sure the meteo input maps are at the right location	datahandling	multinetdf
E209 (page 75)	'Error loading meteomaps	any meteomap	Check if the filenames of the meteomaps are ok. Does the filename exist at this location	datahandling	multinetdf
E210 (page 75)	Meteomap does not have this date	any meteomap	The netCDF file does not contain this date. Check your start and end date	datahandling	readmeteodata
E211 (page 76)	'Error loading meteomaps	any meteomap	Check if the filenames of the meteomaps are ok. Does the filename exist at this location	datahandling	readmeteodata

Continued on next page

Table 1 – continued from previous page

Error No	Description	Settingsfile	Measure	Module	Procedure
E212 (page 76)	'Error loading any other time depending map but not meteomaps	any time depending netCDF map	Check if the filenames of the map is ok. Does the filename exist at this location	datahandling	readnetcdf2
E213 (page 76)	'Error loading netCDF maps without time	any non time depending netCDF map	Check if the filenames of the map is ok. Does the filename exist at this location	datahandling	readnetcdfWithoutTime
E214 (page 76)	'Error loading the initial map	initial netCDF map	Check if the loaction of the initial map is o. Please check filename in initLoad	datahandling	readnetcdfInitial
E215 (page 76)	Trying to read the precipitation map metadata	Precipitation map	Make sure the precipitation netcdf are existing and correct	timestep	checkifDate
E216 (page 77)	Outflow/Inflow point file cannot be loaded	InflowPoints	Check if filename of InflowPoints exists	inflow	initial
E217 (page 77)	Mistake reading inflow file, name of inflow points are used twice	In_Dir	Check inflow file header	inflow	initial
E218 (page 77)	Mistake reading inflow file	In_Dir	Check if filename of inflow file exists	inflow	initial
E219 (page 77)	Mistake in discharge daily netcdf file	Out_MAP_Daily = discharge	Something is wrong with the daily discharge file. Check location and content	'Environment	initial
E220 (page 77)	Output file path is wrong	OUT_Dir, PathOut	Check OUT_Dir and PathOut. OUT_Dir can be used several times in the settings file.	output	appendinfo

Continued on next page

Table 1 – continued from previous page

Error No	Description	Settingsfile	Measure	Module	Procedure
E221 (page 78)	'Error using file in Gauges	Gauges	Please check if file in Gauges exists and is correct	output	initial
E301 (page 78)	Python version is not 64 bit	•	Make sure that you use a 64 bit version of Python	global	main
E302 (page 78)	Settingsfile not found	•	Make sure the settings file exists and is at the right location	configuration	parse_configuration
E303 (page 78)	An error occured while reading metaNetcdf.xml	metaNetcdfFile	Please check that metaNetcdf is a valid .xml file	configuration	read_metanetcdf
E304 (page 78)	Cannot find alternative option file metaNetcdf.xml not found	metaNetcdfFile	Please check in metaNetcdfFile that the file exists	configuration	read_metanetcdf
E305 (page 79)	An error occured while reading alternative metaNetcdf.xml	metaNetcdfFile	Please check that metaNetcdf is a valid .xml file	configuration	read_metanetcdf

6.1 E101

Description Gauges in settingsfile is not a coordinate e.g. Gauges = bad bad

Where Gauges

What to do Put in pairs of coordinates or a map with coordinates

Module datahandling, valuecell

6.2 E102

Description One of the gauges is outside the map extend of the mask map

Where Gauges

What to do Make sure that all gauges are inside the mask map area

Module datahandling, valuecell

6.3 E103

Description Maskmap is not a valid mask map nor valid coordinates nor valid point e.g. MaskMap = 1 2 3 4 5 6

Where MaskMap

What to do Put in a pair of coordinates, a defined rectangle (5 numbers) or a filename

Module datahandling, loadsetclone

6.4 E104

Description MaskMap point does not have a valid value in the river network (LDD)

Where MaskMap

What to do You put in a coordinate as MaskMap, but at this coordinate there is no valid river network value

Module datahandling, maskfrompoint

6.5 E105

Description The map you are loading has a different shape (different cols,rows) than the other maps

Where Any map

What to do Make sure your map has the same resolution, rows, cols than the river network map

Module datahandling, compressArray

6.6 E106

Description The map you are loading has missing cell values (NaNs) where the river network and the mask map has valid values

Where Any map

What to do Check the map and include cell values where the river network map has valid (non NaNs) values

Module datahandling, compressArray

6.7 E107

Description The netCDF map has different attributes (resolution, rows,cols) than your standard map

Where Any netCDF map

What to do Make sure your map has the same resolution, rows, cols as the river network map

Module datahandling, mapattrNetCDF

6.8 E108

Description The tif map has different attributes (resolution, rows,cols) than your standard map

Where Any tif map

What to do Make sure your map has the same resolution, rows, cols as the river network map

Module datahandling, mapattrTiff

6.9 E109

Description The meteo map (e.g. temperature, precipitation, evaporation..) has different attributes (resolution, rows,cols) than your standard map

Where Meteo netCDF map

What to do Make sure your meteo input maps have the same resolution, rows, cols as the river network map. If it is the ET maps, it might be from another run with different mask. Please look at the option: calc_evaporation

Module datahandling, readmeteodata

6.10 E110

Description The netcdf map with a time variable (e.g. waterdemand, land cover, lakes..) has different attributes (resolution, rows,cols) than your standard map

Where All time depending netCDF maps appart from meteo maps

What to do Make sure your netCDF input maps have the same resolution, rows, cols as the river network map

Module datahandling, readnetcdf2

6.11 E111

Description The netcdf maps without time is turned upside dowm

Where Any netCDF map without time

What to do Make sure your map is in right order e.g. latitude coordinate are turned

Module datahandling, readnetcdfWithoutTime

6.12 E112

Description The initial map (load_initial) stack is turned upside down

Where Initload

What to do Make sure your initial map (in initial_load) is in right order e.g. latitude coordinate are turned

Module datahandling, readnetcdfInitial

6.13 E113

Description The initial map stack (initial_load) has different attributes (resolution, rows,cols) than your standard map

Where Initload

What to do Make sure your initial maps (in initial_load)has the same attributes. Maybe your initial maps are from a different run with a different mask?

Module datahandling, readnetcdfInitial

6.14 E114

Description Problem reading initial maps

Where Initload

What to do Initial maps has maybe not the same shape as the mask map. Maybe put load_initial = False

Module datahandling, readnetcdfInitial

6.15 E115

Description Variable with a True or False value can not be read

Where Any flag (True or False)

What to do Make sure the variable has either: varname = False or varname = True. Ttrue or faalse is not working!

Module datahandling, returnBool

6.16 E116

Description One of the variable names in [Option] is written wrong

Where Any variable in [Option]

What to do A keyword in option is maybe written wrong e.g. CaaapillarRise instead CapillarRise

Module datahandling, checkOption

6.17 E117

Description One of the variable names is written wrong

Where Any variable after [Option]

What to do A keyword in option is maybe written wrong e.g. MaaaskMap instead MaskMap. Pay attention: in Linux words are case sensitive!

Module datahandling, cbinding

6.18 E118

Description Timing in the section TIME-RELATED_CONSTANTS is wrong

Where StepStart, SpinUp, StepEnd

What to do Please check the variables StepStart, SpinUp, StartEnd

Module timestep, ctbinding

6.19 E119

Description Either date in StepStart is not a date or in SpinUp or StepEnd it is neither a number or a date

Where StepStart, SpinUp, StepEnd

What to do Please check the variables StepStart, SpinUp, StartEnd - A date is missing

Module timestep, Calendar

6.20 E120

Description First date in StepInit is neither a number or a date

Where StepInit

What to do Check StepInit in INITITIAL CONDITIONS. It is not a date or a number

Module timestep, Calendar

6.21 E121

Description Second value in StepInit is not a number or date nor indicating a repetition of year(y), month(m) or day(d) e.g. 2y for every 2 years or 6m for every 6 month

Where StepInit

What to do Check StepInit. The second part is neither a date or d,m,y

Module timestep, datetosaveInit

6.22 E122

Description Third value in StepInit is not an integer after ‘y’ or ‘m’ or ‘d’

Where StepInit

What to do Check StepInit. The third part after d,m,y is not an integer

Module timestep, datetosaveInit

6.23 E123

Description StepStart has to be a valid date

Where StepStart

What to do Check StepStart. It has to be a date e.g. 01/01/2009

Module timestep, checkifDate

6.24 E124

Description StepEnd is earlier than StepStart

Where StepStart, StepEnd

What to do Check StepStart and StepEnd. StepEnd has to be later than StepStart

Module timestep, checkifDate

6.25 E125

Description Spin Date is smaller/bigger than the first/last time step date

Where StepStart, SpinUp, StepEnd

What to do Check that SpinDate is in between StepStart and StepEnd

Module timestep, checkifDate

6.26 E127

Description Coordinates are not pairs

Where InflowPoints

What to do Check that location in InflowPoints comes in pairs

Module inflow, initial

6.27 E128

Description OUT_MAP_Daily = discharge may be not defined in [OUTPUT]

Where calc_ef_afterRun, Out_MAP_Daily

What to do Make sure that you define Out_MAP_Daily = discharge. Is seems daily discharge is not stored

Module environflow, initial

6.28 E129

Description Output points in Gauges are not pairs e.g. Gauges = 17.5 55.6 18.5

Where Gauges

What to do Check that output-points in Gauges are of coordinate e.g. Gauges = 6.25 51.75

Module output, initial

6.29 E130

Description Out_TSS or Out_MAP is not one of these: daily, monthend, monthtot, monthavg, annualend, annualtot, annualavg

Where Out_TSS, Out_MAP

What to do Please check that the wording after Out_TSS or Out_MAP is correct. Only a few keywords are valid e.g. Out_MAP_Daily, Out_TSS_monthavg

Module output, initial

6.30 E131

Description Second keyword after TSS or MAP is wrong

Where Out_TSS, Out_MAP

What to do Use TSS for point value, AreaSum for sum of area, AreaAvg for average of area e.g. OUT_TSS_AreaSum_Daily

Module output, initial

6.31 E132

Description Variable is not defined in list of variables

Where Out_TSS, Out_MAP

What to do Please correct the writing of the variable name, check also case sensitive e.g. Precipitation instead precipitation

Module output, dynamic

6.32 E201

Description Cannot load the maskmap as a file e.g. MaskMap = notexisting.tif

Where MaskMap

What to do Make sure the file you put in MaskMap is existing and is a map e.g. .nc, .map, .tif

Module datahandling, loadsetclone

6.33 E202

Description Your map is upside down

Where Any map

What to do Make sure your map is in right order e.g. latitude coordinate are turned

Module datahandling, loadmap

6.34 E203

Description Cannot find the map, filename does not exists at this location

Where Any map

What to do the datafile cannot be found, the location is wrong or the file is missing

Module datahandling, loadmap

6.35 E204

Description Trying to read the precipitation map metadata

Where Precipitation map

What to do Make sure the precipitation netcdf are existing and correct

Module datahandling, metaNetCDF

6.36 E205

Description Read error while reading netcdf map

Where any netcdf map

What to do Make sure netcdf map is existing and correct

Module datahandling, readCoordNetCDF

6.37 E206

Description read error while reading meteo maps

Where any meteo maps

What to do Make sure the meteo input maps are in the right location and chave correct attributes

Module datahandling, checkMeteo_Wordclim

6.38 E207

Description read error while reading the wordclim map for downscaling meteo maps

Where Wordlclim maps

What to do Make sure the wordclim maps are in the right location and chave correct attributes

Module datahandling, checkMeteo_Wordclim

6.39 E208

Description Cannot find meteomaps

Where any meteomap

What to do Make sure the meteo input maps are at the right location

Module datahandling, multinetdf

6.40 E209

Description Error loading meteomaps

Where any meteomap

What to do Check if the filenames of the meteomaps are ok. Does the filename exist at this location

Module datahandling, multinetdf

6.41 E210

Description Meteomap does not have this date

Where any meteomap

What to do The netCDF file does not contain this date. Check your start and end date

Module datahandling, readmeteodata

6.42 E211

Description Error loading meteomaps

Where any meteomap

What to do Check if the filenames of the meteomaps are ok. Does the filename exist at this location

Module datahandling, readmeteodata

6.43 E212

Description Error loading any other time depending map but not meteomaps

Where any time dependending netCDF map

What to do Check if the filenames of the map is ok. Does the filename exist at this location

Module datahandling, readnetcdf2

6.44 E213

Description Error loading netCDF maps without time

Where any non time depending netCDF map

What to do Check if the filenames of the map is ok. Does the filename exist at this location

Module datahandling, readnetcdfWithoutTime

6.45 E214

Description Error loading the initial map

Where initial netCDF map

What to do Check if the loaction of the initial map is o. Please check filename in initLoad

Module datahandling, readnetcdfInitial

6.46 E215

Description Trying to read the precipitation map metadata

Where Precipitation map

What to do Make sure the precipitation netcdf are existing and correct

Module timestep, checkifDate

6.47 E216

Description Outflow/Inflow point file cannot be loaded

Where InflowPoints

What to do Check if filename of InflowPoints exists

Module inflow, initial

6.48 E217

Description Mistake reading inflow file, name of inflow points are used twice

Where In_Dir

What to do Check inflow file header

Module inflow, initial

6.49 E218

Description Mistake reading inflow file

Where In_Dir

What to do Check if filename of inflow file exists

Module inflow, initial

6.50 E219

Description Mistake in discharge daily netcdf file

Where Out_MAP_Daily = discharge

What to do Something is wrong with the daily discharge file. Check location and content

Module environment, initial

6.51 E220

Description Output file path is wrong

Where OUT_Dir, PathOut

What to do Check OUT_Dir and PathOut. OUT_Dir can be used several times in the settings file.

Module output, appendinfo

6.52 E221

Description Error using file in Gauges

Where Gauges

What to do Please check if file in Gauges exists and is correct

Module output, initial

6.53 E301

Description Python version is not 64 bit

Where

-

What to do Make sure that you use a 64 bit version of Python

Module global, main

6.54 E302

Description Settingsfile not found

Where

-

What to do Make sure the settings file exists and is at the right location

Module configuration, parse_configuration

6.55 E303

Description An error occured while reading metaNetcdf.xml

Where metaNetcdfFile

What to do Please check that metaNetcdf is a valid .xml file

Module configuration, read_metanetcdf

6.56 E304

Description Cannot find alternative option file metaNetcdf.xml not found

Where metaNetcdfFile

What to do Please check in metaNetcdfFile that the file exists

Module configuration, read_metanetcdf

6.57 E305

Description An error occured while reading alternative metaNetcdf.xml

Where metaNetcdfFile

What to do Please check that metaNetcdf is a valid .xml file

Module configuration, read_metanetcdf

CHAPTER

SEVEN

LIST OF OUTPUT VARIABLES

This list of variables gives an overview of possible output variables.

> **Warning:** this list is still under construction. We have to complete descriptions and units for all variables

ID	VariableName	Description	Unit	Module
V1	modflow	Flag: True if modflow_coupling = True in settings file	•	capillarRise
V2	storGroundwater	simulated groundwater storage	m	capillarRise
V3	specificYield	groundwater reservoir parameters (if ModFlow is not used) used to compute groundwater levels in capillary rise module	m	capillarRise
V4	maxGWCapRise	influence of capillary rise above groundwater level	m	capillarRise
V5	dzRel0100	map of relative elevation above flood plains (max elevation above plain)	•	capillarRise
V6	dzRel0090	map of relative elevation above flood plains (90% elevation above plain)	•	capillarRise
V7	dzRel0080	map of relative elevation above flood plains (80% elevation above plain)	•	capillarRise
V8	dzRel0070	map of relative elevation above flood plains (70% elevation above plain)	•	capillarRise

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V9	dzRel0060	map of relative elevation above flood plains (60% elevation above plain)	•	capillarRise
V10	dzRel0050	map of relative elevation above flood plains (median elevation above plain)	•	capillarRise
V11	dzRel0040	map of relative elevation above flood plains (40% elevation above plain)	•	capillarRise
V12	dzRel0030	map of relative elevation above flood plains (30% elevation above plain)	•	capillarRise
V13	dzRel0020	map of relative elevation above flood plains (20% elevation above plain)	•	capillarRise
V14	dzRel0010	map of relative elevation above flood plains (10% elevation above plain)	•	capillarRise
V15	dzRel0005	map of relative elevation above flood plains (5% elevation above plain)	•	capillarRise
V16	dzRel0001	map of relative elevation above flood plains (1% elevation above plain)	•	capillarRise
V17	capRiseFrac	fraction of a grid cell where capillar rise may happen	•	capillarRise
V18	cut_ef_map	if TRUE calculated maps of environmental flow are cut to the extend of the area map	•	environflow
V19	MAF	Mean of discharge for all days	m3/s	environflow
V20	Q90	10% of the lowest discharge for all days	m3/s	environflow
V21	MMF	Mean of discharge for each month separately	m3/s	environflow

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V22	MQ90	10% of lowest discharge for each month separately	m3/s	environflow
V23	EF_VMF	EF requirement with Variable Monthly Flow: Pastor et al.(2014): Accounting for environmental flow requirements in global water assessments, Hydrol Earth Syst Sci, 18, p5041-5059	m3/s	environflow
V25	potBareSoilEvap	potential bare soil evaporation (calculated with minus snow evaporation)	m	evaporation
V26	cropCorrect	calibrated factor of crop KC factor	•	evaporation
V27	minCropKC	minimum crop factor (default 0.2)	•	evaporation
V28	ETRef	potential evapotranspiration rate from reference crop	m	evaporation
V29	snowEvap	total evaporation from snow for a snow layers	m	evaporation
V30	SnowMelt	total snow melt from all layers	m	evaporation
V31	cropKC	crop coefficient for each of the 4 different land cover types (forest, irrigated, paddy, others)	•	evaporation
V32	totalPotET	Potential evaporation per land use class	m	evaporation
V33	potTranspiration	Potential transpiration (after removing of evaporation)	m	evaporation
V35	Precipitation	Precipitation (input for the model)	m	evaporation
V36	Rain	Precipitation less snow	m	evaporation
V37	prevSnowCover	snow cover of previous day (only for water balance)	m	evaporation

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V38	SnowCover	snow cover (sum over all layers)	m	evaporation
V40	pet_modus	Flag: index which ETP approach is used e.g. 1 for Penman-Monteith	•	evaporationPot
V41	AlbedoCanopy	Albedo of vegetation canopy (FAO,1998) default =0.23	•	evaporationPot
V42	AlbedoSoil	Albedo of bare soil surface (Supit et. al. 1994) default = 0.15	•	evaporationPot
V43	AlbedoWater	Albedo of water surface (Supit et. al. 1994) default = 0.05	•	evaporationPot
V44	TMin	minimum air temperature	K	evaporationPot
V45	TMax	maximum air temperature	K	evaporationPot
V46	Psurf	Instantaneous surface pressure	Pa	evaporationPot
V47	Qair	specific humidity	kg/kg	evaporationPot
V48	Tavg	average air Temperature (input for the model)	K	evaporationPot
V49	Rsdl	long wave downward surface radiation fluxes	W/m2	evaporationPot
V50	albedoLand	albedo from land surface (from GlobAlbedo database)	•	evaporationPot
V51	albedoOpenWater	albedo from open water surface (from GlobAlbedo database)	•	evaporationPot
V52	Rsds	short wave downward surface radiation fluxes	W/m2	evaporationPot
V53	Wind	wind speed	m/s	evaporationPot
V55	EWRef	potential evaporation rate from water surface	m	evaporationPot
V56	adminSegments			evaporation_FUSE
V57	Crops			evaporation_FUSE
V58	cropKC_10day		•	evaporation_FUSE

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V59	fracCrops		•	evaporation_FUSE
V60	fracVegCover	Fraction of area covered by the corresponding landcover type		evaporation_FUSE
V61	fracCrops_Irr			evaporation_FUSE
V62	areaCrops_Irr_segment			evaporation_FUSE
V63	cellArea	Cell area [m^2] of each simulated mesh		evaporation_FUSE
V64	fracCrops_nonIrr			evaporation_FUSE
V65	activatedCrops			evaporation_FUSE
V66	monthCounter			evaporation_FUSE
V67	ratio_a_p_nonIrr			evaporation_FUSE
V68	totalPotET_month_nonIrr			evaporation_FUSE
V69	actTransTotal_month_nonIrr			evaporation_FUSE
V70	Yield_nonIrr			evaporation_FUSE
V71	currentKY			evaporation_FUSE
V72	currentKC			evaporation_FUSE
V73	PET_Sugar1_segments			evaporation_FUSE
V74	PET_Sugar2_segments			evaporation_FUSE
V75	PET_Sugar3_segments			evaporation_FUSE
V76	PET_Sorghum_segments			evaporation_FUSE
V77	PET_crop			evaporation_FUSE
V78	PET_crop_segments			evaporation_FUSE
V79	ETRefAverage_segments			evaporation_FUSE
V80	rainAverage_segments			evaporation_FUSE
V81	recessionCoeff	groundwater storage times this coefficient gives baseflow	•	groundwater
V82	kSatAquifer	groundwater reservoir parameters (if ModFlow is not used), could be used to compute the recession coefficient	m day-1	groundwater
V85	prestorGroundwater	storGroundwater at the beginning of each step	m	groundwater
V86	readAvlStorGroundwater	same as storGroundwater but equal to 0 when inferior to a treshold	m	groundwater
V88	sum_gwRecharge	groundwater recharge	m	groundwater

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V89	nonFossilGroundwaterAbs	groundwater abstraction which is sustainable and not using fossil resources	m	groundwater
V92	InvCellArea	Inverse of cell area of each simulated mesh	m-1	groundwater
V93	baseflow	simulated baseflow (= groundwater discharge to river)	m	groundwater
V94	capillar	Simulated flow from groundwater to the third CWATM soil layer	m	groundwater
V95	MtoM3	Coefficient to change units	•	groundwater
V97	sum_prefFlow	preferential flow from soil to groundwater (summed up for all land cover classes)	m	groundwater
V99	sum_perc3toGW	percolation from 3rd soil layer to groundwater (summed up for all land cover classes)	m	groundwater
V101	sum_capRiseFromGW	capillar rise from groundwater to 3rd soil layer (summed up for all land cover classes)	m	groundwater
V106	sum_landSurfaceRunoff	Runoff concentration above the soil more interflow including all landcover types	m	groundwater
V108	totalET	Total evapotranspiration for each cell including all landcover types	m	groundwater
V111	sampleInflow	location of inflow point	lat/lon	inflow
V112	noinflowpoints	number of inflow points	•	inflow
V113	inflowTs	inflow time series data	m3/s	inflow

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V114	QInM3Old	Inflow from previous day	m3	inflow
V115	totalQInM3	total inflow over time (for mass balance calculation)	m3	inflow
V116	inflowM3	inflow to basin	m3	inflow
V117	DtSec	number of seconds per timestep (default = 86400)	s	inflow
V118	coverTypes	land cover types - forest - grassland - irrPaddy - irrNonPaddy - water - sealed	•	initcondition
V119	loadInit	Flag: if true initial conditions are loaded	•	initcondition
V120	initLoadFile	load file name of the initial condition data	•	initcondition
V121	saveInit	Flag: if true initial conditions are saved	•	initcondition
V122	saveInitFile	save file name of the initial condition data	•	initcondition
V124	discharge	discharge	m3/s	initcondition
V127	interceptCap	interception capacity of vegetation	m	interception
V128	minInterceptCap	Maximum interception read from file for forest and grassland land cover	m	interception
V129	interceptStor	simulated vegetation interception storage	m	interception
V130	availWaterInfiltration	quantity of water reaching the soil after interception, more snowmelt	m	interception
V131	twothird	2/3	•	interception
V132	interceptEvap	simulated evaporation from water intercepted by vegetation	m	interception

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V133	actualET	simulated evapotranspiration from soil, flooded area and vegetation	m	interception
V134	waterBodyID	lakes/reservoirs map with a single ID for each lake/reservoir	•	lakes_reservoirs
V137	UpArea1	upstream area of a grid cell	m2	lakes_reservoirs
V138	waterBodyOut	biggest outlet (biggest accumulation of ldd network) of a waterbody	•	lakes_reservoirs
V139	dirUp	river network in upstream direction	•	lakes_reservoirs
V140	ldd_LR	change river network (put pits in where lakes are)	•	lakes_reservoirs
V141	lddCompress	compressed river network (without missing values)	•	lakes_reservoirs
V142	lddCompress_LR	compressed river network lakes/reservoirs (without missing values)	•	lakes_reservoirs
V143	dirUp_LR	river network direction upstream lake/reservoirs	•	lakes_reservoirs
V144	dirupLen_LR	number of bifurcation upstream lake/reservoir	•	lakes_reservoirs
V145	dirupID_LR	index river upstream lake/reservoir	•	lakes_reservoirs
V146	downstruct_LR	river network downstream lake/reservoir	•	lakes_reservoirs
V147	catchment_LR	catchments lake/reservoir	•	lakes_reservoirs
V148	dirDown_LR	river network direktion downstream lake/reservoir	•	lakes_reservoirs
V149	lendirDown_LR	number of river network connections lake/reservoir	•	lakes_reservoirs

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V150	compress_LR	boolean map as mask map for compressing lake/reservoir	•	lakes_reservoirs
V151	decompress_LR	boolean map as mask map for decompressing lake/reservoir	•	lakes_reservoirs
V152	waterBodyOutC	compressed map biggest outlet of each lake/reservoir	•	lakes_reservoirs
V153	resYearC	compressed map of the year when the reservoirs is operating	•	lakes_reservoirs
V154	waterBodyTypC	water body types 3 reservoirs and lakes (used as reservoirs but before the year of construction as lakes) 2 reservoirs (regulated discharge) 1 lakes (weirFormula)	•	lakes_reservoirs
V155	lakeArea	area of each lake/reservoir	m2	lakes_reservoirs
V156	lakeAreaC	compressed map of the area of each lake/reservoir	m2	lakes_reservoirs
V157	lakeDis0	compressed map average discharge at the outlet of a lake/reservoir	m3 s-1	lakes_reservoirs
V158	lakeDis0C	average discharge at the outlet of a lake/reservoir	m3 s-1	lakes_reservoirs
V159	lakeAC	compressed map of parameter of channel width, gravity and weir coefficient	•	lakes_reservoirs
V160	resVolumeC	compressed map of reservoir volume	Million m3	lakes_reservoirs
V161	waterBodyIDC	compressed map of water body index	•	lakes_reservoirs
V162	lakeEvaFactor	a factor which increases evaporation from lake because of wind	•	lakes_reservoirs

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V163	lakeEvaFactorC	compressed map of a factor which increases evaporation from lake because of wind	•	lakes_reservoirs
V165	lakeResInflowDis	inflow to lakes/reservoirs	m3/s	lakes_reservoirs
V166	reslakeoutflow		m	lakes_reservoirs
V167	reslakeinflow	inflow to lakes/reservoirs	m	lakes_reservoirs
V168	lakeVolume	volume of lakes	m3	lakes_reservoirs
V169	outLake	outflow from lakes	m	lakes_reservoirs
V170	lakeStorage			lakes_reservoirs
V171	lakeInflow			lakes_reservoirs
V172	lakeOutflow			lakes_reservoirs
V173	reservoirStorage			lakes_reservoirs
V174	MtoM3C	conversion factor from m to m3 (compressed map)	•	lakes_reservoirs
V175	EvapWaterBodyM			lakes_reservoirs
V176	lakeResInflowM			lakes_reservoirs
V177	lakeResOutflowM			lakes_reservoirs
V178	lakeInflowOldC	inflow to the lake from previous days	m/3	lakes_reservoirs
V180	ChanQ			lakes_reservoirs
V181	LakeIndex			lakes_reservoirs
V182	chanQKin			lakes_reservoirs
V183	lakeFactor	factor for the Modified Puls approach to calculate retention of the lake	•	lakes_reservoirs
V184	dtRouting	number of seconds per routing timestep	s	lakes_reservoirs
V185	lakeFactorSqr	square root factor for the Modified Puls approach to calculate retention of the lake	•	lakes_reservoirs
V186	lakeVolumeM3C	compressed map of lake volume	m3	lakes_reservoirs
V187	lakeStorageC		m3	lakes_reservoirs
V188	lakeOutflowC	compressed map of lake outflow	m3/s	lakes_reservoirs
V189	lakeLevelC	compressed map of lake level	m	lakes_reservoirs
V190	conLimitC			lakes_reservoirs
V191	normLimitC			lakes_reservoirs
V192	floodLimitC			lakes_reservoirs
V193	adjust_Normal_FloodC			lakes_reservoirs
V194	norm_floodLimitC			lakes_reservoirs

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V195	ID			lakes_reservoirs
V196	minQC			lakes_reservoirs
V197	normQC			lakes_reservoirs
V198	nondmgQC			lakes_reservoirs
V199	deltaO			lakes_reservoirs
V200	deltaLN			lakes_reservoirs
V201	deltaLF			lakes_reservoirs
V202	deltaNFL			lakes_reservoirs
V203	reservoirFillC			lakes_reservoirs
V204	reservoirStorageM3C			lakes_reservoirs
V205	lakeResStorageC			lakes_reservoirs
V206	lakeResStorage			lakes_reservoirs
V207	prelakeResStorage			lakes_reservoirs
V208	waterBodyTypCTemp			lakes_reservoirs
V209	waterBodyTypCTemp			lakes_reservoirs
V210	sumEvapWaterBodyC			lakes_reservoirs
V211	sumlakeResInflow			lakes_reservoirs
V212	sumlakeResOutflow			lakes_reservoirs
V213	lakeResStorage_release_ratio			lakes_reservoirs
V214	lakeResStorage_release_ratioC			lakes_reservoirs
V215	lakeIn			lakes_reservoirs
V216	lakeEvapWaterBodyC			lakes_reservoirs
V217	evapWaterBodyC			lakes_reservoirs
V218	sumLakeEvapWaterBodyC			lakes_reservoirs
V219	noRoutingSteps			lakes_reservoirs
V220	QLakeOutM3Dt			lakes_reservoirs
V221	resEvapWaterBodyC			lakes_reservoirs
V222	sumResEvapWaterBodyC			lakes_reservoirs
V223	InvDtSec			lakes_reservoirs
V224	outflow			lakes_reservoirs
V225	runoff			lakes_reservoirs
V226	sumEvapWaterBodyC			lakes_reservoirs
V227	sumlakeResInflow			lakes_reservoirs
V228	sumlakeResOutflow			lakes_reservoirs
V229	reservoirStorageM3C			lakes_reservoirs
V230	lakeResStorageC			lakes_reservoirs
V231	EvapWaterBodyM_segments			lakes_reservoirs
V232	lakeResStorage_segments			lakes_reservoirs
V233	lakeResInflowM_segments			lakes_reservoirs
V234	lakeResOutflowM_segments			lakes_reservoirs
V235	SUMsumEvapWaterBodyC			lakes_reservoirs
V236	lakeResOutflowDis			lakes_reservoirs
V237	smallpart			lakes_res_small
V238	smalllakeArea			lakes_res_small
V239	smalllakeDis0			lakes_res_small
V240	smalllakeA			lakes_res_small
V241	smalllakeFactor			lakes_res_small
V242	smalllakeFactorSqr			lakes_res_small
V243	smalllakeInflowOld			lakes_res_small

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V244	smalllakeVolumeM3			lakes_res_small
V245	smalllakeOutflow			lakes_res_small
V246	smalllakeLevel			lakes_res_small
V247	smalllakeStorage			lakes_res_small
V248	minsmalllakeVolumeM3			lakes_res_small
V249	preSmalllakeStorage			lakes_res_small
V250	smallLakeIn			lakes_res_small
V251	smallevapWaterBody			lakes_res_small
V252	minsmalllakeStorageM3			lakes_res_small
V253	smalllakeStorageM3			lakes_res_small
V254	smallLakeout			lakes_res_small
V255	smallLakeDiff			lakes_res_small
V256	smallrunoffDiff			lakes_res_small
V257	dynamicLandcover			landcoverType
V258	soilLayers	Number of soil layers	•	landcoverType
V259	landcoverSum			landcoverType
V260	act_SurfaceWaterAbstract			landcoverType
V261	totalSto_segments			landcoverType
V262	sum_runoff_segments			landcoverType
V263	total_baseflow			landcoverType
V264	total_channelStorage			landcoverType
V265	sum_interceptStor	Total of simulated vegetation interception storage including all landcover types	m	landcoverType
V266	minTopWaterLayer			landcoverType
V267	cropDeplFactor			landcoverType
V268	rootFraction1			landcoverType
V269	rootFraction2			landcoverType
V270	maxRootDepth			landcoverType
V271	rootDepth			landcoverType
V272	soildepth	Thickness of the first soil layer	m	landcoverType
V273	soildepth12	Total thickness of layer 2 and 3	m	landcoverType
V274	KSat1			landcoverType
V275	KSat2			landcoverType
V276	KSat3			landcoverType
V277	alpha1			landcoverType
V278	alpha2			landcoverType
V279	alpha3			landcoverType
V280	lambda1			landcoverType
V281	lambda2			landcoverType
V282	lambda3			landcoverType
V283	thetas1			landcoverType
V284	thetas2			landcoverType
V285	thetas3			landcoverType

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V286	thetar1			landcoverType
V287	thetar2			landcoverType
V288	thetar3			landcoverType
V289	GenuM1			landcoverType
V290	genuM1			landcoverType
V291	genuM2			landcoverType
V292	genuM3			landcoverType
V293	GenuInvM1			landcoverType
V294	genuInvM1			landcoverType
V295	genuInvM2			landcoverType
V296	genuInvM3			landcoverType
V297	GenuInvN1			landcoverType
V298	genuInvN1			landcoverType
V299	genuInvN2			landcoverType
V300	genuInvN3			landcoverType
V301	invAlpha1			landcoverType
V302	invAlpha2			landcoverType
V303	invAlpha3			landcoverType
V304	ws1	Maximum storage capacity in layer 1	m	landcoverType
V305	ws2	Maximum storage capacity in layer 2	m	landcoverType
V306	ws3	Maximum storage capacity in layer 3	m	landcoverType
V307	wres1	Residual storage capacity in layer 1	m	landcoverType
V308	wres2	Residual storage capacity in layer 2	m	landcoverType
V309	wres3	Residual storage capacity in layer 3	m	landcoverType
V310	wrange1			landcoverType
V311	wrange2			landcoverType
V312	wrange3			landcoverType
V313	wfc1	Soil moisture at field capacity in layer 1		landcoverType
V314	wfc2	Soil moisture at field capacity in layer 2		landcoverType
V315	wfc3	Soil moisture at field capacity in layer 3		landcoverType
V316	wwp1	Soil moisture at wilting point in layer 1		landcoverType
V317	wwp2	Soil moisture at wilting point in layer 2		landcoverType

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V318	wwp3	Soil moisture at wilting point in layer 3		landcoverType
V319	kUnSat3FC			landcoverType
V320	kunSatFC12			landcoverType
V321	kunSatFC23			landcoverType
V322	cropCoefficientNC_filename			landcoverType
V323	interceptCapNC_filename			landcoverType
V324	coverFractionNC_filename			landcoverType
V325	interflow	Simulated flow reaching runoff instead of groundwater	m	landcoverType
V326	w1	Simulated water storage in the layer 1	m	landcoverType
V327	w2	Simulated water storage in the layer 2	m	landcoverType
V328	w3	Simulated water storage in the layer 3	m	landcoverType
V329	topwater	quantity of water above the soil (flooding)	m	landcoverType
V330	sum_topwater	quantity of water on the soil (flooding) (weighted sum for all landcover types)	m	landcoverType
V331	totalSto	Total soil,snow and vegetation storage for each cell including all landcover types	m	landcoverType
V332	sum_w1			landcoverType
V333	sum_w2			landcoverType
V334	sum_w3			landcoverType
V335	arnoBetaOro			landcoverType
V336	ElevationStD			landcoverType
V337	arnoBeta			landcoverType
V338	adjRoot			landcoverType
V339	maxtopwater	maximum heigth of topwater	m	landcoverType
V340	landcoverSumSum			landcoverType
V341	totAvlWater			landcoverType
V342	gwstore			landcoverType
V343	pregwstore			landcoverType
V344	GWVolumeVariation			landcoverType

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V345	ActualPumpingRate	Actual pumping rate occuring in ModFlow [m3/timestep]		landcoverType
V346	current_modflowPumpingM		m	landcoverType
V347	riceWeight			landcoverType
V348	sum_fracVegCover			landcoverType
V349	modflow_timestep	Chosen ModFlow model timestep (1day, 7days, 30days…)		landcoverType
V350	presumed_sum_gwRecharge	Previous groundwater recharge [m/timestep] (used for the ModFlow version)	m	landcoverType
V351	sumed_sum_gwRecharge			landcoverType
V352	pretotalSto	Previous totalSto	m	landcoverType
V355	modflowPumpingM			landcoverType
V359	sum_directRunoff			landcoverType
V360	sum_actTransTotal			landcoverType
V361	sum_actBareSoilEvap			landcoverType
V362	sum_openWaterEvap			landcoverType
V363	sum_interceptEvap			landcoverType
V364	addtoevapotrans			landcoverType
V365	sum_runoff	Runoff above the soil, more interflow, including all landcover types	m	landcoverType
V366	sum_interflow			landcoverType
V367	nonIrrDemand			landcoverType
V368	totalIrrDemand			landcoverType
V369	totalETM3_segments			landcoverType
V370	rainM3_segments			landcoverType
V371	channelStorage_segments			landcoverType
V372	channelStorage			landcoverType
V373	prechannelStorage_segments			landcoverType
V374	prechannelStorage			landcoverType
V375	prelakeResStorage_segments			landcoverType
V376	pretotalSto_segments			landcoverType
V377	storGroundwater_segments			landcoverType
V378	prestorGroundwater_segments			landcoverType
V379	sum_interceptStor_segments			landcoverType
V380	totalET_segments			landcoverType
V381	EvapoChannel_segments			landcoverType
V382	EvapoChannel			landcoverType
V383	act_nonIrrConsumption_segments			landcoverType
V384	act_nonIrrConsumption			landcoverType
V385	gwstore_segments			landcoverType
V386	GWVolumeVariation_segments			landcoverType
V387	capillar_segments			landcoverType

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V388	baseflow_segments			landcoverType
V389	sum_gwRecharge_segments			landcoverType
V390	GW_Pumping_segments			landcoverType
V391	GW_Pumping			landcoverType
V392	ActualPumpingRate_segments			landcoverType
V393	sum_actTransTotal_segments			landcoverType
V394	actTransTotal_forest_segments			landcoverType
V395	actTransTotal_forest			landcoverType
V396	actTransTotal_grasslands_segments			landcoverType
V397	actTransTotal_grasslands			landcoverType
V398	actTransTotal_paddy_segments			landcoverType
V399	actTransTotal_paddy			landcoverType
V400	actTransTotal_nonpaddy_segments			landcoverType
V401	actTransTotal_nonpaddy			landcoverType
V402	sum_interceptEvap_segments			landcoverType
V403	sum_openWaterEvap_segments			landcoverType
V404	sum_actBareSoilEvap_segments			landcoverType
V405	act_totalIrrConsumption_segments			landcoverType
V406	act_totalIrrConsumption			landcoverType
V407	act_SurfaceWaterAbstract_segments			landcoverType
V408	sum_perc3toGW_segments			landcoverType
V409	sum_prefFlow_segments			landcoverType
V410	act_bigLakeResAbst_segments			landcoverType
V411	act_bigLakeResAbst			landcoverType
V412	act_SurfaceWaterAbstract			landcoverType
V413	current_modflowPumpingM			landcoverType
V414	act_totalWaterWithdrawal			landcoverType
V415	act_SurfaceWaterAbstract			landcoverType
V416	cwatbudg_old			landcoverType
V417	storcwat_old			landcoverType
V418	gwVariation_old			landcoverType
V419	presumed_sum_gwRecharge			landcoverType
V420	sum_gwRecharge			landcoverType
V421	baseflow			landcoverType
V422	capillar			landcoverType
V423	ActualPumpingRate			landcoverType
V424	GWVolumeVariation			landcoverType
V425	GWVolumeVariation			landcoverType
V426	demand_old			landcoverType
V427	GW_pumping			landcoverType
V428	sum_availWaterInfiltration			landcoverType
V429	sumirrConsumption			landcoverType
V430	waterWithdrawal			landcoverType
V431	nonIrruse			landcoverType
V432	returnFlow			landcoverType
V433	sumsum_Precipitation			landcoverType
V434	sumsum_gwRecharge			landcoverType
V437	cellLength	length of a grid cell	m	miscInitial
V438	PixelArea	area of a grid cell	m2	miscInitial

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V439	InvCellLength	inverse cell length	m-1	miscInitial
V440	DtDay	seconds in a timestep (default=86400)	s	miscInitial
V441	InvDtDay	inverse seconds in a timestep (default=86400)	s-1	miscInitial
V442	DtSecChannel	seconds in a substep of channel routing	s	miscInitial
V443	MMtoM	Coefficient to change units	•	miscInitial
V444	MtoMM	Coefficient to change units	•	miscInitial
V445	M3toM	Coefficient to change units	•	miscInitial
V446	con_precipitation	conversion factor for precipitation	•	miscInitial
V447	con_e	conversion factor for evaporation	•	miscInitial
V448	modflowsteady	True if modflow_steadystate = True in settings file	•	readmeteo
V449	preMaps	choose between steady state precipitation maps for steady state modflow or normal precipitation maps	•	readmeteo
V450	tempMaps	choose between steady state temperature maps for steady state modflow or normal maps	•	readmeteo
V451	evaTMaps	choose between steady state ETP water maps for steady state modflow or normal maps	•	readmeteo

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V452	eva0Maps	choose between steady state ETP reference maps for steady state modflow or normal maps	•	readmeteo
V453	wc2_tavg	High resolution WorldClim map for average temperature	K	readmeteo
V454	wc4_tavg	upscaled to low resolution WorldClim map for average temperature	K	readmeteo
V455	wc2_tmin	High resolution WorldClim map for min temperature	K	readmeteo
V456	wc4_tmin	upscaled to low resolution WorldClim map for min temperature	K	readmeteo
V457	wc2_tmax	High resolution WorldClim map for max temperature	K	readmeteo
V458	wc4_tmax	upscaled to low resolution WorldClim map for max temperature	K	readmeteo
V459	wc2_prec	High resolution WorldClim map for precipitation	m	readmeteo
V460	wc4_prec	upscaled to low resolution WorldClim map for precipitation	m	readmeteo
V461	demHigh	digital elevation model high resolution	m	readmeteo
V462	demAnomaly	digital elevation model anomaly (high resolution - low resolution)	m	readmeteo
V463	meteomapsscale	if meteo maps have the same extend as the other spatial static maps -> meteomapsscale = True	•	readmeteo
V464	meteodown	if meteo maps should be downscaled	•	readmeteo

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V465	prec	precipitation in m	m	readmeteo
V466	temp	average temperature in Celsius deg	C°	readmeteo
V467	Tmin	minimum temperature in Celsius deg	C°	readmeteo
V468	Tmax	maximum temperature in celsius deg	C°	readmeteo
V469	WtoMJ	Conversion factor from [W] to [MJ] for radiation: 86400 * 1E-6	•	readmeteo
V471	runoff_peak	peak time of runoff in seconds for each land use class	s	runoff_concentration
V472	tpeak_interflow	peak time of interflow	s	runoff_concentration
V473	tpeak_baseflow	peak time of baseflow	s	runoff_concentration
V474	maxtime_runoff_conc	maximum time till all flow is at the outlet	s	runoff_concentration
V475	runoff_conc	runoff after concentration - triangular-weighting method	m	runoff_concentration
V476	directRunoff	Simulated surface runoff	m	runoff_concentration
V477	landSurfaceRunoff	Runoff concentration above the soil more interflow	m	runoff_concentration
V478	openWaterEvap	Simulated evaporation from open areas	m	sealed_water
V479	actTransTotal	Total actual transpiration from the three soil layers	m	sealed_water
V480	actBareSoilEvap	Simulated evaporation from the first soil layer	m	sealed_water
V482	numberSnowLayers	Number of snow layers (up to 10)	•	snow_frost
V483	glaciertransportZone	Number of layers which can be mimiced as glacier transport zone	•	snow_frost
V485	deltaInvNorm	Quantile of the normal distribution (for different numbers of snow layers)	•	snow_frost

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V486	DeltaTSnow	Temperature lapse rate x std. deviation of elevation	C°	snow_frost
V487	SnowDayDegrees	day of the year to degrees: 360/365.25 = 0.9856	•	snow_frost
V488	summerSeasonStart	day when summer season starts = 165	•	snow_frost
V489	IceDayDegrees	days of summer (15th June-15th Sept.) to degree: 180/(259-165)	•	snow_frost
V490	SnowSeason	seasonal melt factor	m C°-1 day-1	snow_frost
V491	TempSnow	Average temperature at which snow melts	C°	snow_frost
V492	SnowFactor	Multiplier applied to precipitation that falls as snow	•	snow_frost
V493	SnowMeltCoef	Snow melt coefficient - default: 0.004	•	snow_frost
V494	IceMeltCoef	Ice melt coefficnet - default 0.007	•	snow_frost
V495	TempMelt	Average temperature at which snow melts	C°	snow_frost
V496	SnowCoverS	snow cover for each layer	m	snow_frost
V497	Kfrost	Snow depth reduction coefficient, (HH, p. 7.28)	m-1	snow_frost
V498	Afrost	Daily decay coefficient, (Handbook of Hydrology, p. 7.28)	•	snow_frost
V499	FrostIndexThreshold	Degree Days Frost Threshold (stops infiltration, percolation and capillary rise)	•	snow_frost
V500	SnowWaterEquivalent	Snow water equivalent, (based on snow density of 450 kg/m3) (e.g. Tarboton and Luce, 1996)	•	snow_frost

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V501	MaskMap	Mask map to limit calculation to a mask	•	snow_frost
V502	FrostIndex	FrostIndex - Molnau and Bissel (1983), A Continuous Frozen Ground Index for Flood Forecasting	•	snow_frost
V503	extfrostindex	Flag for second frostindex	•	snow_frost
V504	FrostIndexThreshold2	FrostIndex2 - Molnau and Bissel (1983), A Continuous Frozen Ground Index for Flood Forecasting		snow_frost
V505	frostInd1	forstindex 1		snow_frost
V506	frostInd2	frostindex 2		snow_frost
V507	frostindexS	array for frostindex		snow_frost
V508	Snow	Snow (equal to a part of Precipitation)	m	snow_frost
V512	percolationImp	Fraction of area covered by the corresponding landcover type	m	soil
V513	cropGroupNumber	soil water depletion fraction, Van Diepen et al., 1988: WOFOST 6.0, p.86, Doorenbos et. al 1978	•	soil
V514	cPrefFlow	Factor influencing preferential flow (flow from surface to GW)	•	soil
V515	act_irrConsumption	actual irrgation water consumption	m	soil
V516	rws	Transpiration reduction factor (in case of water stress)	•	soil
V517	prefFlow	Flow going directly from rainfall to groundwater	m	soil
V518	infiltration	Water actually infiltrating the soil	m	soil
V520	capRiseFromGW	Simulated capillary rise from groundwater	m	soil

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V521	NoSubSteps	Number of sub steps to calculate soil percolation	•	soil
V522	perc1to2	Simulated water flow from soil layer 1 to soil layer 2	m	soil
V523	perc2to3	Simulated water flow from soil layer 2 to soil layer 3	m	soil
V524	perc3toGW	Simulated water flow from soil layer 3 to groundwater	m	soil
V525	theta1	fraction of water in soil compartment 1 for each land use class	•	soil
V526	theta2	fraction of water in soil compartment 2 for each land use class	•	soil
V527	theta3	fraction of water in soil compartment 3 for each land use class	•	soil
V531	gwRecharge	groundwater recharge	m	soil
V534	nonIrrReturnFlow			waterbalance
V535	localQW			waterbalance
V536	channelStorageBefore			waterbalance
V537	sum_balanceStore			waterbalance
V538	sum_balanceFlux			waterbalance
V539	catchmentAll			waterbalance
V540	catchmentNo			waterbalance
V541	unmetDemand			waterbalance
V542	act_irrDemand			waterbalance
V543	act_nonIrrDemand			waterbalance
V544	returnflowIrr			waterbalance
V546	nonIrrReturnFlowFraction			waterbalance
V547	sumsideflow			waterbalance
V548	sumIrrDemand			waterbalance
V549	outlets			waterbalance
V550	sum_irrDemand			waterbalance
V551	sum_act_SurfaceWaterAbstract			waterbalance
V552	pretotalSoil			waterbalance
V553	totalSoil			waterbalance
V554	sumP			waterbalance
V555	sumETA			waterbalance
V556	noOutpoints			waterbalance
V557	evalCatch			waterbalance
V558	area			waterbalance

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V559	catchment			waterbalance
V560	sumRunoff			waterbalance
V561	sumDelta1			waterbalance
V562	sumDelta2			waterbalance
V563	sumAll			waterbalance
V564	allocSegments			waterdemand
V565	segmentArea			waterdemand
V566	reservoir_command_areas			waterdemand
V567	swAbstractionFraction			waterdemand
V568	domesticTime			waterdemand
V569	industryTime			waterdemand
V570	livestockTime			waterdemand
V571	domWithdrawalVar			waterdemand
V572	domConsumptionVar			waterdemand
V573	indWithdrawalVar			waterdemand
V574	indConsumptionVar			waterdemand
V575	livVar			waterdemand
V576	uselivestock			waterdemand
V577	demand_unit			waterdemand
V578	use_environflow			waterdemand
V579	unmetDemandPaddy			waterdemand
V580	unmetDemandNonpaddy			waterdemand
V581	efficiencyPaddy			waterdemand
V582	efficiencyNonpaddy			waterdemand
V583	returnfractionIrr			waterdemand
V584	alphaDepletion			waterdemand
V585	modflowPumping			waterdemand
V586	modflowDepth2			waterdemand
V587	modflowStorGW2			waterdemand
V588	modflowDepth2_segments			waterdemand
V589	modflowTopography			waterdemand
V590	crops			waterdemand
V591	head2			waterdemand
V592	demand_Segment			waterdemand
V593	lakeResStorage_ratio_CA			waterdemand
V594	lakeResStorage_ratio			waterdemand
V595	print_modflowPumpingM			waterdemand
V596	act_bigLakeResAbst_alloc			waterdemand
V597	act_channelAbstract			waterdemand
V598	act_LocalLakeAbstract			waterdemand
V599	leakageC_daily_segments			waterdemand
V600	leakageC			waterdemand
V601	act_indDemand			waterdemand
V602	act_domDemand			waterdemand
V603	act_livDemand			waterdemand
V604	totalWaterDemand			waterdemand
V605	act_irrWithdrawal			waterdemand
V606	act_nonIrrWithdrawal			waterdemand
V607	act_indConsumption			waterdemand

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V608	act_domConsumption			waterdemand
V609	act_livConsumption			waterdemand
V610	act_totalWaterConsumption			waterdemand
V611	ind_efficiency			waterdemand
V612	dom_efficiency			waterdemand
V613	liv_efficiency			waterdemand
V614	act_LocalLakeAbstract_segments			waterdemand
V615	envFlowm3s			waterdemand
V616	envFlow			waterdemand
V617	channelAlpha			waterdemand
V618	chanLength			waterdemand
V619	readAvlChannelStorageM			waterdemand
V620	industryDemand			waterdemand
V621	pot_industryConsumption			waterdemand
V622	domesticDemand			waterdemand
V623	pot_domesticConsumption			waterdemand
V624	livestockDemand			waterdemand
V625	pot_livestockConsumption			waterdemand
V626	pot_nonIrrConsumption			waterdemand
V627	pot_irrConsumption			waterdemand
V628	irrDemand			waterdemand
V629	minsmalllakeStorage			waterdemand
V630	act_smallLakeResAbst			waterdemand
V631	actLakeResAbst			waterdemand
V632	leakageC_daily			waterdemand
V633	pot_GroundwaterAbstract			waterdemand
V634	renewableAvlWater			waterdemand
V635	act_irrNonpaddyWithdrawal			waterdemand
V636	act_irrPaddyWithdrawal			waterdemand
V637	act_irrPaddyDemand			waterdemand
V638	act_irrNonpaddyDemand			waterdemand
V639	Pumping_daily			waterdemand
V640	act_indWithdrawal			waterdemand
V641	act_domWithdrawal			waterdemand
V642	act_livWithdrawal			waterdemand
V643	waterDemand			waterdemand
V644	returnFlow_segments			waterdemand
V645	addtoevapotrans_segments			waterdemand
V646	waterDemandLost			waterdemand
V647	waterDemandLostarea			waterdemand
V648	sum_IrrDemand			waterdemand
V649	sum_waterWithdrawal			waterdemand
V650	leakage_Veer			waterdemand_beforeAllocSegments2
V651	frac_used_Segment			waterdemand_beforeAllocSegments2
V652	lakeResStorage_alloc			waterdemand_beforeAllocSegments2
V653	metRemainSegment			waterdemand_beforeAllocSegments2
V654	pot_GroundwaterAbstract			waterdemand_beforeAllocSegments2
V655	act_nonIrrConsumption			waterdemand_beforeAllocSegments2
V656	waterquality			waterquality1

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V657	celllenght			waterquality1
V658	downdist			waterquality1
V659	totalCrossSectionArea			waterquality1
V660	travelDistance			waterquality1
V661	travelTime			waterquality1
V662	waterLevel			waterquality1
V663	waterTemperature			waterquality1
V664	dirupLen			routing_kinematic
V665	dirupID			routing_kinematic
V666	dirDown			routing_kinematic
V667	lendirDown			routing_kinematic
V668	ups			routing_kinematic
V669	UpArea			routing_kinematic
V670	beta			routing_kinematic
V671	chanMan			routing_kinematic
V672	chanGrad			routing_kinematic
V673	chanWidth			routing_kinematic
V674	chanDepth			routing_kinematic
V675	invbeta			routing_kinematic
V676	invchanLength			routing_kinematic
V677	invdtRouting			routing_kinematic
V678	totalCrossSectionAreaBankFull			routing_kinematic
V679	TotalCrossSectionAreaBankFull			routing_kinematic
V680	chanWettedPerimeterAlpha			routing_kinematic
V681	alpPower			routing_kinematic
V682	invchannelAlpha			routing_kinematic
V683	readAvlChannelStorage			routing_kinematic
V684	channelAlphaPcr			routing_kinematic
V685	chanLengthPcr			routing_kinematic
V686	SUMEvapoChannel			routing_kinematic
V687	sumbalance			routing_kinematic
V688	dynamicFracWat			routing_kinematic
V689	QDelta			routing_kinematic
V690	inflowDt			routing_kinematic
V691	Pumping_daily_segments			routing_kinematic
V692	disold			routing_kinematic
V693	modflowexe	Path to the ModFlow.exe file	•	groundwater_modflow
V694	PathModflow	Path to the ModFlow folder where input data and ModFlow runs are stored	•	groundwater_modflow
V695	PathModflowOutput	Path to the ModFlow folder where ModFlow runs are stored	•	groundwater_modflow
V696	res_ModFlow	Chosen ModFlow model resolution		groundwater_modflow

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V697	Ndays_steady	Number of steady state run before the transient simulation	•	groundwater_modflow
V698	nlay	Number of ModFlow layers	•	groundwater_modflow
V699	Gleesonindex			groundwater_modflow
V83	head_development			groundwater_modflow
V700	actual_thick	Array (nlays, nrows, ncols) of layers thickness	m	groundwater_modflow
V701	coef	A coefficient defined by the user to artificially increase layer thickness		groundwater_modflow
V702	delv2	Array (nlays, nrows, ncols) of layers thickness times coef	m	groundwater_modflow
V703	riverPercentage	Array (nrows, ncol) defining the percentage of rivers on each ModFlow cell		groundwater_modflow
V704	hk0	Array (nrows, ncol) defining the groundwater permeability [m/s]		groundwater_modflow
V705	poro	Array (nrows, ncol) defining the groundwater porosity []		groundwater_modflow
V707	basin	Array (nrows, ncol) defining wich cells are active (1 or 0)		groundwater_modflow
V708	waterTable3	Array (nrows, ncol) defining DRAIN altitude [masl] on each cell	m	groundwater_modflow
V709	botm	Array (nlays+1, nrows, ncols) of layers top and botm [masl]	m	groundwater_modflow
V710	modflowtotalSoilThickness	Array (nrows, ncol) used to compute water table depth in post-processing	m	groundwater_modflow
V711	nameModflowModel	Name of the ModFlow model (used for ModFlow output)		groundwater_modflow

Continued on next page

Table 1 – continued from previous page

ID	VariableName	Description	Unit	Module
V712	steady_previous	True if a previous simulated map is used to defined the initial water table		groundwater_modflow
V713	head	Simulated ModFlow water level [masl]	m	groundwater_modflow
V714	modflow_text_to_write			groundwater_modflow
V715	modflow_compteur	Counts each day relatively to the chosen ModFlow timestep, allow to run ModFlow only once by timestep		groundwater_modflow
V716	writeerror			groundwater_modflow
V717	nameerrorfile			groundwater_modflow
V718	storGroundwater1			groundwater_modflow
V719	modflowStorGW			groundwater_modflow
V720	modflowWaterLevel	Simulated ModFlow water level [masl]		groundwater_modflow
V721	premodflowWaterLevel			groundwater_modflow
V722	sumstorGW			groundwater_modflow
V723	sumstorGW2			groundwater_modflow
V724	totaldifference			groundwater_modflow
V725	prestorGroundwater1			groundwater_modflow
V726	rgw			groundwater_modflow
V727	GWVolumeVariation_cell			groundwater_modflow
V728	GW_storage_ModFlow			groundwater_modflow
V729	preGW_storage_ModFlow			groundwater_modflow
V730	waterTable3			ModFlow_modelV5
V731	modflowDepth			ModFlow_modelV5
V732	Volume_modflow			modflow_steady_transient
V734	modflowGwStore			modflow_steady_transient

CHAPTER

EIGHT

DEMO OF THE MODEL

8.1 Resolution

CWatM can be run globally at 0.5° or separately for any basin or any clipping of a global map.
Depending on the data provided the model can also run for any other resolutions (e.g. 5 arcmin).
Timestep is daily, output of maps, time series can be daily, monthly, yearly

Here some outputs of the global run on 0.5° are shown:

8.2 Demo 1 - NetCDF videos

8.2.1 Global discharge

One year run example: 1/1/1991- 31/12/1992

8.2.2 Global potential evaporation [mm/day]

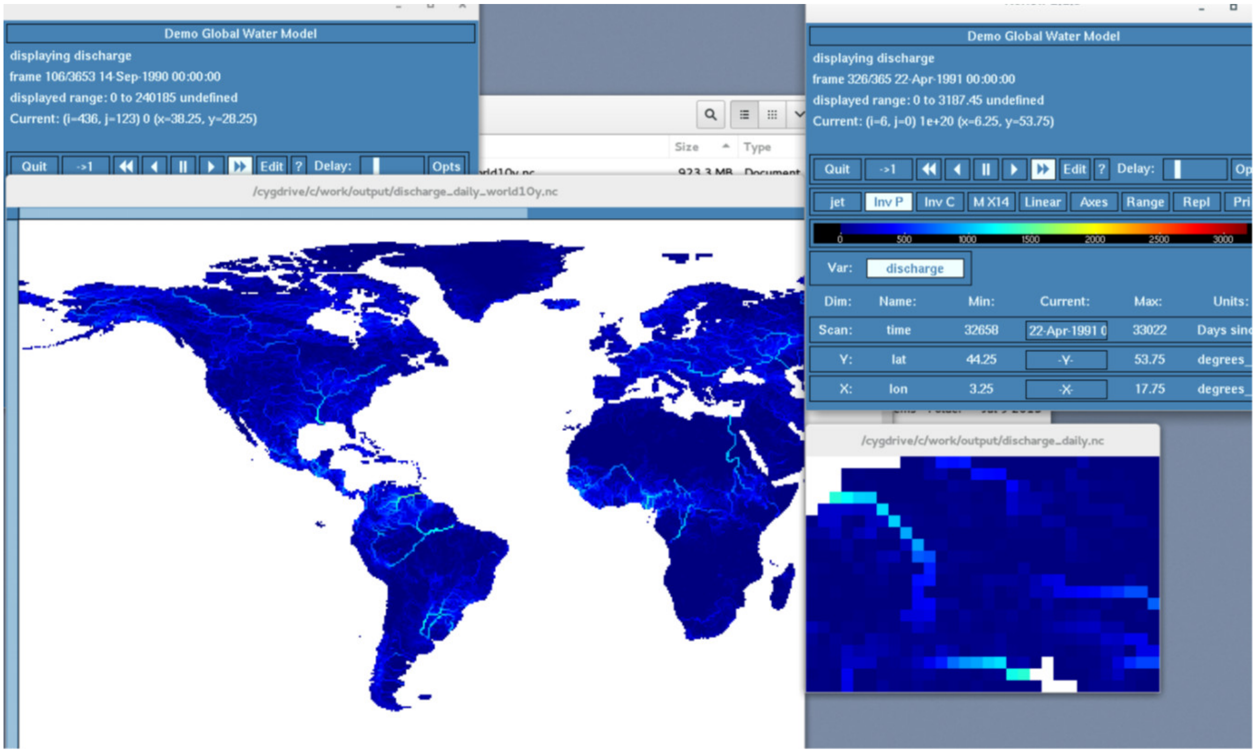
One year run example

8.2.3 Global soil moisture [mm/mm]

One year run example

8.3 Demo 2 - NcView output

Global discharge as world map
Output from NcView



8.4 Demo 3 - NcView timeserie

Discharge as timeseries
Output from NcView

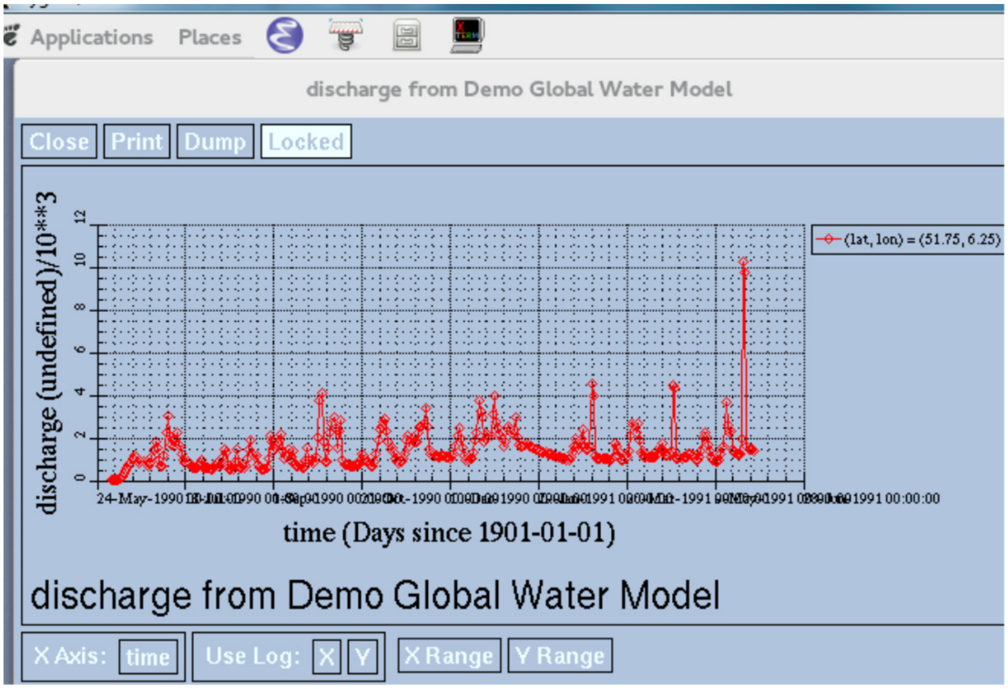
8.5 Demo 4 - Monthly timeserie

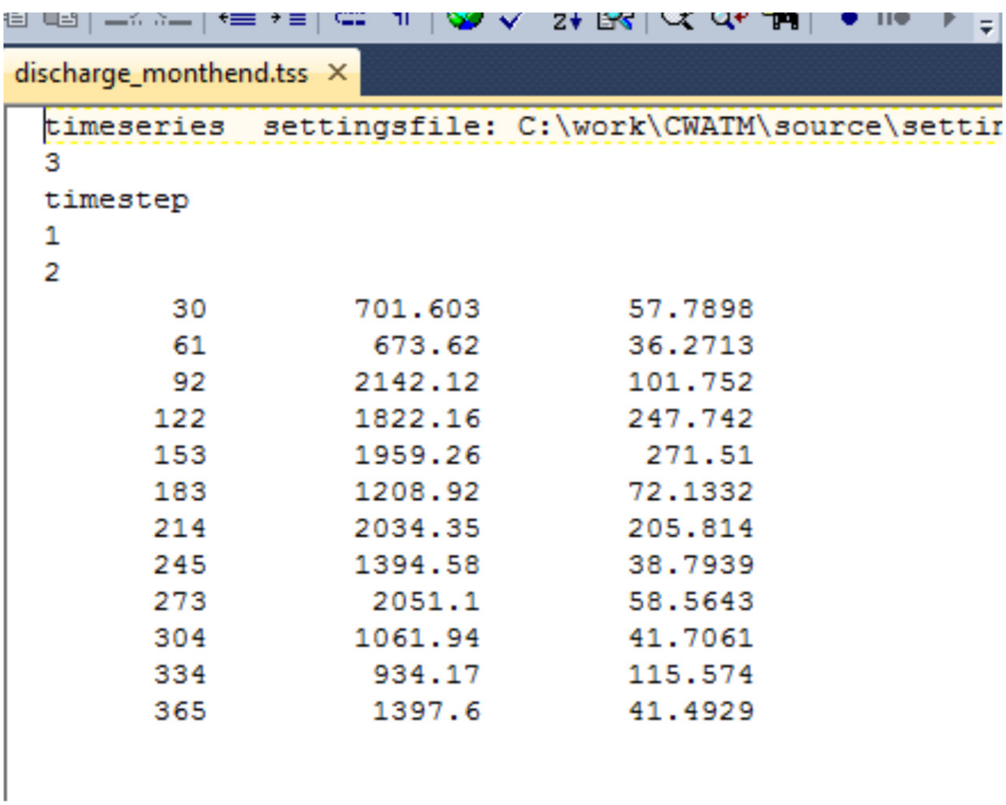
Discharge as monthly timeseries

8.6 Demo 5 - PCRaster Aguila output

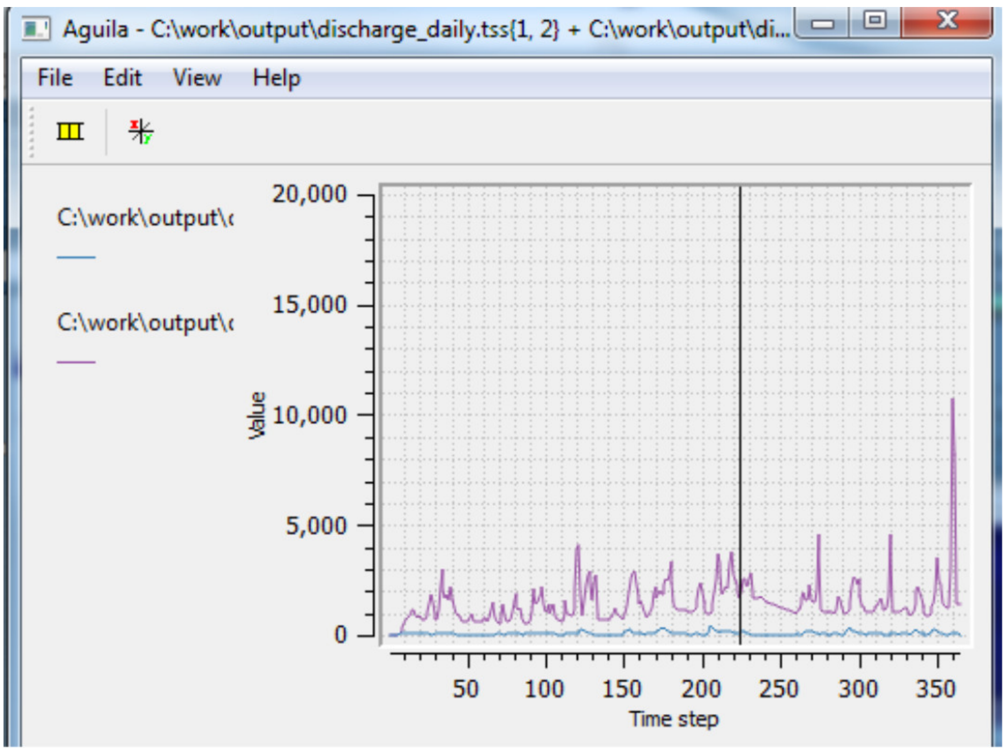
Discharge as timeseries Output from PCRaster Aquila[40]

[40] http://pcraster.geo.uu.nl/projects/developments/aguila/




discharge_monthend.tss

```
timeseries  settingsfile: C:\work\CWATM\source\settir
3
timestep
1
2
       30        701.603        57.7898
       61         673.62        36.2713
       92        2142.12        101.752
      122        1822.16        247.742
      153        1959.26         271.51
      183        1208.92        72.1332
      214        2034.35        205.814
      245        1394.58        38.7939
      273         2051.1        58.5643
      304        1061.94        41.7061
      334         934.17        115.574
      365         1397.6        41.4929
```


Aguila - C:\work\output\discharge_daily.tss{1, 2} + C:\work\output\di...
File
Edit
View
Help
C:\work\output\
C:\work\output\
20,000
15,000
10,000
5,000
0
Value
50
100
150
200
250
300
350
Time step

CHAPTER

NINE

THE MODEL ITSELF

Contents

9.1 Performance

Computational run time (on a linux single node - 2400 MHz with Intel Xeon CPU E5- 2699A v4):

Daily timestep on 0.5 deg

Global: 100 years in appr. 12h = 7.2min per year

	Process	sum % runtime
1	Read Meteo Data	6.2
2	Et pot	7.6
3	Snow	8.8
4	Soil	59.4
5	Groundwater	59.5
6	Runoff conc	70.1
7	Lakes	70.4
8	Routing	95.5
9	Output	100

For the global setting, soil processes with 50% computing time is the most time consuming part, followed by routing with 25% and runoff concentration with 10%.

Rhine: 640 years in appr. 4.5h = 0.4min per year

	Process	sum % runtime
1	Read Meteo Data	79.4
2	Et pot	80.5
3	Snow	80.9
4	Soil	88.8
5	Groundwater	88.9
6	Runoff conc	89.6
7	Lakes	89.8
8	Routing	99.6
9	Output	100

For the Rhine basin reading input maps 79% is by far the most time consuming process (not in absolute time but compared to the other processes), followed by routing (kinematic wave) 10% and the soil processes (8%).

9.2 Updates

Note:

Update history taken from github log

git log —pretty=format:"%ad - %an : %s" —date=short —graph > github.log

Most recent updates on top

```
* 2020-10-08 - CWatM : chg: Error handling improved, included numbering of error␣
↪handling add: pytest checks error handling
* 2020-10-02 - CWatM : add: Co2 data, check if climate data are upside down
* 2020-08-17 - CWatM : chk: added water withdrawal from neigbor cells
* 2020-07-02 - CWatM : Fix: some minor fixes to adjust the waterbalance, mainly water␣
↪demand e.g. calculation of return flow, lost to evaporation, dealing with fossil gw␣
↪and return flow of fossil gw
* 2020-06-15 - CWatM : Add: added self.var.tws total water storage, dis_outlet, sum_
↪soil, lakeReservoirStorage as variable Add: dis_outlet as discharge only at the␣
↪outlet points
* 2020-06-15 - CWatM : Add: added self.var.tws total water storage, dis_outlet, sum_
↪soil, lakeReservoirStorage as variable Add: dis_outlet as discharge only at the␣
↪outlet points
* 2020-06-03 - CWatM : chg: renamed water_demand/environmental_flow to water_demand/
↪environmental_need chg: docu/sourcecode.rst changed the graphic to display the␣
↪modules
* 2020-06-03 - CWatM : CHG: improved description for each class with defined global␣
↪Variables Add: water_demand.py in water_demand (moved from __init__) CHG: change␣
↪file encoding to uft-8 again
* 2020-06-02 - CWatM : CHG: added Luca's description of variables in each class␣
↪comment
* 2020-05-28 - Jens de Bruijn : Fix spelling error
| * 2020-05-28 - Jens de Bruijn : split water demand module
| * 2020-05-28 - Jens de Bruijn : some small pep8 changes
| * 2020-05-28 - Jens de Bruijn : requirement pytest-report to pytest-html
| * 2020-05-28 - Jens de Bruijn : fix error with usage function
* | 2020-05-28 - Jens de Bruijn : Fix decoding error
|/
* | 2020-05-28 - CWatM : CHG: cleaning cwatm_initial.py - put parts in data-handling␣
↪loadsetclone
```

(continues on next page)

(continued from previous page)

```
* | 2020-05-28 - CWatM : Chg: put the part of checking meteorological forcing data to␣
↪fit with mask map in readmeteo.py
* |   2020-05-28 - CWatM : CHG: Merge Jens changed self.var structure
|\ \
| * \   2020-05-28 - Peter Burek : Merge pull request #13 from iiasa/var-restructure
| |\ \
| | * \   2020-05-28 - Peter Burek : Merge branch 'develop' into var-restructure
| | |\ \
| | |/ /
| |/| |
| | * | 2020-05-19 - Jens de Bruijn : small bugfix
| | * | 2020-05-19 - Jens de Bruijn : revert some unnecessary changes
| | * | 2020-05-19 - Jens de Bruijn : return firstout from var
| | * |   2020-05-18 - Jens de Bruijn : Merge pull request #12 from iiasa/var-
↪variable-restructuring
| | |\ \
| | | * | 2020-04-15 - Jens de Bruijn : all variables to model.var
* | | | | 2020-05-28 - CWatM : Chk: preparation to merge with Jens change self.var␣
↪structure
|/ / / /
* | | |   2020-05-28 - Peter Burek : Merge pull request #11 from iiasa/simplify-run_
↪cwatm.py
|\ \ \ \
| * | | | 2020-04-17 - Jens de Bruijn : removed unneccesary import
| * | | | 2020-04-17 - Jens de Bruijn : simplify run_cwatm
* | | | | 2020-05-25 - CWatM : add: test self.var description in soil
* | | | |   2020-05-25 - CWatM : Merge branch 'develop' of https://github.com/iiasa/
↪CWATM_priv into develop
|\ \ \ \ \
| | |/ / /
| |/| | |
| * | | | 2020-05-18 - Jens de Bruijn : include automatically generated settingsfiles␣
↪and wordfiles temp files
* | | | | 2020-05-25 - CWatM : add: addition self.var description to soil as test
|/ / / /
* | | | 2020-05-15 - CWatM : add: added different option for ETP 2: Milly and Dunne␣
↪method 3: Yang et al. Penman Montheith correction method
| | | * 2020-04-27 - Mikhail Smilovic : Updates
| | | * 2020-04-27 - Mikhail Smilovic : Initial values and new pumping variable
| | | * 2020-04-27 - Mikhail Misha Smilovic : Fixed usingAllocSegments
| | | * 2020-04-27 - Mikhail Smilovic : Initialize variables and fix rootFrac
| | | *   2020-04-27 - Mikhail Smilovic : Merge branch 'develop' into Mikhail
| | | |\
| |_|_|/
|/| | |
* | | |   2020-04-20 - CWatM : Merge branch 'develop' of https://github.com/iiasa/
↪CWATM_priv into develop
|\ \ \ \
| * | | | 2020-04-20 - Mikhail Misha Smilovic : Corrected self.var.sumlakeResOutflow␣
↪and removed "somtimes_closed" feature
| |/ / /
* | | | 2020-04-20 - CWatM : new: put documentation for pytesting in pytesting new:␣
↪put documentation for docu in docu chg: changed pytesting fix: fixed find closest␣
↪option if option is misspelled
|/ / /
* | | 2020-04-16 - CWatM : chg: added some lines on docu/setup.rst
* | |   2020-04-16 - CWatM : Making a new cwat version which can be installed by pip␣
↪and manual
```

(continues on next page)

(continued from previous page)

```
|\ \ \
| * | | 2020-04-16 - Peter Burek : Delete .travis.yml
| * | | 2020-04-16 - Jens de Bruijn : make run_cwatm.py work both directly and as
↪'cwatm' from command line.
| * | |   2020-04-16 - Peter Burek : Merge pull request #10 from iiasa/jens
| |\ \ \
| | * \ \   2020-04-15 - Jens de Bruijn : Merge branch 'jens' of https://github.com/
↪iiasa/CWATM_priv into jens
| | |\ \ \
| | | * | | 2020-04-15 - Jens de Bruijn : include license (was already present in␣
↪docs) + include readme as long_description
| | * | | | 2020-04-15 - Jens de Bruijn : include license (was already present in␣
↪master branch and docs) + include readme as long_description
| | |/ / /
| | * | |   2020-04-15 - Jens de Bruijn : Merge branch 'jens' of https://github.com/
↪iiasa/CWATM_priv into jens
| | |\ \ \
| | | * | | 2020-04-15 - Jens de Bruijn : fix circular reference
| | * | | | 2020-04-15 - Jens de Bruijn : moved to cwatm folder
| | * | | | 2020-04-15 - Jens de Bruijn : remove nonexisting page
| | * | | | 2020-04-15 - Jens de Bruijn : update documentation to reflect running␣
↪from command line
| | * | | | 2020-04-15 - Jens de Bruijn : make cwatm runnable from the command line
| | * | | | 2020-04-15 - Jens de Bruijn : fix circular reference
| | |/ / /
| | * | | 2020-04-15 - Jens de Bruijn : fix pip installation + restructure document
| | * | | 2020-04-15 - Jens de Bruijn : create requirements.txt for documentation
| | * | | 2020-04-15 - Jens de Bruijn : change \over to \frac
| | * | | 2020-04-15 - Jens de Bruijn : fix logo path
| | * | | 2020-04-15 - Jens de Bruijn : included packages for testing in setup.py
| | |/ /
| * | | 2020-04-16 - Jens de Bruijn : remove .vscode folder
| |/ /
* | | 2020-04-16 - CWatM : cwamt which runs under pip install and manual install
* | | 2020-04-16 - CWatM : Upadating cwatm to work from pip install and manual install
|/ /
* | 2020-04-14 - CWatM : chk: trying to use Travis with pytest fix: some date␣
↪problems using 360 days
* |   2020-04-14 - CWatM : Merge branch 'develop' of https://github.com/iiasa/CWATM_
↪priv into develop
|\ \
| * \   2020-04-14 - Jens de Bruijn : Merge branch 'develop' of github.com:iiasa/
↪CWATM_priv into develop
| |\ \
| * | | 2020-04-14 - Jens de Bruijn : removed some unneccesary code + some renaming
* | | | 2020-04-14 - CWatM : chk: small changes to run with Travis
| |/ /
|/| |
* | | 2020-04-14 - CWatM : fix: 2nd fix for monthly netcdf file base on different␣
↪netcdf calendars fix: included metanetcdf.xml in cwatm folder chk: in case there is␣
↪not metanetcdf at the location defined in settingsfile -> look into cwatm folder␣
↪fix; include metaxml into setu
* | | 2020-04-14 - CWatM : fix: fixed a bug in the new meteo data use with 360 days
| | * 2020-04-14 - Mikhail Smilovic : Removed file related to leakage
| | * 2020-04-14 - Mikhail Misha Smilovic : Increase MODFLOW soil layer by␣
↪soildepth[0]
| | * 2020-04-14 - Mikhail Misha Smilovic : Reset summed up groundwater pumping
```

(continues on next page)

(continued from previous page)

```
| | * 2020-04-14 - Mikhail Misha Smilovic : Improved demand2pumping option
| | * 2020-04-14 - Mikhail Smilovic : Allocation segments and cleaning
| | * 2020-04-14 - Mikhail Misha Smilovic : Remove FUSE landcover commands and clean␣
↪up rootFrac
| | * 2020-04-10 - Mikhail Misha Smilovic : act_gw bug fix
| | * 2020-04-10 - Mikhail Misha Smilovic : Soil depth fix and Aquifer begins below␣
↪soil layer
| | * 2020-04-10 - Mikhail Misha Smilovic : Channel abstractions fix
| |/
|/|
* | 2020-04-10 - CWatM : chg: reads meteo data with different netcdf calendar and␣
↪unit (days, minutes)
* | 2020-04-10 - CWatM : Chg: tested last version for global 30, 5, rhine 5.30, Upper␣
↪Bhima Add: CWatM can use different calendar as meteo input e.g 360 days Chg:␣
↪improved setting mask in global dataset and meteoset Todo: meteo datasets should␣
↪have days from , make this flexible to minutes, ...
* |   2020-04-09 - CWatM : Merge branch 'develop' of https://github.com/iiasa/CWATM_
↪priv into develop
|\ \
| |/
| * 2020-02-20 - Jens de Bruijn : removed unused function parameter
| * 2020-02-19 - Jens de Bruijn : improve setup authors + include myself as an author
| * 2020-02-19 - Jens de Bruijn : Making CWatM installable as a pip package
| * 2020-02-19 - Jens de Bruijn : removed source for python 2
| * 2020-02-19 - Jens de Bruijn : fixes to documentation
| * 2020-02-19 - Jens de Bruijn : fix docstring
| * 2020-02-18 - Jens de Bruijn : Making CWatM installable as a pip package
* | 2020-04-09 - CWatM : Chg: Using meteo datasets with 360 days, no_leap etc.␣
↪automatically chg: calculting the position of the area map inside meteomaps, global␣
↪data sets
|/
*   2020-02-12 - CWatM : New: pytest framwork to test features of CWATM in different␣
↪environemnts (scales, basins, options) Chg: Changed cwatm3.py and globals.py to run␣
↪with pytest
|\
| * 2020-02-07 - Jens de Bruijn : make gitignore more general, works for all Python␣
↪versions now
| * 2020-02-07 - Jens de Bruijn : init
| * 2020-02-07 - Jens de Bruijn : changed time.clock() to time.perf_counter() as time.
↪clock() ensuring Python3.8 support. time.perf_counter was added in Python3.3
* | 2020-02-12 - CWatM : New: pytest framwork to test features of CWATM in different␣
↪environemnts (scales, basins, options) Chg: Changed cwatm3.py and globals.py to run␣
↪with pytest
|/
* 2020-02-06 - CWatM : Bugfix: corrected a bug that gave some error message when␣
↪using CWatM for 5min version Chg: Changed some internal structure to make it run␣
↪with pytest.ini (cwatm3.py, output.py, globals.py, datahandling.py, etc.) New: A␣
↪version which can be tested with a pytest framework
*   2019-12-05 - CWatM : Merge branch 'develop' of https://github.com/iiasa/CWATM_
↪priv into develop
|\
| * 2019-11-26 - Mikhail Misha Smilovic : Update of module for Segments and unmet_div_
↪ww
| * 2019-11-26 - Mikhail Misha Smilovic : Fix to update act_irrConsumption for not␣
↪LimitAbstraction
| *   2019-11-04 - Mikhail Misha Smilovic : Merge pull request #7 from␣
↪mikhailsmilovic/develop
```

(continues on next page)

(continued from previous page)

```
| |\
| | * 2019-11-04 - Mikhail Smilovic : Fix: all valid ldd values included
| | * 2019-11-04 - Mikhail Smilovic : Revert "Fix: all valid ldd values included"
| | * 2019-11-04 - Mikhail Smilovic : Fix: all valid ldd values included
| | *   2019-11-04 - Mikhail Misha Smilovic : Merge pull request #7 from iiasa/develop
| | |\
| | |/
| |/|
| * | 2019-11-04 - Mikhail Misha Smilovic : fix, added colon
| | *   2019-11-04 - Mikhail Misha Smilovic : Merge pull request #6 from iiasa/develop
| | |\
| | |/
| |/|
| * | 2019-11-04 - Mikhail Misha Smilovic : Clean up: remove print('hello')
| * |   2019-11-04 - Mikhail Misha Smilovic : Merge pull request #6 from␣
↪mikhailsmilovic/develop
| |\ \
| | |/
| | * 2019-11-04 - Mikhail Smilovic : Moving options into Settings file
| | *   2019-11-04 - Mikhail Misha Smilovic : Merge pull request #5 from iiasa/develop
| | |\
| | |/
| |/|
| * | 2019-11-04 - Mikhail Misha Smilovic : True --> 'True'
| * | 2019-11-04 - Mikhail Misha Smilovic : clean up of "sometimes_closed" feature
| | *   2019-10-31 - Mikhail Misha Smilovic : Merge pull request #4 from iiasa/develop
| | |\
| | |/
| |/|
| * |   2019-10-31 - Mikhail Smilovic : Merge branch 'develop' into pr/4
| |\ \
| | |/
| | * 2019-10-31 - Mikhail Smilovic : fixed negative pumping and pyc git ignore
| | * 2019-10-31 - Mikhail Smilovic : Activates pumping through modflow to meet gw␣
↪demand
| | * 2019-10-31 - Mikhail Smilovic : delete pyc files
| | * 2019-10-31 - Mikhail Smilovic : Removing pyc files and including Sarati␣
↪settings file
| | * 2019-10-30 - Mikhail Smilovic : rootFraction disabled
| | * 2019-10-30 - Mikhail Smilovic : Revert "Revert "Revert "Revert "Beginning␣
↪demand2pumping feature""""
| | * 2019-10-30 - Mikhail Smilovic : Revert "Revert "Revert "Beginning␣
↪demand2pumping feature"""
| | * 2019-10-30 - Mikhail Smilovic : Revert "Revert "Beginning demand2pumping feature
↪""
| | * 2019-10-30 - Mikhail Smilovic : Revert "Beginning demand2pumping feature"
| | * 2019-10-30 - Mikhail Smilovic : Beginning demand2pumping feature
| | * 2019-10-30 - Mikhail Misha Smilovic : Include sometimes_closed option
| | * 2019-10-30 - Mikhail Misha Smilovic : sometimes_closed option
| | * 2019-10-30 - Mikhail Misha Smilovic : Updated settings file
| | * 2019-10-30 - Mikhail Misha Smilovic : Test: small edits
* | | 2019-12-05 - CWatM : chg: adjusted downscaleing of meteo data. Now it checks if␣
↪the wordclim data fits to the map extend of the precipitation data
|/ /
* | 2019-10-19 - CWatM : Add: Point can be used as maskmap. this will generate the␣
↪upstream catchment Add: output as catchment sum or avg Add: doucu is updated
* | 2019-10-18 - CWatM : NEW: added additional time series output: tss as catchment␣
↪sum or average New: more checks to prevent white space in paths Fix: try to remove␣
↪all compiled code
```

(continues on next page)

(continued from previous page)

```
* |   2019-10-18 - CWatM : Merge branch 'develop' of https://github.com/iiasa/CWATM_
↪priv into develop
|\ \
| |/
| * 2019-09-25 - Mikhail Misha Smilovic : Update logo and CWATM --> CWatM
| * 2019-09-25 - Mikhail Misha Smilovic : New CWatM logo
| * 2019-09-23 - Mikhail Smilovic : Updates
| * 2019-09-13 - Mikhail Smilovic : Allows for the option 'usingAllocSegments'
* | 2019-10-18 - CWatM : NEW: added aditional time series output: tss as catchment␣
↪sum or average New: more checks to prevent whitespaces in pathes
|/
* 2019-09-11 - CWatM : fix: change reservoir size (from Mikhail)
* 2019-09-11 - CWatM : fix: data handling, using maskmap with col row celllenght lon␣
↪lat again
* 2019-08-06 - CWatM : Version 1.04
* 2019-08-06 - CWatM : Version 1.04
* 2019-06-27 - CWatM : fix: bug fix for initial value, small lakes was not stored
* 2019-06-03 - CWatM : New: Groundwater
* 2019-05-08 - CWatM : chk: changes omodflow groundwater_modflow.py chk data␣
↪handling: saving netcdf with modflow resolution
* 2019-05-03 - CWatM : fix: fixed some coding in groundwater
* 2019-04-30 - CWatM : chk: fixed cropcorrect load in evapopartionpot.py chk: library␣
↪for groundwater flopy only loaded when ModFlow is used
* 2019-04-30 - CWatM : Add: First version which include the ModFlow version from LG
* 2019-04-04 - CWatM : add: waterquality parts,: watertemp, waterlevel, travel time
* 2019-03-08 - CWatM : chk: improved some pics, intro viedo from Junko
*   2019-03-07 - CWatM : Merge branch 'develop' of https://github.com/iiasa/CWATM_
↪priv into develop
|\
| * 2019-03-05 - Mikhail Misha Smilovic : useSmallLakes commented out
* | 2019-03-07 - CWatM : chg: cwatm.py no predefined setting.file
|/
* 2019-03-05 - CWatM : changes for 1km version in readmeto and initcond
* 2019-01-24 - CWatM : Fix: removed double mult with soilddepth in waterdemand -␣
↪thank you Simon Moulds Fix: ldd with 1km reso was not working: changed kinematic,␣
↪and kinematic_sub and lake_reservoirs Fix: read_meteo was not reading meteo
* 2019-01-17 - CWatM : fix; cleaned snow_frost.html add: docu added to data.rst
* 2019-01-16 - CWatM : Add: added tesxt pics to data.rst in docu
*   2019-01-16 - CWatM : Merge branch 'develop' of https://github.com/iiasa/CWATM_
↪priv into develop
|\
| * 2019-01-16 - Mikhail Misha Smilovic : corrected under/overlines
| * 2019-01-15 - Mikhail Misha Smilovic : Added resolution
* | 2019-01-16 - CWatM : Add: some addition to docu: data.rst
|/
* 2019-01-14 - CWatM : test2 readme pic
* 2019-01-14 - CWatM : test1 pic readme
* 2019-01-14 - CWatM : chk: docu sourcecode and setup.doctree chk: checks
* 2019-01-14 - CWatM : Chk: change checking of data, mainly check.py and data␣
↪handling.py
* 2019-01-12 - CWatM : chk: changed parts in timestep.py so it does not need python-
↪dateutil library
* 2019-01-11 - CWatM : Chk: fix: add library python-dateutil as requirement in setup␣
↪and tutorial docu
* 2019-01-11 - CWatM : add: added file rhine_basin.zip as test case for cwatm.doctree␣
↪add: cwatmexe.zip include rhine_basin.zip - a whole package now: executable cwatm +␣
↪test case
```

(continues on next page)

(continued from previous page)

```
* 2019-01-11 - Mikhail Misha Smilovic : Updated authors' list and developers photo.
*   2019-01-10 - Mikhail Misha Smilovic : Merge pull request #2 from iiasa/mikhail
* 2018-06-05 - Mikhail Misha Smilovic : Update evaporation.py
* 2019-01-10 - Mikhail Misha Smilovic : Added an 'M' to CWAT
* 2019-01-10 - CWatM : Add: added a batch file to cwatmexe.zip
* 2019-01-08 - CWatM : Python 2.7 version support has stopped We maintaining from now␣
↪on Python 3.7 version
* 2019-01-08 - CWatM : Chg: Added header for autodocu to each function, class (all␣
↪changes only Python3.7 version) Chg: Put the downscaling functions in fron of␣
↪readmeteo: no longer internal functions of readmeteo Fix: Cleaned the code, removed␣
↪not used functions Chg: Improved the documentation, especially the autodocu of␣
↪source code
* 2019-01-05 - CWatM : fix: corrected some warnings from PCCharm code inspector
* 2019-01-04 - CWatM : add: adding executable cwatm.exe
* 2019-01-04 - CWatM : Fix: new water demand changes did not use the same variable␣
↪name act_surfacewater in waterdemand and routing_kinematic. changed this in both␣
↪version 2.7 and 3.7 Add python: added a report command to report data as .map or .
↪tif for debugging
* 2018-12-20 - CWatM : Python3.7 New: replaced pcraster framework by own framework␣
↪Removed folder pcraster2 New: added save conditions for warmstart -> you can add a␣
↪10d or 6m or 2y after the first date -> the initial data will be saved every 10d␣
↪(or whatever number), or 6 month or 2 year
* 2018-12-17 - CWatM : New: Python 3 test code
*   2018-12-17 - CWatM : Merge branch 'develop' of https://github.com/iiasa/CWATM_
↪priv into develop
|\
| *   2018-12-12 - Community Water Model : Merge pull request #3 from iiasa/
↪waterdemand_update
| |\
| | *   2018-12-12 - Community Water Model : Merge branch 'develop' into waterdemand_
↪update
| | |\
| | |/
| |/|
| | * 2018-08-15 - Unknown : modify irrConsuption to act_irrConsumption in␣
↪landcoverType and soil modules
| | * 2018-08-15 - Unknown : potential and actual values are explicitly written in␣
↪waterdemand module
| | * 2018-08-08 - Unknown : modified efficiency vaiables ;)
| | * 2018-08-07 - Unknown : modified read-netcdf for wateruse data
| | * 2018-07-24 - Yusuke : Added act_nonIrrConsumption conponents
| | * 2018-07-24 - Yusuke : Clean up before editing
* | | 2018-12-17 - CWatM : New: python 3 test version
* | | 2018-12-17 - CWatM : New: Added Python source code: Further test required, but␣
↪it seems to work. -> Plan in 2019 further development will use Python 3.7 coding␣
↪New: Building a executable .exe with Python 3  seems to work as well. Further␣
↪testing -> 2019 an installation setup will be produced using cx_freeze and Inno␣
↪setup to make an easy start on Windows (no Python background will be required for␣
↪CWATM users)
|/ /
* | 2018-12-12 - CWatM : Put Yusuke's version of waterdemand in (soil, landtypes,␣
↪waterdemand)
* | 2018-12-12 - CWatM : Fix: checkmap -c option now checks maps first (but can be␣
↪improved) new flag: usemeteodownscaling in [meteo] for using meteo downscaling Fix:␣
↪can now use rivernetwork as map or tif again (ldd.map) changes in initial and data_
↪handling
* | 2018-12-11 - CWatM : in sync with version on p drive
```

(continues on next page)

(continued from previous page)

```
* | 2018-12-11 - CWatM : Small change in tutorial, added output variable added␣
↪calibration tutorial, to be extended
* | 2018-09-24 - CWatM : chk: waterdemand can use water demand netcdf with m/s or␣
↪million m3 per month/year
* | 2018-08-07 - CWatM : fix: reading meteo map with no leap year (365 day maps) new:␣
↪using a cover map to put addition values in
|/
* 2018-07-09 - CWatM : Fix: waterbalance for soil Chg: output of tss from 3-d␣
↪variable e.g actualET[1]
* 2018-06-27 - CWatM : fix: corrected storing initial values for the next warm start␣
↪chk: changed environmental flow (EF)settings file -      loading EF is now in water␣
↪demand
* 2018-06-07 - CWatM : chg: outcommented a library call in data_handling    #from␣
↪netcdftime import utime chg: added the sum of ET_actual again
* 2018-05-17 - CWatM : Changed waterbalance Changed waterbodies in large and small␣
↪lakes and reservoirs
* 2018-04-24 - CWatM : Fix: bugfix to read waterdemand map
* 2018-04-19 - CWatM : Change: meteo data can be clipped before and used. CWAT␣
↪detects if it is a global map or a regional one e.g using only meteo data set for␣
↪the Rhine.
* 2018-04-16 - CWatM : Change; in waterdemand, landcovertyp and soil cjhange variable␣
↪names     Gross = demand = withdrawal, netto = consumptiom      all vraibales names␣
↪now are ..demand or .. consumption
* 2018-04-13 - CWatM : test
* 2018-04-13 - CWatM : Change: netcdf output as monthly or annual map has now a␣
↪adequate monthly or yearly time step e.g. Months since 1901-01-01
* 2018-04-03 - CWatM : Change: CWATM can be used with a smaller meteo dataset e.g. to␣
↪use a demo dataset for the Rine with pr, tavg, ETRef, EWref
* 2018-04-03 - CWatM : Change: CWAT can be used with a smaller meteo dataset  e.g. to␣
↪download a smaller test meteo dataset for the Rhine
* 2018-04-03 - CWatM : Chg: running cwatm with a smaller meteo dataset in order to␣
↪make a test catchment (e.g. Rhine) with a small meteo dataset
* 2018-03-20 - CWatM : Added: - small lakes - calc environmental flow - 5 arcmin␣
↪version - downscale 30min meteo dataset to 5min
* 2017-11-20 - CWatM : fix: replace strftime with .year or .month etc fix: looks for >
↪ 1e20 and -1e20 in each map and change these to standard zero value (default =0)
* 2017-10-30 - CWatM : Fix: bug fix to save maps with a SpinUp <> None
* 2017-10-27 - CWatM : Fix: reading meteo maps - every data > 1e12 is set to 0 Add:␣
↪maxtopwater in prg and settings.ini Fix: calibration routine
* 2017-09-21 - CWatM : bugfix: snow with more layers than 3
* 2017-09-20 - CWatM : chg: water demand, small lakes, land cover
* 2017-08-29 - CWatM : chg: water demand , soil add: error handling for output maps
* 2017-08-17 - CWatM : new: water demand is working chg: soil especially paddy and␣
↪non paddy irrigation bug: checked water balance
* 2017-07-13 - CWatM : fix: small bugfix, to run precipitation maps with the suffix .
↪nc4
* 2017-07-13 - CWatM : chg: soil part - using different maps -> map folder has to be␣
↪updated! chg: meteo maps do not have to be merge before -> stack of maps can  be␣
↪used add: inflow to a catchment (still to work on)
* 2017-05-23 - CWatM : chk: saving of netcdf with fixed number of time and with fixed␣
↪chunk size -> less diskspace used chk: a few more error handlings added
* 2017-05-19 - CWatM : chk: Chaznged soil calculation to Arno scheme and Mualem - van␣
↪Genuchten equation new: put in a lot of checks for the settingsfile e.g. check True␣
↪and false (not mispelled like ture). Check timing, check output variables chk: a␣
↪lot more error messages are given out if something is wrong chk: output netcdf time␣
↪is calculate in advanced in order to reduce size of output netcdf -> data_handling␣
↪line 789 sets it to this value
```

(continues on next page)

(continued from previous page)

```
* 2017-05-10 - CWatM : chk: bugfix cropKC per land cover new: snow evaporation␣
↪included new: Calibration routine added
* 2017-04-20 - CWatM : fix: output to netcdf - in output and data_handling fix:␣
↪output as a time series without header with the option -h new: readme.md for github
* 2017-04-18 - CWatM : Transfer to new IIASA domain and making it private in branch␣
↪develop
* 2017-04-18 - CWatM : Transfer to new IIASA CWAT domain
* 2017-04-18 - CWatM : ready for transfer to iiasa
* 2017-04-13 - CWatM : data handling: faster read of meteo data
* 2017-04-06 - CWatM : soil - Copy (2).py- removed bug in calculation of soildepth -␣
↪change calc of arno beta
* 2017-04-06 - CWatM : Merge branch 'branch2' of https://github.com/CWatM/CWatM into␣
↪branch2
* 2017-04-06 - Community Water Model : Create LICENSE
* 2017-04-06 - CWatM : Updated soil, removed bug in calculating the soil depth␣
↪changed how arno beta is calculated
* 2017-02-03 - CWatM : - made CWATM run under cygwin   (for other linux version the␣
↪c++ code has to be compiled) - fixed reading maskmap from rectangle
* 2017-02-02 - CWatM : set realtive file path to c++ routine
* 2017-02-02 - CWatM : - New kinematic routing    - c++ routine include     TODO:␣
↪make it usable for linu/Unix - removed pcraster GIS commands - new output routine␣
↪for time series - Budyko output.html - corrected bug in snow modules - corrected␣
↪bug in init read/save module - WORKING on lakes/reservoirs TODO: bug in reading␣
↪maskmap from coordinates
* 2017-01-17 - CWatM : init condition - save more than 1 date
* 2017-01-16 - CWatM : Lake/reservoirs routing
* 2016-12-22 - CWatM : updated soil , initconditions etc
* 2016-12-16 - CWatM : runoff concentration
* 2016-12-08 - CWatM : With sphinx documentation making files
* 2016-12-07 - CWatM : Update
* 2016-12-07 - CWatM : Preferential flow, frost
* 2016-11-10 - CWatM : Cacluation Evaporation from climate data
* 2016-10-21 - CWatM : Changed soil + test
* 2016-10-18 - CWatM : Waterdemand included
* 2016-10-03 - CWatM : last August update - waterbalance
* 2016-08-26 - CWatM : water balance 7
* 2016-08-26 - CWatM : water balance 6
* 2016-08-25 - CWatM : water Balance 5
* 2016-08-24 - CWatM : water balance 4 Checks ok : soil , groundwater, routing,␣
↪waterdemand Missing: reservoirs, sum up to catchments
* 2016-08-23 - CWatM : water balance 3
* 2016-08-23 - CWatM : water balance 2
* 2016-08-22 - CWatM : Water balance check 1 Output on screen
* 2016-08-19 - CWatM : initial condition
* 2016-08-17 - CWatM : Spin up
* 2016-08-17 - CWatM : output netcdf add attributes
* 2016-08-10 - CWatM : output + time
* 2016-08-10 - CWatM : date and time
* 2016-08-09 - CWatM : output 3
* 2016-08-09 - CWatM : output 2
* 2016-08-08 - CWatM : output timeseries
* 2016-08-03 - CWatM : waterbodies 1 Checked routing - working :)
* 2016-08-02 - CWatM : routing 3
* 2016-08-01 - CWatM : routing 2
* 2016-08-01 - CWatM : routing 1
* 2016-07-29 - CWatM : some changes I do not know anymore
* 2016-07-26 - CWatM : soil + groundwater
```

(continues on next page)

(continued from previous page)

```
* 2016-07-26 - CWatM : soil check3
* 2016-07-25 - CWatM : soil check2
* 2016-07-25 - CWatM : check soil module
* 2016-07-24 - CWatM : soil update
* 2016-07-24 - Burek : Soil and groundwater
* 2016-07-22 - CWatM : soil
* 2016-07-21 - CWatM : till waterdemand - soil
* 2016-07-20 - CWatM : Next step interception
* 2016-07-19 - CWatM : changing irrigationarea part
* 2016-07-15 - CWatM : Initial procedure for soil, groundwater, waterdemand
* 2016-07-13 - CWatM : include: snow frost
* 2016-04-02 - CWatM : Initial commit
```

CHAPTER

TEN

DATA

Contents

10.1 Data requirements

10.2 Data format

In general data format is netCDF (version3 or version4)

For the mask map (to define the area of calculation) or the stations (to define the time series outputs) in can be either netCDF, Geotiff or PCRaster maps

10.3 Data storage structure

```
project
├── README.txt
│
├── areamaps
│   └── maskmap, stationmap
│
├── landcover
│   ├──forest
│   │   ├── cropCoefficientForest_10days
│   │   ├── interceptcapForest10days
│   │   ├── maxRootdepth, minSoilDepthFrac
│   │   └── rootFraction1, rootFraction2
│   │
│   ├── grassland (same var as forest)
│   │
│   ├── irrNonPaddy (same var as forest)
│   │
│   └── irrPaddy (same var as forest)
│
├───landsurface
│   ├── fractionlandcover, global_clone
│   │
│   ├── albedo
│   │   └── albedo
│   │
│   ├── topo
│   │   └── dz_Rel_hydro1k, elvstd , tanslope
│   │
│   └── waterDemand
│       └── domesticWaterDemand, industryWaterDemand, irrigationArea, efficiency
│
├── soil
│   ├── alpha, forest_alpha, lamdba, forest_lambda, ksat, forest_ksat, thetas, forest_
↪thetas, thetar, forest_thetar
│   └── cropgrp
│
├── groundwater
│   └── kSatAquifer, recessionCoeff, specificYield
│
└── routing
    ├── ldd, catchment, cellarea
    │
    ├── kinematic
    │   └── chanbnkf, chanbw, changrad, chanleng, chanman
    │
    └── lakereservoirs
        ├── lakeResArea, lakeResDis,lakeResID, lakeResType, lakeResVolRes,␣
↪lakeResYear,
        └── smallLakesRes, smalllakesresArea, smalllakesresDis, smallwatershedarea
```

10.4 Static data

10.4.1 Mask map

- mask map or coordinates to model only regions or catchments (value in mask = 1)
- maps or coordinates for station to print time series

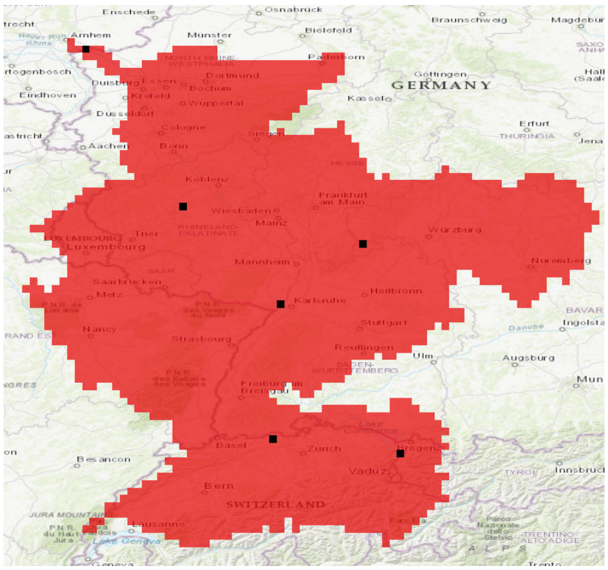


Figure 1: Mask map for the Rhine basin at 5' showing in addition 6 stations

> **Warning:** Make sure any cell defined in the mask map has a value (not NaN!) in the following map. A missing value in a cell will lead to a missing value in the result maps from the process this map is linked to.
>
> The routing process will carry this missing value downstream!

10.4.2 Landsurface

Digital elevation model and river channel network

The model uses a digital elevation model and its derivate (e.g. standards deviation, slope) as variables for the snow processes and for the routing of surface runoff. The Shuttle Radar Topography Mission - SRTM (Jarvis et al., 2008)[41] is used for latitudes <= 60 deg North and DEM Hydro1k (US Geological Survey Center for Earth Resources Observation and Science)[42] is used for latitudes > 60 deg North

[41] Jarvis, A., H. I. Reuter, A. Nelson and E. Guevara (2008). Hole-filled SRTM for the globe Version 4, available from the CGIAR-CSI SRTM 90m Database (http://srtm.csi.cgiar.org).

[42] US Geological Survey Center for Earth Resources Observation and Science Hydro1k. U. E. Land Processes Distributed Active Archive Center (LP DAAC), Sioux Falls, SD.

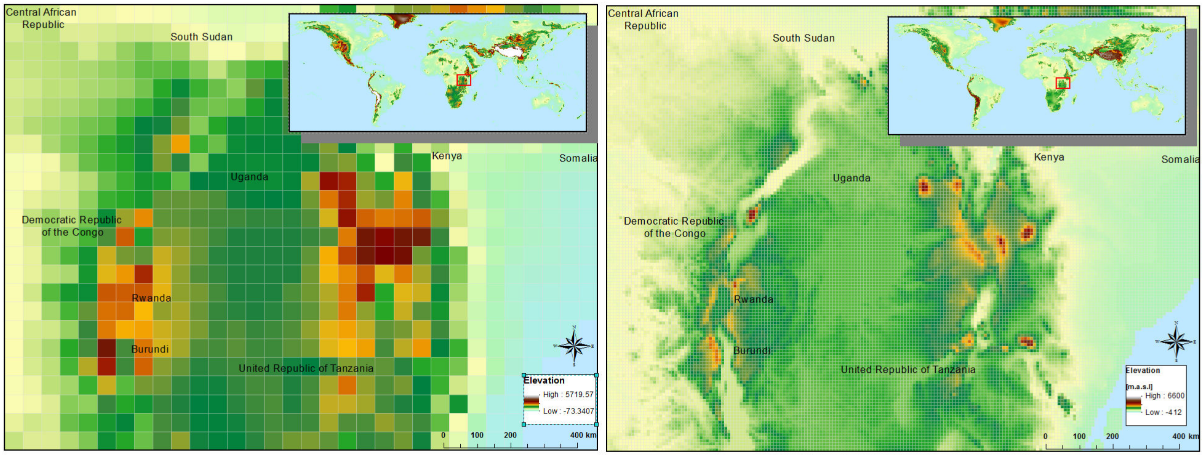


Figure 1: Digital elevation based on SRTM for 30’ and 5’

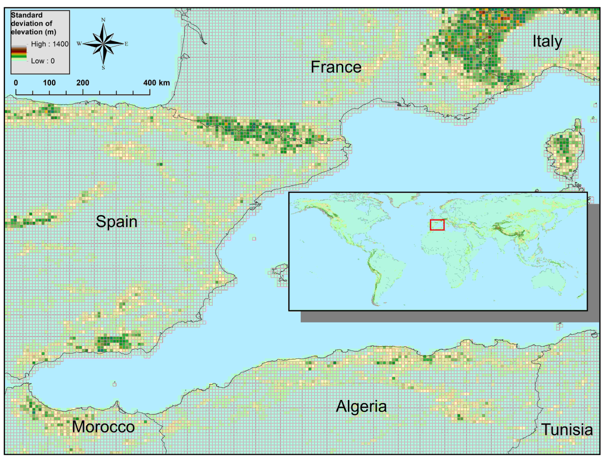


Figure 2: Standard deviation of elevation based on SRTM for 5’

10.4.3 River drainage maps

The river drainage map or local drain direction (LDD) is the essential component to connect the grid cells in order to express the flow direction from one cell to another and forming a river network from the springs to the mouth.

The approach to find the flow direction is in theory quite simple: There are eight valid output directions relating to the eight adjacent cells into which flow could travel. This approach is commonly referred to as an eight-direction (D8) flow model. The direction from each cell to its steepest downslope neighbour is chosen as flow direction. If the flow direction for each cell is given, a raster of accumulated flow into each cell can be calculated. Figure 4 shows the steps from DEM to flow direction to flow accumulation. Flow direction is shown in PC-Raster coding of the direction (ArcGIS uses another coding).

CWatM uses a local drainage direction map which defines the dominant flow direction in one of the eight neighboring grid cells (D8 flow model). This forms a river network from the springs to the mouth of a basin. To be compliant with the ISIMIP framework the 0.5° drainage direction map (DDM30) of (Döll and Lehner, 2002)[43] is used. For higher resolution e.g. 5’ different sources of river network maps are available e.g. HydroSheds (Lehner et al., 2008)[44] – DRT

[43] Döll, P. and B. Lehner (2002). “Validation of a new global 30-min drainage direction map.” Journal of Hydrology 258(1): 214-231.

[44] Lehner, B., K. Verdin and A. Jarvis (2008). “New global hydrography derived from spaceborne elevation data.” Eos 89(10): 93-94.

(Wu et al., 2011)[45] and CaMa-Flood (Yamazaki et al., 2009)[46]. These approaches uses the same hydrological sound digital elevation model but differ in the upscaling methods. Zhao et al. (2017)[47] shows the importance of routing schemes and river networks in peak discharge simulation. For CWatM the DDM30 is used for 0.5° and DRT is used for 5'.

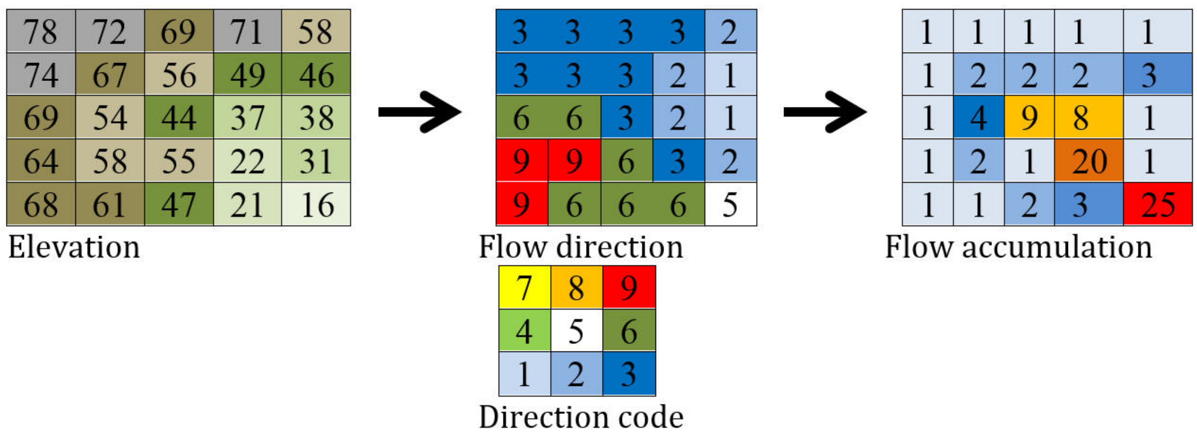


Figure 3: From elevation to flow accumulation

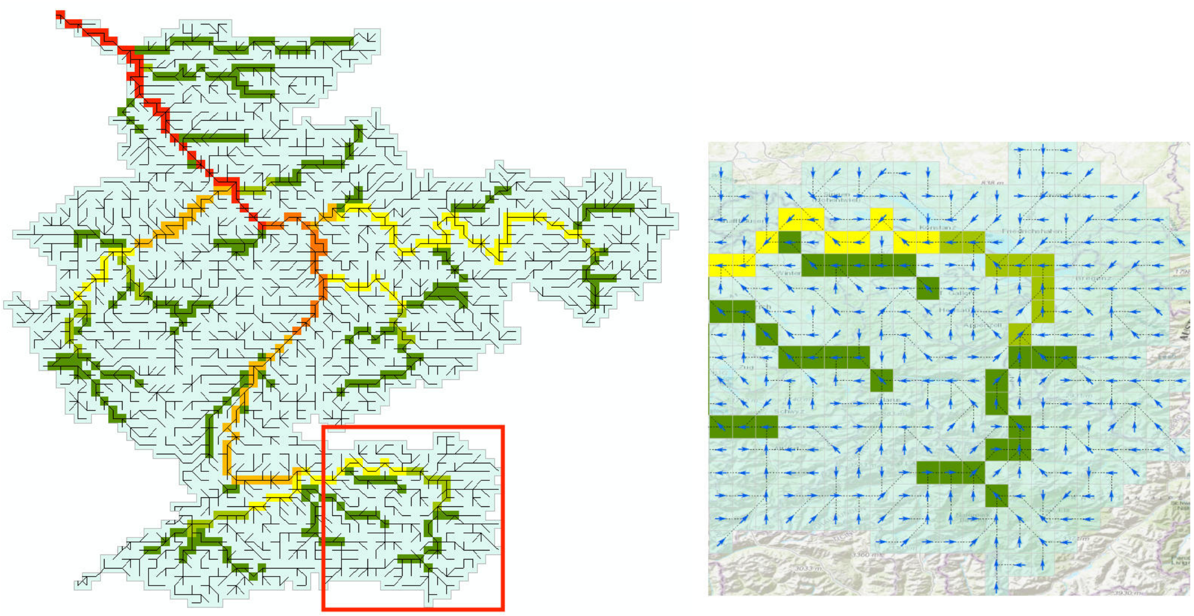


Figure 4: River network for the Rhine basin

10.4.4 River channel maps

Channel maps are describing the geometry like the length, slope, width and depth of the main channel inside a grid cell. Data used to get the geometry are mainly taken from elevation model and channel network.

[45] Wu, H., J. S. Kimball, N. Mantua and J. Stanford (2011). "Automated upscaling of river networks for macroscale hydrological modeling." Water Resources Research 47(3).

[46] Yamazaki, D., T. Oki and S. Kanae (2009). "Deriving a global river network map and its sub-grid topographic characteristics from a fine-resolution flow direction map." Hydrology and Earth System Sciences 13(11): 2241-2251.

[47] Zhao, F., Veldkamp, T. I. E., Frieler, K., Schewe, J., Ostberg, S., Willner, S., Schauberger, B., Gosling, S., N. , Müller Schmied, H., Portmann, F., T. , Leng, G., Huang, M., Liu, X., Tang, Q., Hanasaki, N., Biemans, H., Gerten, D., Satoh, Y., Pokhrel, Y., Stacke, T., Ciais, P., Chang, J., Ducharne, A., Guimberteau, M., Wada, Y., Kim, H., & Yamazaki, D. (2017). The critical role of the routing scheme in simulating peak river discharge in global hydrological models. Environmental Research Letters, 12(7), 075003

Methodology

Flow through the channel is simulated using the kinematic wave equations. The basic equations used are the equations of continuity and momentum. The continuity equation is:

$\frac{\delta Q}{\delta x} + \frac{\delta A}{\delta t} = q$

where:
Q: channel discharge [m3 s-1],
A: cross-sectional area of the flow [m2]
q: amount of lateral inflow per unit flow length [m2 s-1].

The momentum equation can also be expressed as (Chow et al., 1988):

$A = \alpha Q^{\beta}$

The coefficients and are calculated by putting in Manning's equation

$Q = Av = \frac{AR^{2/3}\sqrt{So}}{n} = \frac{A^{5/3}\sqrt{So}}{nP^{2/3}}$

where:
v: velocity [m/s]
n: Manning's roughness coefficient
P: wetted perimeter of a cross-section of the surface flow [m]
R: hydraulic Radius R=A/P

Solving this for and gives:

$\alpha = (\frac{nP^{2/3}}{\sqrt{So}})^{\beta}$ and $\beta = 0.6$

To calculate CWatM uses static maps of:
P: wetted perimeter approximated in CWatM: P = channel width + 2 * channel bankful depth
n: Manning's coefficient
S0: gradient (slope) of the water surface: S0 = elevation/channel length

Channel length

The network upscaling method of Wu et al. (2011)[45] is tracing the finer river network inside the coarser resolution. Channel length of 5' is traced from original SRTM channel length with the diagonal path taken to be 2 straight path.

Channel gradient

Channel gradient (or channel slope) is the average gradient of the main river inside a cell.

The approach taken here is to take the elevation from where the fine resolution channel enters the coarser grid cell and the elevation where it leaves the grid cell. Channel gradient is then calculated as:

Channel gradient = (elevation[in] –elevation[out]) / channel length.

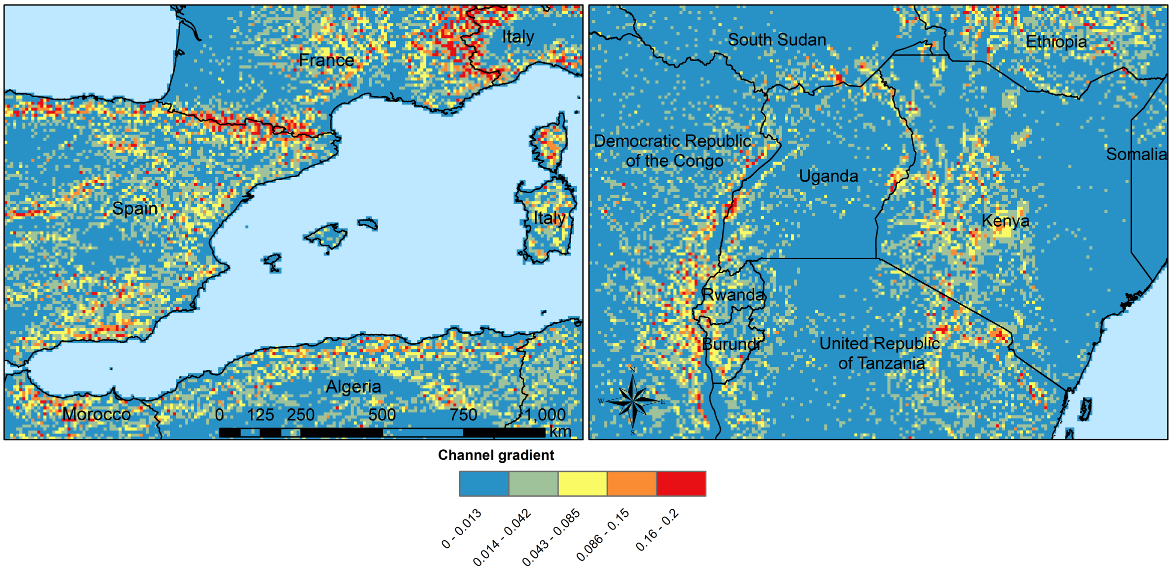


Figure x: Channel gradient at 5 in % or tan()'

Manning's roughness

Manning's roughness coefficient (n) is one of the calibration parameter in CWATM. But on subbasin level an estimation of the spatial distribution of n is needed. n normally range between 0.025 (low land rivers) and 0.075 (mountainous rivers with a lot of vegetation, gravels). A low n = smooth surface results in a faster travel time and higher peaks. A high n = rough surface results in slower travel time and lower peaks. Inspection of the riverbed will reveal characteristics related to roughness. A treatment of the use of Manning's coefficients is in McCuen (1998)[48]. Below is a first-approximation of Manning's coefficients for some widely observed beds:

```
n = 0.04 - 0.05          Mountain streams
n = 0.035                    Winding, weedy streams
n = 0.028 - 0.035        Major streams with widths > 30m at flood stage
n = 0.015                    Clean, earthen channels
```

For the base map of Manning a regression function is used with 0.025 as the minimum value for flatland rivers with large upstream areas. A maximum of 0.015 is added for flatland rivers and small upstream areas (upstream area dependent) and another maximum of 0.030 is added if in mountainous areas (elevation dependent):

```
Manning =0.025 + 0.015 * min(50/upstream,1) + 0.030*min(DEM/2000,1)
Where:
upstream: upstream catchment area [km]
DEM:           elevation from Digital elevation model [m]
```

48 McCuen, R. H. (1998). Hydrologic Analysis and Design. Upper Saddle River, NJ, USA: Prentice Hall.

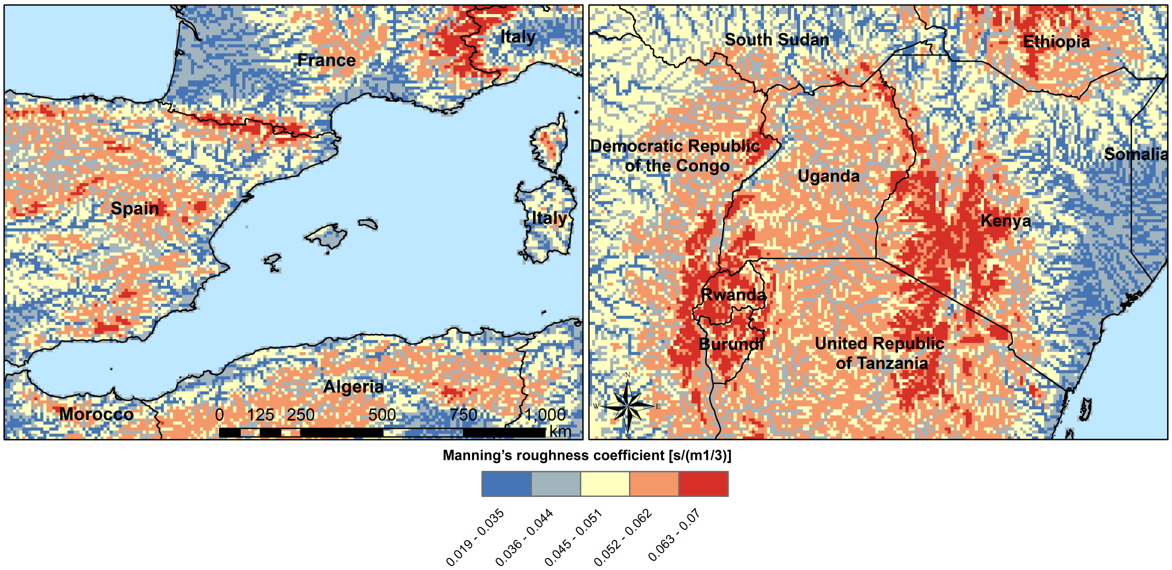


Figure x: Manning's roughness coefficient for 5'

Channel Bottom Width

The channel bottom width is calculated in two steps with the first step using a simply regression between channel width and upstream area and the second uses a better correlated one between average discharge and channel width. First the channel bottom width is calculated by a simply regression between upstream catchment area and width:

```
Channel width=upstreamArea ×0.0032
```

This first map is used to run CWatM to get an estimate on average discharge.

In the second step a regression formula from Pistocchi et al. 2006[49] is used to calculate the channel bottom width with average discharge as regressor, because discharge seems to be better correlated to width than upstream area. This is quite obvious if you look at small alpine catchment with high precipitation and therefore high discharge and on the other side at big, almost semiarid catchments on the Iberian peninsula with low average discharge:

```
Channel width=average Q ^ 0.539
```

[49] Pistocchi, A., & Pennington, D. (2006). European hydraulic geometries for continental SCALE environmental modelling. Journal of Hydrology, 329(3-4), 553-567

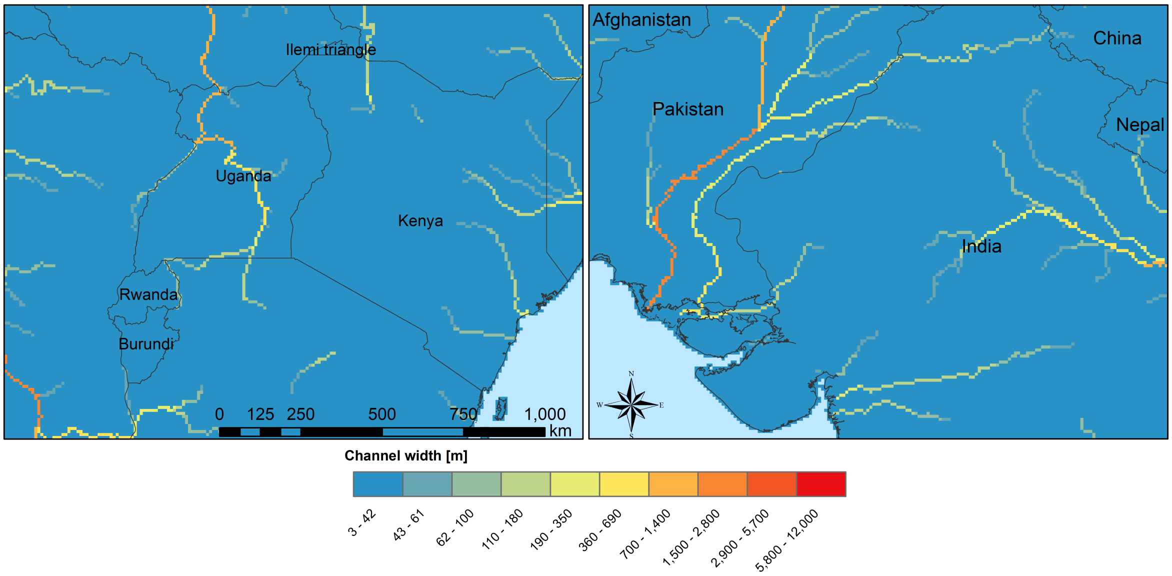


Figure 6: Channel width at 5'

Channel bankful depth

Instead of deriving channel hydraulic properties from a non linear correlation with the upstream area we are using the Manning's equation to get a better estimate. But for the first estimate (same as for channel bottom width) we use a correlation with upstream area:

```
Channel bankful depth = 0.27 upstreamArea^0.33
```

In the second step we use the Manning's equation. We adopt a rectangular cross section and we assume depth is small compared to width. So the perimeter is assumed to be:

```
P = 1.01 * channel bottom width
```

Discharge for bankful discharge is assumed to be two times the average discharge (Qavg)

$$Q = 2 * Qavg$$

$$Q = \frac{A^{5/3}\sqrt{So}}{nP^{2/3}} \approx \frac{Wh^{5/3}\sqrt{So}}{n(1.01W)^{2/3}}$$

Where:
W: Channel width
h: bankful depth
Q: bankful discharge ~ 2 * average discharge

As we now know all the other variables we can solve this equation for bankful depth with some assumption:

This leads to the equation:

$$Channelbankfuldepth(h) = 1.004N^{3/5}Q^{3/5}W^{-3/5}So^{-3/10}$$

Where:

W: Channel width
Q: bankful discharge ~ 2 * average discharge

10.4.5 Soil and soil hydraulic properties

Modeling of unsaturated flow and transport processes can be done with the 1D Richard equation, which requires a high spatial and temporal distribution of the soil hydraulic properties

$\frac{\delta\Theta}{\delta t} = \frac{\delta}{\delta z}[K(\Theta(\frac{\delta h(\Theta)}{\delta z} - 1)] - S(\Theta)$ (1D Richard equation)

Where:
: soil volumetric moisture content [L3/L3]
t: time [T]
h: soil water pressure head [L]
K(): unsaturated hydraulic conductivity [L/T]
z: vertical coordinate
S: source sink term [T-1]

With the simplification the 1D Richard equation e.g. flow of soil moisture is entirely gravity-driven and matrix potential gradient is zero this implies a flow that is always in downward direction at a rate that equals the conductivity of the soil. The relationship can now be described with the model of Mualem (1976)[50] and with the van Genuchten model (1980)[51] equation. Please find a full description of the soil process modeling in Burek et al. 2020: https: //doi.org/10.5194/gmd-13-3267-2020

$K(\Theta) = K_s(\frac{\Theta-\Theta_r}{\Theta_s-\Theta_r})^{0.5}\{1-[1-(\frac{\Theta-\Theta_r}{\Theta_s-\Theta_r})^{1/m}]^m\}^2$ (Van Genuchten equation)

Where:
Ks: saturated conductivity of the soil [cm/d-1]
K(): unsaturated conductivity
Θ Θ_s Θ_r : actual, maximum and residual amounts of moisture in the soil [mm]
m: is calculated from the pore-size index λ : $m = \frac{\lambda}{\lambda+1}$

The soil hydraulic parameter Θ_s Θ_r λ and K_s are needed to simulated soil water transport for the van Genuchten model.

The infiltration capacity of the soil is using the Xinanjiang (also known as VIC/ARNO) model (Todini, 1996)[52]
The soil hydraulic parameter α (inverse of air entry suction) is needed for calculating infiltration capacity

[50] Mualem, Y. (1976). A New Model for Predicting the Hydraulic Conductivity of Unsaturated Porous Medial. Water Resources Research, Vol. 12, 513-522

[51] Van Genuchten, M. T. (1980). A Closed Form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils. Soil Science Society of America Journal

[52] Todini, E. (1996). The ARNO rainfall—runoff model. Journal of Hydrology, 175(1), 339-382

Harmonized World Soil Database

The Harmonized World Soil Database 1.2 (HWSD) FAO et al. (2012)[53] - Version 1.2 7 March, 2012 was developed by the Land Use Change and Agriculture Program of IIASA (LUC) and the Food and Agriculture Organization of the United Nations (FAO). The HWSD is a 30 arc-second raster database with over 16000 different soil mapping units that combines existing regional and national updates of soil information worldwide – the European Soil Database (ESDB), the 1:1 million soil map of China, various regional SOTER databases (SOTWIS Database), and the Soil Map of the World – with the information contained within the 1:5000000 scale FAO-UNESCO Soil Map of the World. The resulting raster database is linked to harmonized soil property data.

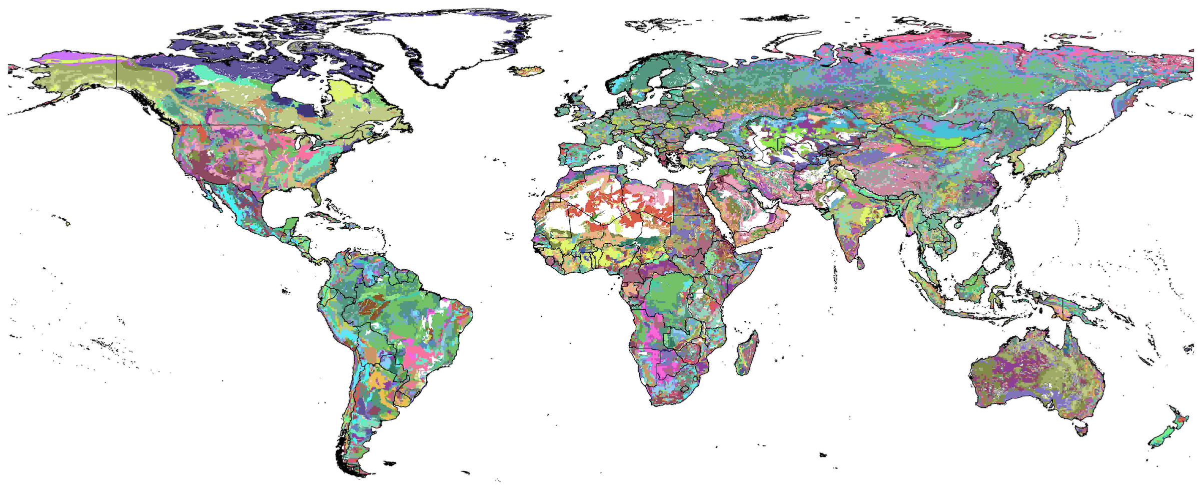


Figure 7: Harmonized World Soil Database Index, FAO et al. (2012)

From the HWSD the standard soil properties like texture, porosity, soil minerals (% of sand, clay), organic mater and bulk density are used. For example Bulk density second soil layer 5-30 cm depth:

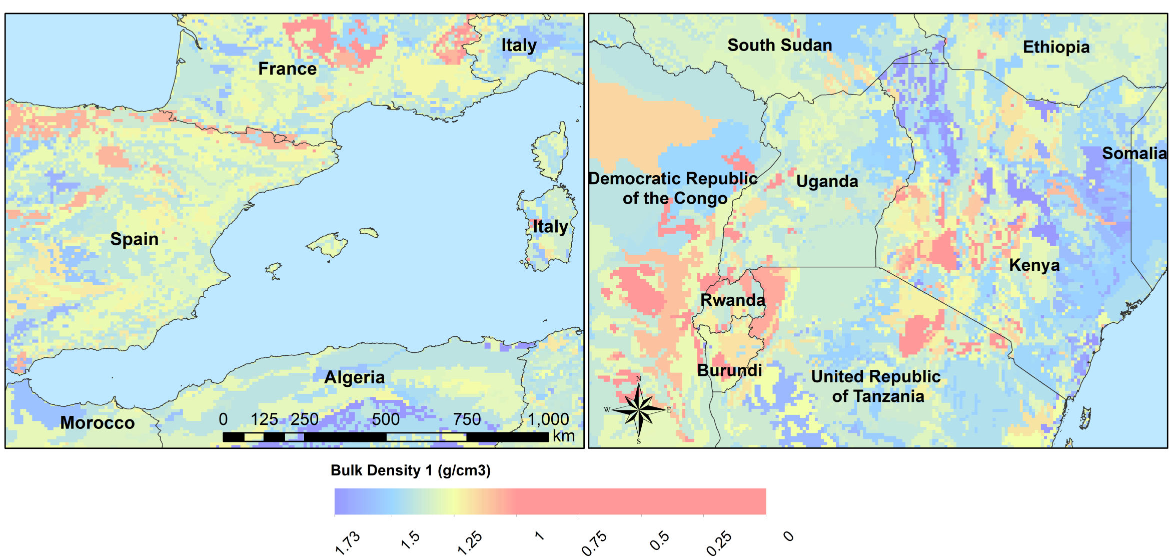


Figure 8: Bulk density second soil layer 5-30 cm at 5’

[53] FAO, IIASA, ISRIC, ISSCAS, & JRC. (2012). Harmonized World Soil Database (version 1.2). http://www.fao.org/soils-portal/soil-survey/soil-maps-and-databases/harmonized-world-soil-database-v12/en/

Pedotransfer function Rosetta3

Soil parameters required by CWatM are obtained from soil properties by using a pedotransfer function.

A pedotransfer is used from Zhang and Schaap 2016[54] to transfer the standard soil properties (soil texture, porosity, organic mater and bulk density) to the van Genuchten model parameters: Θ_s (maximal amount of moisture) Θ_r (residual amount of moisture) λ (pore-size index) K_s (saturated conductivity of the soil) and α (inverse of air entry suction)

Rosetta3 code is available at: http://www.cals.arizona.edu/research/rosettav3.html

For example s and Ks:

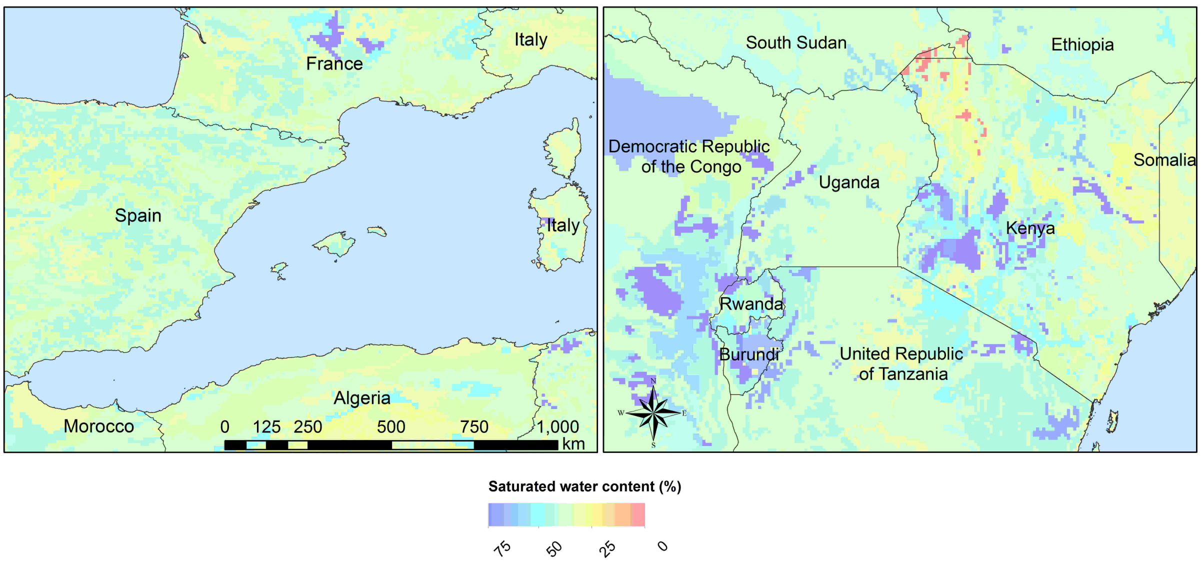


Figure 9: Soil volumetric moisture content (s) [%] second soil layer 5-30 cm at 5'

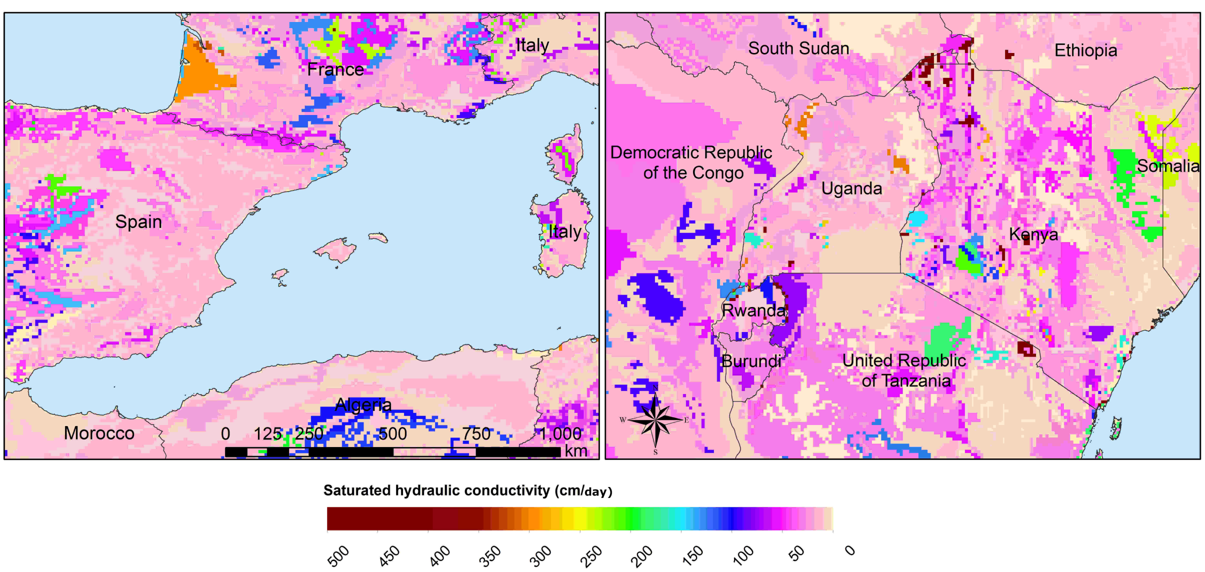


Figure 10: Saturated hydraulic conductivity (Ks) [cm/day] second soil layer 5-30 cm at 5'

[54] Zhang, Y., Schaap, M.,(2017): Weighted recalibration of the Rosetta pedotransfer model with improved estimates of hydraulic parameter distributions and summary statistics (Rosetta3),Journal of Hydrology,Volume 547,Pages 39-53,ISSN 0022-1694,https://doi.org/10.1016/j.jhydrol.2017.01.004. (http://www.sciencedirect.com/science/article/pii/S0022169417300057)

10.4.6 Groundwater

For groundwater modeling maps of the recession constant of the hydraulic conductivity and the storage coefficient are needed. Gleeson et al., (2011)[55] and Gleeson et al. (2014)[56] can provide data for this.

Global RecessionConstant GLIM: [1/day] based on drainage theory (linear reservoir)
Global SatHydraulicConductivity: Mean permeability of consolidated and unconsolidated geologic units below the soil [log10 m2]
Global StorageCoefficient [m/m]: specific yields or storage coefficients

Data:
GLHYMPS—Global Hydrogeology Maps of permeability and porosity (Gleeson et al., 2014)
http://crustalpermeability.weebly.com/data-sources.html
http://spatial.cuahsi.org/gleesont01/

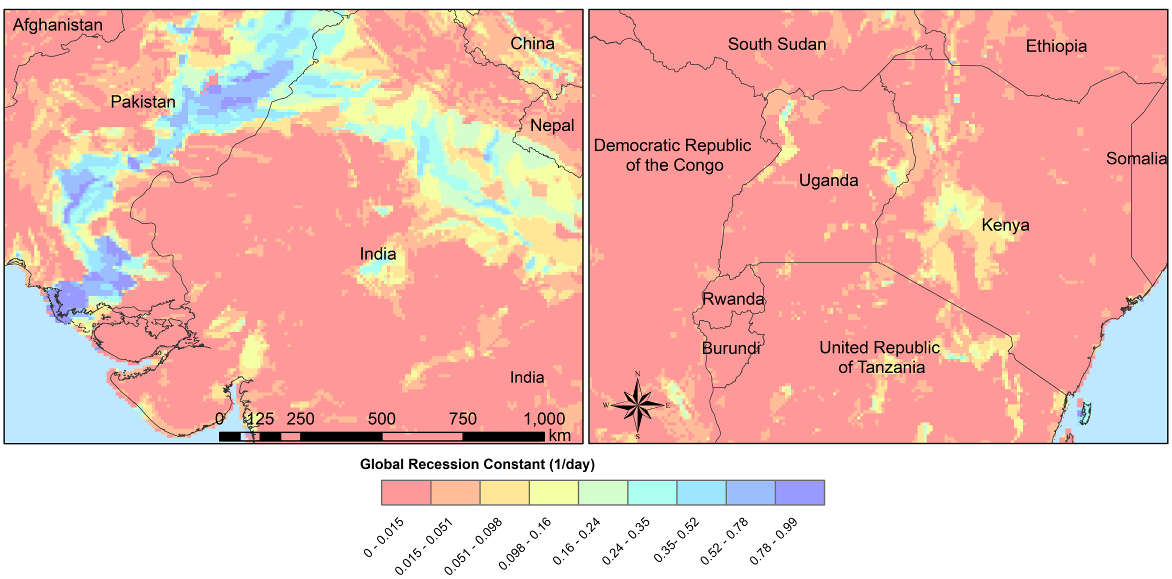


Figure 11: Recession constant GLIM: [1/day] at 5'

Lakes and Reservoirs

The HydroLakes database http://www.hydrosheds.org/page/hydrolakes (Lehner et al. (2011)[57]; Messager et al. (2016)[58], provides 1.4 million global lakes and reservoirs with a surface area of at least 10ha. CWatM differentiate between big lakes and reservoirs which are connected inside the river network and smaller lakes and reservoirs

[55] Gleeson, T., L. Smith, N. Moosdorf, J. Hartmann, H. H. Dürr, A. H. Manning, L. P. H. van Beek, and A. M. Jellinek (2011), Mapping permeability over the surface of the Earth, Geophys. Res. Lett., 38, L02401, doi:10.1029/2010GL045565.
[56] Gleeson, T., N. Moosdorf, J. Hartmann and L. P. H. Van Beek (2014). "A glimpse beneath earth's surface: GLobal HYdrogeology MaPS (GLHYMPS) of permeability and porosity." Geophysical Research Letters 41(11): 3891-3898.
[57] Lehner, B., C. R. Liermann, C. Revenga, C. Vörösmarty, B. Fekete, P. Crouzet, P. Döll, M. Endejan, K. Frenken, J. Magome, C. Nilsson, J. C. Robertson, R. Rödel, N. Sindorf and D. Wisser (2011). "High-resolution mapping of the world's reservoirs and dams for sustainable river-flow management." Frontiers in Ecology and the Environment 9(9): 494-502.
[58] Messager, M. L., B. Lehner, G. Grill, I. Nedeva and O. Schmitt (2016). "Estimating the volume and age of water stored in global lakes using a geo-statistical approach." 7: 13603.

which are part of a single grid cell and part of the runoff concentration within a grid cell. Therefore the HydroLakes database is separated into "big" lakes and reservoirs with an area 100 km^2 or a upstream area 5000 km^2 and "small" lakes which represents the non-big lakes. All lakes and reservoirs are combined at grid cell level but big lakes can have the expansion of several grid cells.
Lake and reservoir (LR) data are specified by an id for each LR, type of LR (1 for lake, 2 for reservoir), area of LR, year of constraction of reservoir and average discharge at the outlet of LR.

10.5 Temporal data for each year

10.5.1 Crop coefficient

Based on: MIRCA2000—Global data set of monthly irrigated and rainfed crop areas around the year 2000. http://www.uni-frankfurt.de/45218023/MIRCA (Portmann et al., 2010)[59]

10.5.2 Land cover

Land cover is used to calculate fraction of water, forest, irrigated area, rice irrigated area, sealed (impermeable area) and the remaining fraction for each cell. For each fraction the soil module runs separately. The total runoff of each cell is calculated by weighting the cell according to the different fractions.

Source: https://lta.cr.usgs.gov/GLCC (US Geological Survey Center for Earth Resources Observation and Science)

Forest

Forest land cover is used from from Hansen et al. (2013)[60]

[59] Portmann, F. T., S. Siebert and P. Döll (2010). "MIRCA2000—Global monthly irrigated and rainfed crop areas around the year 2000: A new high-resolution data set for agricultural and hydrological modeling." Global Biogeochemical Cycles 24(1): n/a-n/a.

[60] Hansen, M. C., P. V. Potapov, R. Moore, M. Hancher, S. A. Turubanova, A. Tyukavina, D. Thau, S. V. Stehman, S. J. Goetz, T. R. Loveland, A. Kommareddy, A. Egorov, L. Chini, C. O. Justice, and J. R. G. Townshend. 2013. "High-Resolution Global Maps of 21st-Century Forest Cover Change." Science 342 (15 November): 850–53. Data available on-line from: http://earthenginepartners.appspot.com/science-2013-global-forest.

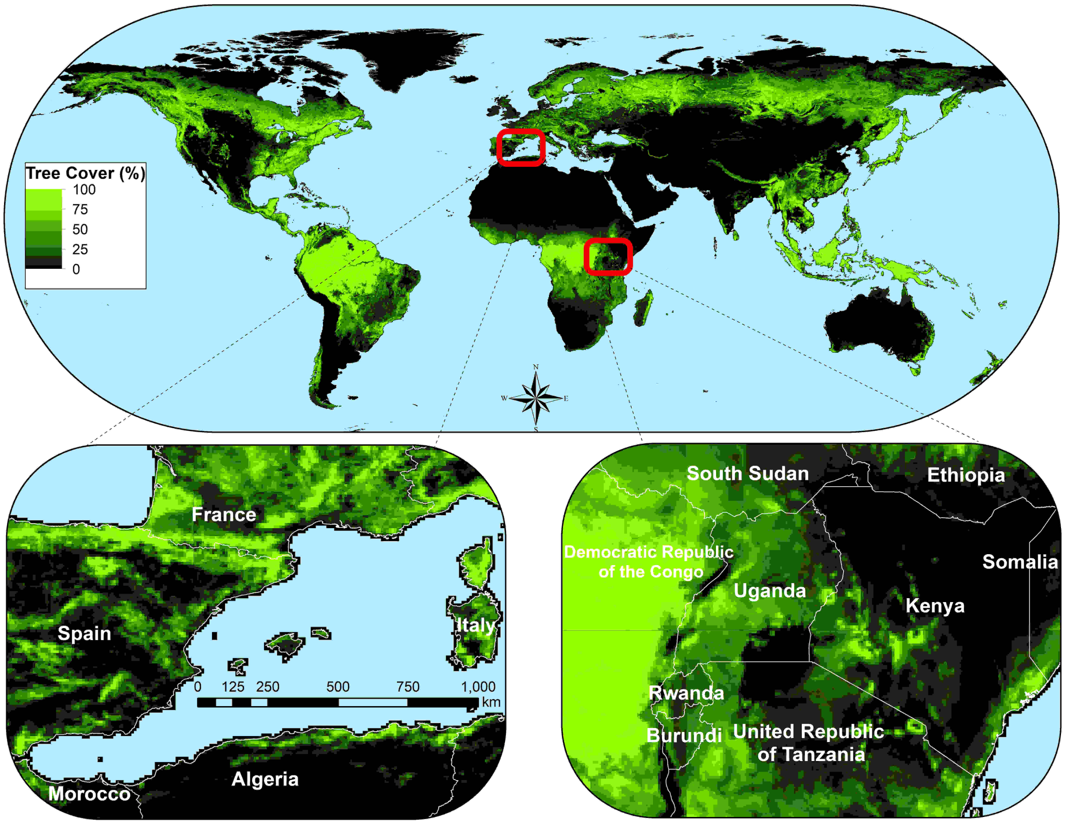


Figure 12: Tree cover in 2010 at 5'

Sealed

Urban area or impervious surface area (ISA) based on.

Based on 1km version of Elvidge et al. (2007)[61]
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3841857/
ftp://ftp.ngdc.noaa.gov/DMSP/

Future projection based on:

Transient, future land use pattern generated by the LU model MAgPIE (Popp et al. 2014[62]; Stevanovic et al. 2016[63]), assuming population growth and economic as in SSP2 and climate change scenario RCP6.0

[61] Elvidge, C. D., Tuttle, B. T., Sutton, P. C., Baugh, K. E., Howard, A. T., Milesi, C., Bhaduri, B., Nemani, R. (2007). Global Distribution and Density of Constructed Impervious Surfaces. Sensors (Basel, Switzerland), 7(9), 1962-1979. doi:10.3390/s7091962

[62] Popp, A., Humpenöder, F., Weindl, I., Bodirsky, B. L., Bonsch, M., Lotze-Campen, H., Müller, C., Biewald, A., Rolinski, S., Stevanovic, M., & Dietrich, J. P. (2014). Land-use protection for climate change mitigation. Nature Climate Change, 4, 1095

[63] Stevanović, M., Popp, A., Lotze-Campen, H., Dietrich, J. P., Müller, C., Bonsch, M., Schmitz, C., Bodirsky, B. L., Humpenöder, F., and Weindl, I.(2016): The impact of high-end climate change on agricultural welfare, Science Advances, 2, 2016. http://advances.sciencemag.org/content/2/8/e1501452

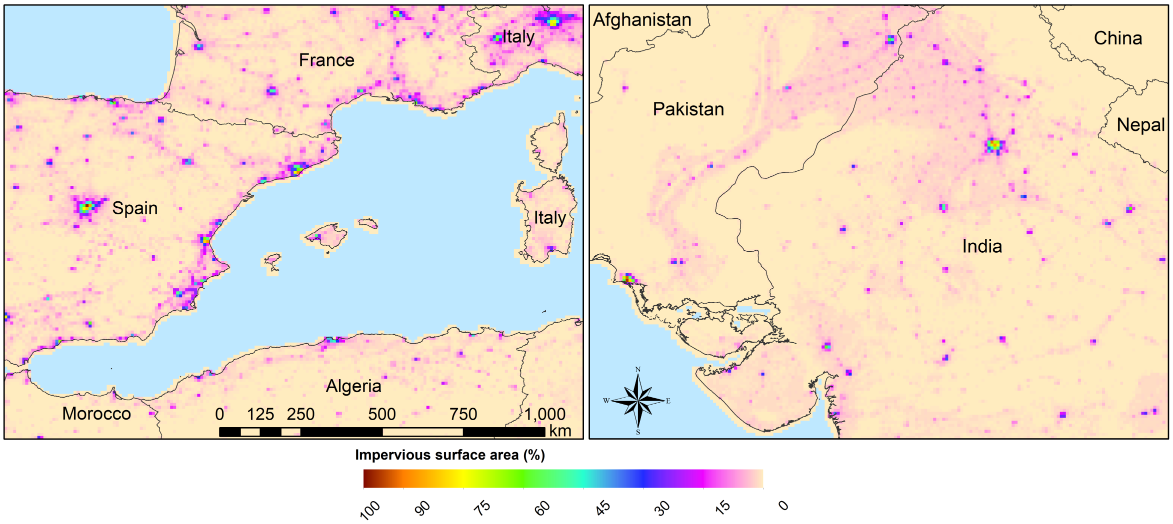


Figure 13: Sealed area in 2010 at 5'

Albedo

Global Albedo dataset from Muller et al., (2012)[64]

10.6 Continous temporal data

10.6.1 Meteorological data

- max, min, avg temperature [K]
- humidity (relative[%] or specific[%])
- surface pressure [Pa]
- radiation (short wave and long wave downwards) [W m-2]
- windspeed [m/s]

If potential evaporation is already calculated in a prerun or from external source

- Precipitation [Kg m-2 s-1] or [m] or [mm] (can be adjusted by a conversion factor in the settings file)
- Temperature (avg) [K]
- Potential evaporation [Kg m-2 s-1] or [m] or [mm] (can be adjusted by a conversion factor in the settings file)

From observation: (see ISI-MIP 2a)

- WFDEI.GPCC (Weedon et al. 2014)[65] WFD—Watch forcing data set: 0.5 3/6 hourly meteorological forcing from ECMRWF reanalysis (ERA40) bias-corrected and extrapolated by CRU TS and GPCP (rainfall) and corrections for under catch

[64] Muller, P. J., P. Lewis, J. Fischer, P. North and U. Framer (2012). The ESA GlobAlbedo Project for mapping the Earth's land surface albedo for 15 Years from European Sensors., paper presented at IEEE Geoscience and Remote Sensing Symposium (IGARSS) IEEE Geoscience and Remote Sensing Symposium (IGARSS) 2012. Munich, Germany. http://www.globalbedo.org

[65] Weedon, G. P., G. Balsamo, N. Bellouin, S. Gomes, M. J. Best and P. Viterbo (2014). "The WFDEI meteorological forcing data set: WATCH Forcing data methodology applied to ERA-Interim reanalysis data." Water Resources Research 50(9): 7505-7514.

- PGMFD v.2 - Princeton (Sheffield et al. 2006)[66]
- GSWP3 (Kim et al.)[67]
- MSWEP (Beck et al. 2017)[68]

From Global Circulation models GCMs (see ISI-Mip 2b)

- HadGem2-ES (Met Office Hadley Centre, UK)
- IPSL-CM5A-LR (Institut Pierre-Simon Laplace, France)
- GFDL-ESM2M (NOAA, USA)
- MIROC-ESM-CHEM (JAMSTEC, AORI, University of Tokyo, NIES, Japan)
- NorESM1-M (Norwegian Climate Centre, Norway)

10.7 References

- Döll, P. and S. Siebert (2002). "Global modeling of irrigation water requirements." Water Resources Research 38(4): 81-811.
- Siebert, S., P. Döll, J. Hoogeveen, J. M. Faures, K. Frenken and S. Feick (2005). "Development and validation of the global map of irrigation areas." Hydrology and Earth System Sciences 9(5): 535-547.

[66] Sheffield, J., G. Goteti and E. F. Wood (2006). "Development of a 50-year high-resolution global dataset of meteorological forcings for land surface modeling." Journal of Climate 19(13): 3088-3111.

[67] Kim, H., S. Watanabe, E.-C. Chang, K. Yoshimura, Y. Hirabayashi, J. Famiglietti and T. Oki "Century long observation constrained global dynamic downscaling and hydrologic implication [in preparation]."

[68] Beck, H. E., A. I. J. M. Van Dijk, V. Levizzani, J. Schellekens, D. G. Miralles, B. Martens and A. De Roo (2017). "MSWEP: 3-hourly 0.25° global gridded precipitation (1979-2015) by merging gauge, satellite, and reanalysis data." Hydrology and Earth System Sciences 21(1): 589-615.

CHAPTER

ELEVEN

CALIBRATION TOOL

Calibration tool for hydrological models
in ../CWATM/calibration

using a distributed evolutionary algorithms in python: DEAP library
http://deap.readthedocs.io/en/master/
https://github.com/DEAP/deap/blob/master/README.md

Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau and Christian Gagné, "DEAP: Evolutionary Algorithms Made Easy", Journal of Machine Learning Research, vol. 13, pp. 2171-2175

The calibration tool was created by Hylke Beck 2014 (JRC, Princeton) hylkeb@princeton.edu
Thanks Hylke for making it available for use and modification
Modified by Peter Burek

The submodule Hydrostats was created 2011 by:
Sat Kumar Tomer (modified by Hylke Beck)
Please see his book Python in Hydrology[69]

11.1 Calibration method

Calibration is using an evolutionary computation framework in Python called DEAP (Fortin et al., 2012). We used the implemented evolutionary algorithm NSGA-II (Deb et al., 2002) for single objective optimization. As objective function we used the modified version of the Kling-Gupta Efficiency (Kling et al., 2012), 2012), with r as the correlation coefficient between simulated and observed discharge (dimensionless), as the bias ratio (dimensionless) and as the variability ratio.

$KGE' = 1 - \sqrt{(r-1)^2) + (\beta-1)^2 + (\gamma-1)^2}$
where: $\beta = \frac{\mu_s}{\mu_o}$ and $\gamma = \frac{CV_s}{CV_o} = \frac{\sigma_s/\mu_s}{\sigma_o/\mu_o}$

[69] http://greenteapress.com/pythonhydro/pythonhydro.pdf

Where CV is the coefficient of variation, is the mean streamflow [m3 s1] and is the standard deviation of the streamflow [m3 s1]. KGE', r, and have their optimum at unity. The KGE' measures the Euclidean distance from the ideal point (unity) of the Pareto front and is therefore able to provide an optimal solution which is simultaneously good for bias, flow variability, and correlation. For a discussion of the KGE objective function and its advantages over the often used Nash–Sutcliffe Efficiency (NSE) or the related mean squared error see (Gupta et al., 2009). The calibration uses general a population size (μ) of 256, a recombination pool size () of 32.The number of generations was set to 30, which we found was sufficient to achieve convergence for stations

11.1.1 Further ideas for calibration

- Regionalization see (Samaniego et al. 2017) and (Beck et al. 2016)
- Using Budyko see (Greve et al. 2016)

11.2 Suggested calibration parameters

Snow
1. Snowmelt coefficient in [m/C deg/day] as a degree-day factor
Evapotranspiration
2. Crop factor as an adjustment to crop evapotranspiration
Soil
3. Soil depth factor: a factor for the overall soil depth of soil layer 1 and 2
4. Preferential bypass flow: empirical shape parameter of the preferential flow relation
5. Infiltration capacity parameter: empirical shape parameter b of the ARNO model
Groundwater
6. Interflow factor: factor to adjust the amount which percolates from interflow to groundwater
7. Recession coefficient factor: factor to adjust the base flow recession constant (the contribution from groundwater to baseflow)
Routing
8. Runoff concentration factor: a factor for the concentration time of run-off in each grid-cell
9. Channel Manning's n factor: a factor roughness in channel routing
10. Channel, lake and river evaporation factor: factor to adjust open water evaporation
Reservoir & lakes
11. Normal storage limit: the fraction of storage capacity used as normal storage limit
12. Lake A factor : factor to channel width and weir coefficient as a part of the Poleni weir equation

11.3 Calibration tool structure

```
calibration
│-  readme.txt
│-  readme.txt
│
└--observed_data
│   └─ lobith2006.cvs, ...
│
└--templates
│   └-- runpy.bat, runpy.sh
│   └-- settings.ini
```

11.4 How it works

The calibration tool builds up a single-objective obtimization framework using the Python libray DEAP For each run it triggers the run of the hydrological model:

- using a template of the settings file
- replacing the output folder in this template file
- replace placeholders with the values of calibration parameters, the limit of the parameter range is given in the file: ParamRanges.csv

After each run the model run is compared to observed values (e.g. observed_data/lobith2006.csv)

After the calibration, statistics and the best run is printed output

11.5 What is needed

1. The template files in ../templates have to be adjusted

- runpy.bat: the path to cwatm.py have to be set correctly (for linux a .sh file has to be created)
- The actual version of a cwatm settings file has to modified:
- replacing the output folder with the placeholder: %run_rand_id

```
#-------------------------------------------------------
# CALIBARTION PARAMETERS
#-------------------------------------------------------
[CALIBRATION]

# These are parameter which are used for calibration
# could be any parameter, but for an easier overview, tehey are collected here
# in the calibration template a placeholder (e.g. %arnoBeta) instead of value

OUT_Dir = %run_rand_id
```

- putting the output variables in e.g. OUT_TSS_Daily = discharge or monthly average discharge OUT_TSS_MonthAvg = discharge

```
OUT_TSS_Daily = discharge
OUT_TSS_MonthAvg = discharge
```

- delete all the output variables in the template (mostly at the end of the file)
- replacing calibration parameter values with a placeholder: e.g. %SnowMelt

```
# Snow  SnowMeltCoef = 0.004
SnowMeltCoef = %SnowMelt
# Cropf factor correction
crop_correct =  %crop
#Soil
soildepth_factor = %soildepthF
#Soil preferentialFlowConstant = 4.0, arnoBeta_factor = 1.0
preferentialFlowConstant = %pref
arnoBeta_add = %arnoB
# interflow part of recharge factor = 1.0
```

(continues on next page)

(continued from previous page)

```
factor_interflow = %interF
# groundwater recessionCoeff_factor = 1.0
recessionCoeff_factor = %reces
# runoff concentration factor runoffConc_factor = 1.0
runoffConc_factor = %runoff
#Routing manningsN factor [0.1 - 10.0] default  1.0
manningsN = %CCM
# reservoir  normal storage limit (fraction of total storage, [-]) [0.15 - 0.85]␣
↪default 0.5
normalStorageLimit = %normalStorageLimit
# lake parameter - factor to alpha: parameter of of channel width and weir␣
↪coefficient  [0.33 - 3.] dafault 1.
lakeAFactor = %lakeAFactor
# lake wind factor - factor to evaporation from lake [0.8 - 2.] dafault 1.
lakeEvaFactor = %lakeEvaFactor
```

2. the range of parameter space has to be defined in ParamRanges.csv

```
ParameterName,MinValue,MaxValue
SnowMelt,0.001,0.007
crop,0.8,3.0
soildepthF, 0.8,1.8
pref,0.5,8
arnoB,0.01,1.0
interF, 0.33,3.0
reces,0.1,10
runoff,0.1,5
CCM,0.1,10.0
normalStorageLimit,0.15,0.85
lakeAFactor,0.333,3.0
lakeEvaFactor,0.5,3.0
No,1,100
```

3. The observed discharge has to be provided in an .cvs file e.g. observed_data/lobith2006.csv

In the template settings the date has to be set, so that the period of observed discharge is between SpinUp and StepEnd

```
#-------------------------------------------------------
[TIME-RELATED_CONSTANTS]
#-------------------------------------------------------

# StepStart has to be a date e.g. 01/06/1990
# SpinUp or StepEnd either date or numbers
# SpinUp: from this date output is generated (up to this day: warm up)

StepStart = 1/1/1990
SpinUp = 1/1/1995
StepEnd =  31/12/2010
```

4. And empty ../catchments directory needs to be created

5. A few option in the settings.txt have to be adjusted (how many runs?, a first run with standard parameters? etc)

```
[DEFAULT]
Root = /c/watmodel/CWATM
RootPC = C:/watmodel/CWATM
Rootbasin = calibration_rhine

ForcingStart = 1/1/2000
ForcingEnd = 31/12/2010
timeperiod = daily

[ObservedData]
Qtss = observed_data/lobith.csv
Column = lobith
Header = River: Rhine  station: Lobith

[Validate]
Qtss = observed_data/lobith_val.csv
ValStart = 1/1/1990
ValEnd = 31/12/1999


[Path]
Templates = templates
SubCatchmentPath = catchments
ParamRanges = ParamRanges.csv

[Templates]
ModelSettings = settings.ini
RunModel = runpy.sh

[Option]
firstrun = False
para_first = [0.0022, 1.72, 1.24, 7.07, 0.55, 1.92, 2.81, 0.74,1.34,0.35,2.04,1.0, 1.]
# Snowmelt, crop KC, soil depth,pref. flow, arno beta, interflow factor, groundwater␣
↪recession,
# runoff conc., routing, manning factor, normalStorageLimit, lakeAFactor,
↪lakeEvaFactor,No of run
bestrun = True

[DEAP]
maximize = True
use_multiprocessing = 1
ngen = 30
mu = 256
lambda_ = 32
```

6. run python calibration_single.py settings.txt

11.6 Recommendations

1. Run the model first to store the pot. evaporation results

Afterwards use the stored evaporation to run the calibration
calc_evaporation = False

2. Run the model and store the last day to be used as initial condition for the calibration runs

Best is to use a long term run for this.

```
[INITITIAL CONDITIONS]
#-------------------------------------------------------

# for a warm start initial variables a loaded
# e.g for a start on 01/01/2010 load variable from 31/12/2009
load_initial = False
initLoad = $(FILE_PATHS:PathRoot)/init/Rhine_19891231.nc

# saving variables from this run, to initiate a warm start next run
# StepInit = saving date, can be more than one: 10/01/1973 20/01/1973
save_initial = False
initSave = $(FILE_PATHS:PathRoot)/init/Rhine
StepInit = 31/12/1989 31/12/2010
```

load_initial = False
save_initial = True

During calibration use:
load_initial = True
save_initial = False

3. Use a long SpinUp time (> 5 years to give groundwater enough time)

11.7 References

- Beck, H. E., A. I. J. M. van Dijk, A. de Roo, D. G. Miralles, T. R. McVicar, J. Schellekens and L. A. Bruijnzeel (2016). "Global-scale regionalization of hydrologic model parameters." Water Resources Research 52(5): 3599-3622.
- Deb, K., A. Pratap, S. Agarwal and T. Meyarivan (2002). "A fast and elitist multiobjective genetic algorithm: NSGA-II." IEEE Transactions on Evolutionary Computation 6(2): 182-197.
- Fortin, F. A., F. M. De Rainville, M. A. Gardner, M. Parizeau and C. Gagńe (2012). "DEAP: Evolutionary algorithms made easy." Journal of Machine Learning Research 13: 2171-2175.

- Greve, P., L. Gudmundsson, B. Orlowsky and S. I. Seneviratne (2016). “A two-parameter Budyko function to represent conditions under which evapotranspiration exceeds precipitation.” Hydrology and Earth System Sciences 20(6): 2195-2205.
- Gupta, H. V., H. Kling, K. K. Yilmaz and G. F. Martinez (2009). “Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling.” Journal of Hydrology 377(1-2): 80-91.
- Kling, H., M. Fuchs and M. Paulin (2012). “Runoff conditions in the upper Danube basin under an ensemble of climate change scenarios.” Journal of Hydrology 424-425: 264-277.
- Samaniego, L., R. Kumar, S. Thober, O. Rakovec, M. Zink, N. Wanders, S. Eisner, H. Müller Schmied, E. Sutanudjaja, K. Warrach-Sagi and S. Attinger (2017). “Toward seamless hydrologic predictions across spatial scales.” Hydrology and Earth System Sciences 21(9): 4323-4346.

CHAPTER

TWELVE

CALIBRATION TUTORIAL

12.1 What you need

Python 3.7.x 64 bit and a running CWatM (libraries netCDF4, numpy, scipy, GDAL) In addition: **library**

deap Calibration is using a distributed evolutionary algorithms in python: DEAP library

Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau and Christian Gagné, "DEAP: Evolutionary Algorithms Made Easy", Journal of Machine Learning Research, vol. 13, pp. 2171-2175

You can install it with: Pip install deap (you might change into the folder ../python/Scripts/)

- Make sure that python 3.7.x is working
- Make sure that CWatM is running in non calibration mode
- For some of the following steps it is easier to have PCRaster installed: http://pcraster.geo.uu.nl/

12.2 Running calibration

1. Look into the settings file of the calibration folder.
2. look into runCalibration.bat. If python is in your computer path everything should be ok, otherwise put in the path to python
3. look into templates/runpy.bat. Put the path to python in if necessary
4. look into templates/settings.ini. Put the pathes in a right way that it fits to your computer:

```
[FILE_PATHS]
#-------------------------------------------------------
PathRoot = P:/watmodel/CWATM/calibration_tutorial
PathOut = $(PathRoot)/output
PathMaps = $(PathRoot)/CWATM_data/cwatm_input
PathMeteo = $(PathRoot)/climate
```

5. in observed_data/yukon2001.cvs you find the observed data:

```
- make sure the name in the header is the same as in [ObservedData] Column
- make sure that there are enough data in (from ForcingStart to ForcingEnd)
```

6. make sure the folder catchments is empty! Before each try this folder has to be empty

12.3 Run runCalibration.bat

1. go for testing (see below)
2. go for testing again (see below)
3. Change use_multiprocessing =1 in settings.txt
4. Run runCalibration.bat and after some time something should appear on your window

12.4 For testing

- Change use_multiprocessing = 0 in settings.txt
- Delete catchments but keep the empty folder
- Run runCalibration.bat and wait till catchment/00_001 gets filled, then interrupt

1. Change to catchments/00_001
2. Run runpy00_001.bat
3. See what errors come up and change settings-Run00_001.ini
4. Change template/setting.ini in the same way
5. Do this again and again till no error

12.5 Running it on your computer

It will be really slow on Windows using data on the the server – next step run it on your PC

- copy the whole folder P:watmodelCWATMcalibration_tutorial to your PC (only 15 GB)
- (but maybe you have already parts of it on your computer – like the big climate input files)
- Make it work on your computer:

```
Changing file paths in templates/settings.ini, setting.txt
Changing the path for python in runCalibration.bat and templates/runpy.bat
```

12.6 Preparation for another catchment

12.6.1 Preparing the observed dataset – discharge

Calibration works by comparing simulated discharge with observed discharge using an objective function: Here we use the Kling-Gupta Efficiency but we can also use Nash-Sutcliffe Efficiency . Please find some more information on the objective function an on the evolutionary computation framework used for calibration on: https://cwatm.github.io/calibration.html

- The observed values can be stored as daily values or monthly values

- The observed values should be at least cover 5 years (best is 10-15 years)
- The observed discharge has to be stored as textfile in:

```
./observed_data/nameofstation.cvs
And has to look like this:
date,yukon_pilot_station
2001-04-01,1302.6
2001-04-02,1302.6
2001-04-03,1302.6
2001-04-04,1302.6
...
...
2013-12-31,2647.6
```

- Or:

```
date ,zhutuo
2002-01-01,3229.0
2002-02-01,2979.2
2002-03-01,3229.0
```

Format:

- Date format like this year-month-day [yyyy-mm-dd]
- Separated by a comma
- Discharge in [m3/s]
- If a value is missing that is not a problem (as long as the time series is long enough):

```
it should like this: (no value after the comma)
2002-01-12,
```

- For each day (or month) a line

Settings.txt

In the settings file the lines:

```
[ObservedData]
Qtss = observed_data/zhutuo_2002month.csv
Column = zhutuo
Header = River: Yangtze  station: Zhutuo
```

Should correspondent to the name and header in the observed discharge.cvs

The lines:

```
ForcingStart = 1/1/2002
ForcingEnd = 31/12/2013
```

Should correspondent to the amount of lines in the observed discharge.cvs

12.7 Creating an initial netcdf file for warm start

It is best to have a long warm up phase especially for groundwater: See also: https://cwatm.github.io/setup.html#initialisation

You can run CWatM for a couple of years (20 years or more) and store the last days storage values in a file. This file can be read in to enable a ‘warm” start

- change use_multiprocessing = 0 in settings.txt
- Delete catchments but keep the empty folder
- Run runCalibration.bat and wait till catchment/00_001 gets filled, then interrupt
- Change to catchments/00_001

Open the settings-Run_001.init

- Change load_initial = True to load_initial = False
- save_initial = True
- initSave = $(FILE_PATHS:PathRoot)/CWATM_init/testx
- StepInit = 31/01/1996 (change it to a date 1 month after your StepStart)
- Run runpy00_001.bat

There should be a file ./CWATM_init/testx_19960131.nc

- Change to: load_initial = True
- initLoad = $(FILE_PATHS:PathRoot)/CWATM_init/testx_19960131.nc
- Run runpy00_001.bat

If it work then it used the initial file you generate before (that was just a test)

Now change to:

- StepStart = 1/1/1961
- StepEnd = 31/12/2013
- load_initial = False
- save_initial = True
- initSave = $(FILE_PATHS:PathRoot)/CWATM_init/station_name
- StepInit = 31/12/2013
- Run runpy00_001.bat

This should have generated a file ./CWATM_init/station_name_20131231.nc

And again:

- StepStart = 1/1/1961 (some 20 years or longer)
- StepEnd = 31/12/1995 (a day before your normal running day)
- load_initial = True
- initLoad = $(FILE_PATHS:PathRoot)/CWATM_init/station_name_20131231.nc
- save_initial = True
- initSave = $(FILE_PATHS:PathRoot)/CWATM_init/station_name
- StepInit = 31/12/1995 (a day before your running day)
- Run runpy00_001.bat

This should have generated a file ./CWATM_init/station_name_19951231.nc

And last part:

- Change StepStart and StepEnd back to original values
- load_initial = True
- initLoad = $(FILE_PATHS:PathRoot)/CWATM_init/station_name_19951231.nc
- save_initial = False
- Run runpy00_001.bat

If it works, do the same in the ./template/settings.ini

Note: You have now a "warm" start for every calibration run

12.8 Cutting out a catchment as mask map

See the .doc file in P:watmodelCWATMcalibration_tutorialcalibrationtoolscut_catchmentFor a description:

Requirements: PCRASTER:

We do no need the python version, I think downloading, extracting and setting of the paths in P:watmodelCWATMcalibration_tutorialcalibrationtoolscut_catchmentcatchconfig_win.ini Creating the 2 potential evaporation files in advance

Potential evaporation is Calculated with Penman-Monteith in CWatM, but it is not part of the calibration = there is no change in pot. Evaporation. In order to make the calibration computational faster the results of pot evaporation could be stored and used every time.

For the 30min this is done already as global map set, but for the 5min these files become too big. So they have to be produced for each basin separately

Same preparation as for **Creating an initial netcdf file for warm start** see above There should be a folder catchments00_001 with a working run for 001.

Open the settings-Run_001.init

Change:

```
[Option] calc_evaporation = True
[TIME-RELATED_CONSTANTS] SpinUp = None
[EVAPORATION]
OUT_Dir = $(FILE_PATHS:PathOut)
OUT_MAP_Daily = ETRef, EWRef
```

Run runpy00_001.bat There should be a file ETRef.nc and EWRef in the output directory

Rename the files e.g. ETRef.nc to ETRef_yangtze.nc, EWRef.nc to EWRef_yangtze.nc and copy it to PathMeteo (or somewhere else, you have to put the path in)

Open the settings-Run_001.init

Change:

```
[Option] calc_evaporation = False
[TIME-RELATED_CONSTANTS] SpinUp = -> to the time it was before
[Meteo]
daily reference evaporation (free water)
E0Maps = $(FILE_PATHS:PathMeteo)/EWRef_yangtze
daily reference evapotranspiration (crop)
ETMaps = $(FILE_PATHS:PathMeteo)/ETRef_yangtze
[EVAPORATION]
OUT_Dir = $(FILE_PATHS:PathOut)       !!! outcomment this again - important
OUT_MAP_Daily = ETRef, EWRef
```

Test it: Run runpy00_001.bat

And change the settings.ini in templates in the same way

12.9 Calibration of a downstream catchment

Calibration of a downstream catchment (upstream catchment is already calibrated) can be done using:

- The catchment area of the downstream catchment minus the upstream catchment
- The missing discharge from the upstream catchment is replaced by an inflow file

1. Cut the mask map, so that the upstream catchment is NOT in the mask map anymore
2. Detect the point(s) downstream of the inflow points
3. Run the best calibration scenario(s) of the upstream catchments again to produce long timeserie(s) of the outlet(s) point
4. Create an inflow file from the long timeseries of outlet(s)
5. Create a downstream calibration settings (directories, templates etc.)

Test the catchment!

6. Change the settings file of the downstream calibration so that it includes the inflow from upstream

Test it! 7. Create initial file for warm start

12.9.1 Cutting the mask map

Assuming you have a mask map of the whole catchment (e.g. Yangtze.map and the station points (here Zhutuo 105.75 28.75 and Yichang 111.25 30.75 1. Creating catchment for Zhutuo: catchment 105.75 28.75 ldd_yangtze.map zhu1.map 2. Creating catchment for Yichang: catchment 111.25 30.75 ldd_yangtze.map yi1.map 3. Creating Yichang without Zhutuo:

```
pcrcalc a2.map = cover(scalar(zhu1.map)*2,scalar(yi1.map))
pcrcalc yichang.map = boolean(if(a2.map eq 1,a2.map))
```

Result is a maskmap: Yichang.map

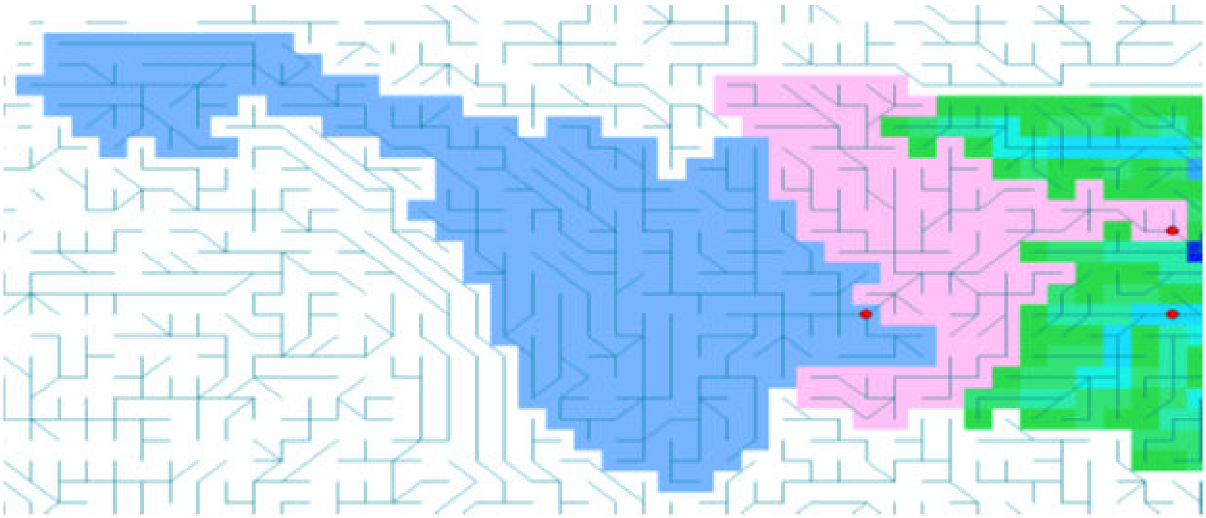


Figure 1: Upstream catchment (blue) and downstream catchment (red)

12.9.2 Detecting the downstream point

The inflow point of the new catchment has to be in the new mask and preferable one grid cell in flow direction below the upstream station e.g. 1 gridcell North East of Zhutuo (see purple circle in fig. 2)

The inflow point has the lon/lat 106.25 29.25

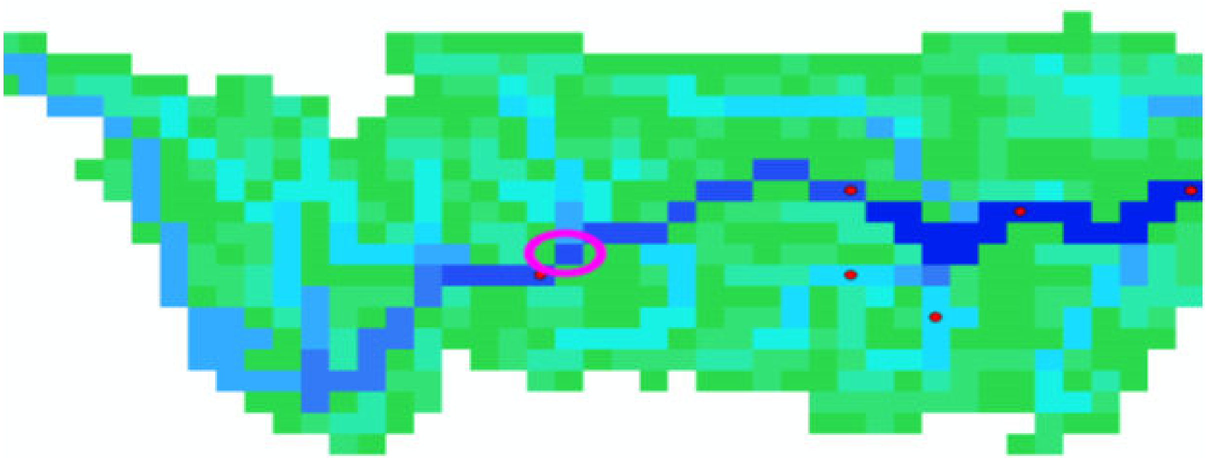


Figure 2: Downstream point

12.9.3 Run the best calibration scenario upstream

In order to get a long inflow timeserie for the inflow point (here: Zhutuo) you need to run the best scenario of the upstream catchment (here: 31_best)

- Change into the folder ../catchments/best
- Change settings file from:

```
StepStart = 1/1/1996
SpinUp = 1/1/2002
StepEnd =  31/12/2013
```

- To:

```
StepStart = 1/1/1990
SpinUp = 1/1/1996
StepEnd =  31/12/2013
```

Results is a time series from 1/1/1990 – 31/12/2013 in: discharge_daily.tss

12.9.4 Create an inflow file from the long timeseries of outlet(s)

- Create a folder ../inflow
- Copy the ../catchments/31_best/discharge_daily.tss to ../inflow/zhutuo.tss

12.9.5 Create a downstream calibration settings (directories, templates etc.)

Create downstream calibration settings as before

- Copy everything from upstream catchment (e.g. zhutuo) but not catchments
- Create empty catchments folder
- Create a observed discharge file in observed
- Change settings.txt accordingly
- Change settings.ini accordingly

Test the catchment setting!

But do not create an initial run yet!

12.9.6 Change the settings file

Change the settings file of the downstream calibration so that it includes the inflow from upstream Change the part of the settings.ini:

```
[Option]
inflow = True
[INFLOW]
#-------------------------------------------------------
# if option inflow = true
# the inflow from outside is added as inflowpoints
In_Dir = $(FILE_PATHS:PathRoot)/calibration/calibration_yichang/inflow
# nominal map with locations of (measured)inflow hydrographs [cu m / s]
InflowPoints = 106.25 29.25
InLocal = True
.
# if InflowPoints is a map, this flag is to identify if it is global (False) or local␣
↪(True)
# observed or simulated input hydrographs as time series [cu m / s]
# Note: that identifiers in time series have to correspond to InflowPoints
```

(continues on next page)

(continued from previous page)

```
# can be several timeseries in one file or different files e.g. main.tss mosel.tss
QInTS = zhutuo.tss
```

Test it!

Generate initial file for warm start Use initial file for calibration

12.10 Joining best sub-basin results to calibration maps

1. You need all runs done for all sub-basins
2. A region map

For each subbasin a unique number e.g. Zambezi basin

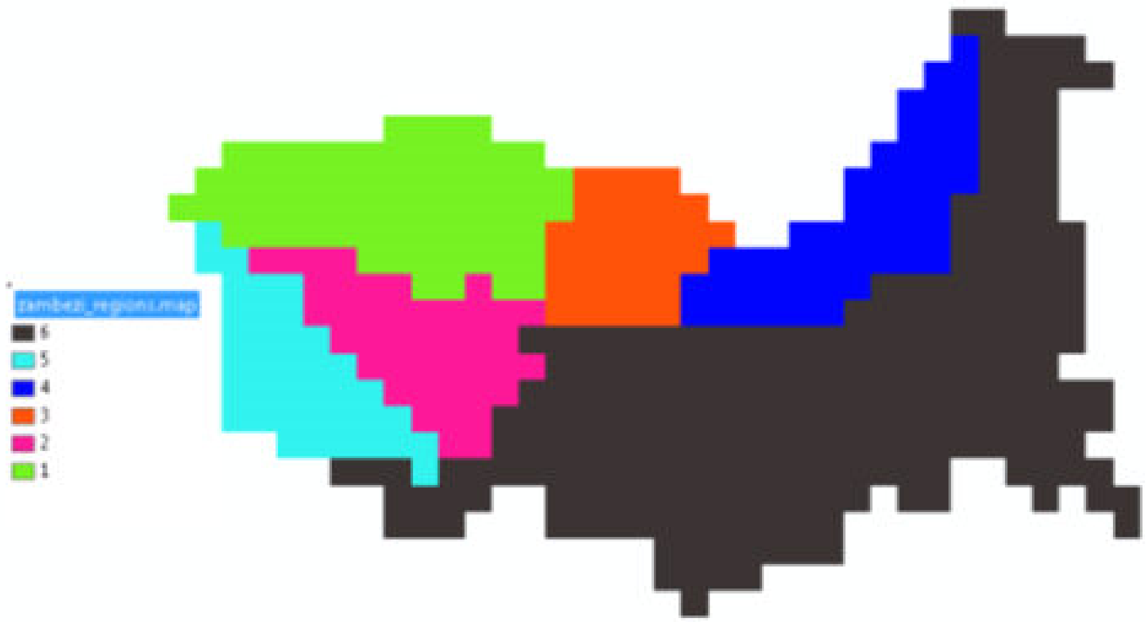


Figure 3 Sub-basin map with a unique identifier for each subbasin

3. You need a working PCRaster installation
4. The settings file settings.txt has to be changed:

```
[DEFAULT]
Root = P:/watmodel/CWATM/calibration/calibration_zambezi
# root directory where all subbasin are in
.
[Catchments]
catch = lukulu, katima, kafue, luangwa, kwando, tete
# name of the subbasin, has to be the same as the folder name in root
# the order has to be the same as in the region map
.
[region]
regionmap = P:/watmodel/CWATM/calibration_tutorial/calibration/
↪CreateCalibrationMaps/zambezi_regions.map
# region map, the order has to be the same a [Catchment]
```

(continues on next page)

(continued from previous page)

```
.
[Path]
Templates = %(Root)s/templates
SubCatchmentPath = %(Root)s/catchments
ParamRanges = %(Root)s/Join/ParamRanges.csv
.
Result = P:/watmodel/CWATM/calibration_tutorial/calibration/CreateCalibrationMaps/
↪results
# here are the results
.
PCRHOME = C:\PCRaster\bin
# Where is your PCraster installation?
```

5. Run python CAL_5_PARAMETER_MAPS.py

12.11 Calibration tool structure

12.12 References

- Beck, H. E., A. I. J. M. van Dijk, A. de Roo, D. G. Miralles, T. R. McVicar, J. Schellekens and L. A. Bruijnzeel (2016). "Global-scale regionalization of hydrologic model parameters." Water Resources Research 52(5): 3599-3622.
- Deb, K., A. Pratap, S. Agarwal and T. Meyarivan (2002). "A fast and elitist multiobjective genetic algorithm: NSGA-II." IEEE Transactions on Evolutionary Computation 6(2): 182-197.

- Fortin, F. A., F. M. De Rainville, M. A. Gardner, M. Parizeau and C. Gagńe (2012). “DEAP: Evolutionary algorithms made easy.” Journal of Machine Learning Research 13: 2171-2175.
- Greve, P., L. Gudmundsson, B. Orlowsky and S. I. Seneviratne (2016). “A two-parameter Budyko function to represent conditions under which evapotranspiration exceeds precipitation.” Hydrology and Earth System Sciences 20(6): 2195-2205.
- Gupta, H. V., H. Kling, K. K. Yilmaz and G. F. Martinez (2009). “Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling.” Journal of Hydrology 377(1-2): 80-91.
- Kling, H., M. Fuchs and M. Paulin (2012). “Runoff conditions in the upper Danube basin under an ensemble of climate change scenarios.” Journal of Hydrology 424-425: 264-277.
- Samaniego, L., R. Kumar, S. Thober, O. Rakovec, M. Zink, N. Wanders, S. Eisner, H. Müller Schmied, E. Sutanudjaja, K. Warrach-Sagi and S. Attinger (2017). “Toward seamless hydrologic predictions across spatial scales.” Hydrology and Earth System Sciences 21(9): 4323-4346.

CHAPTER

THIRTEEN

LICENSE AND DOWNLOAD INFO

Contents

13.1 GNU General public license

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. http://fsf.org
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

13.1.1 Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

13.1.2 TERMS AND CONDITIONS

0. Definitions

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

> a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
>
> b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
>
> c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
>
> d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

> a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

> a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
>
> b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
>
> c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
>
> d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
>
> e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
>
> f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

13.1.3 Terms and Conditions of Use of the Community Water Model

Stipulations regarding the use of the Commumunity Water Model (hereinafter *CWatM*) provided here by the International Institute for Applied System Analysis, Laxenburg, Austria (IIASA) are as follows:

1. Copyright

The owners of the copyright of CWatM are Peter Burek, Yoshihide Wada, Yusuke Satoh and Peter Greve (hereinafter the "Developer") at IIASA.

2. Reprints and quotations

In the event of the publication of a document or scientific paper using CWatM or its modified version, the following paper must be referenced:

Burek, P., Satoh, Y., Kahil, T., Tang, T., Greve, P., Smilovic, M., Guillaumot, L., and Wada, Y.: Development of the Community Water Model (CWatM v1.04) A high-resolution hydrological model for global and regional assessment of integrated water resources management, Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2019-214, in review, 2019.

Link to paper in Geoscientific Model Development[70]

[70] https://www.geosci-model-dev-discuss.net/gmd-2019-214

3. Usage

Usage is regulated by GNU General Public License V3 (see above)

4. Final Remarks

We as developers belief that CWatM should be utilize to encourage ideas and to advance hydrological, environmental science and stimulate integration into other science disciplines.

CWatM is based on existing knowledge of hydrology realized with Python and C++. Especially ideas from HBV, PCR-GLOBE, LISFLOOD, H08, MatSiro are used for inspiration.

Your support is more then welcome and highly appreciated

The developers of CWAT Model

13.2 Download

13.2.1 Download pdf

`CWATM_MANUAL.pdf`

13.2.2 Source code - Community Water Model

The source code of CWatM is freely available under the GNU General Public License.

Please see its *Terms and Conditions of Use of the Community Water Model* (page 165)

> Source code on Github repository of CWATM[71]

Please use the actual Python 3.7 version
From 2019 we are not maintaining the Python 2.7 version
In case of trouble, try the executable version cwatmexe.zip

> **Warning:** The source code is free, but we can give only limited support, due to limited person power!

13.2.3 Global dataset

If you are interested in obtaining the gloabl data set,
please send an email to wfas.info@iiasa.ac.at
We will give you access to our ftp server

[71] https://github.com/CWatM/CWatM

CHAPTER

FOURTEEN

SOURCE CODE

Contents

14.1 Source code on Github

The source code of CWatM is freely available under the GNU General Public License.
Please see its *Terms and Conditions of Use of the Community Water Model* (page 165)

Source code on Github repository of CWATM[73]

Please use the actual Python 3.7 version
From 2019 we are not maintaining the Python 2.7 version
In case of trouble, try the executable version cwatmexe.zip

Warning: The source code is free, but we can give only limited support, due to limited person power!

[73] https://github.com/CWatM/CWatM

14.2 Source code

```
--------------------------------------------------
######## ##          ##  ####  ######  ##    ##
##       ##          ## ##  ##   ##   ####  ####
##        ##        ##  ##  ##   ##   ## #### ##
##        ##   ##   ## ########  ##  ##   ##   ##
##         ## #### ##  ##    ##  ##  ##        ##
##         ####  #### ##      ## ## ##          ##
##########  ##    ##  ##      ## ## ##          ##

Community WATer Model
--------------------------------------------------
```

14.2.1 Modules of CWatM

The source code of CWatM has a modular structure. Modules for data handling, output, reading as parsing the setting files are in the **management_modules** folder.

Modules for hydrological processes e.g. snow, soil, groundwater etc. are located in the folder **hydrological_modules**.
The kinematic routing and the C++ routines (for speeding up the computational time) are in the folder
hydrological_modules/routing_reservoirs.
Fig. 1 shows the modules of CWatM and their connections
Fig. 2 shows a profile with of the workflow and timing of CWatM.

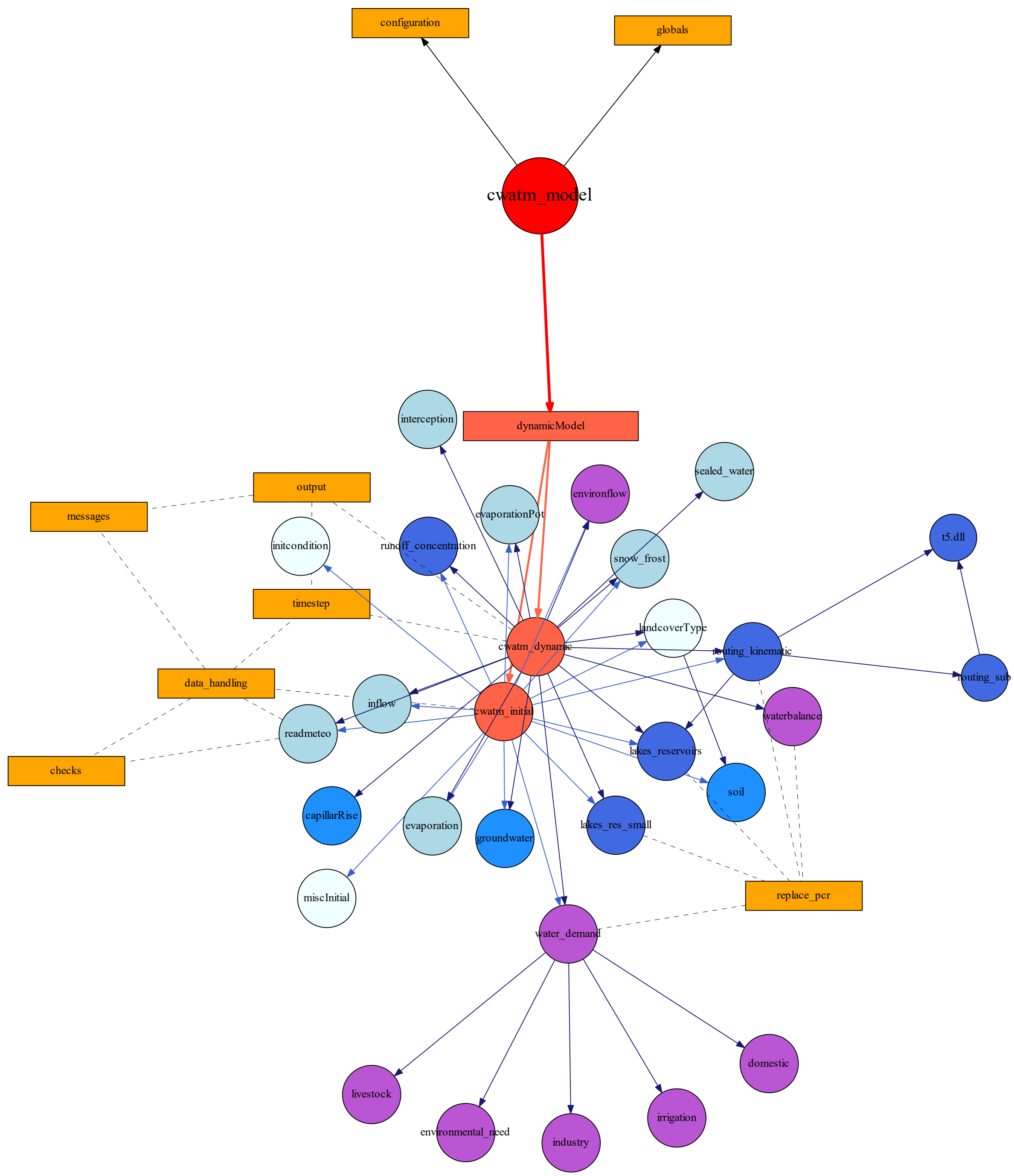


Figure 1: Schematic graph of CWatM modules

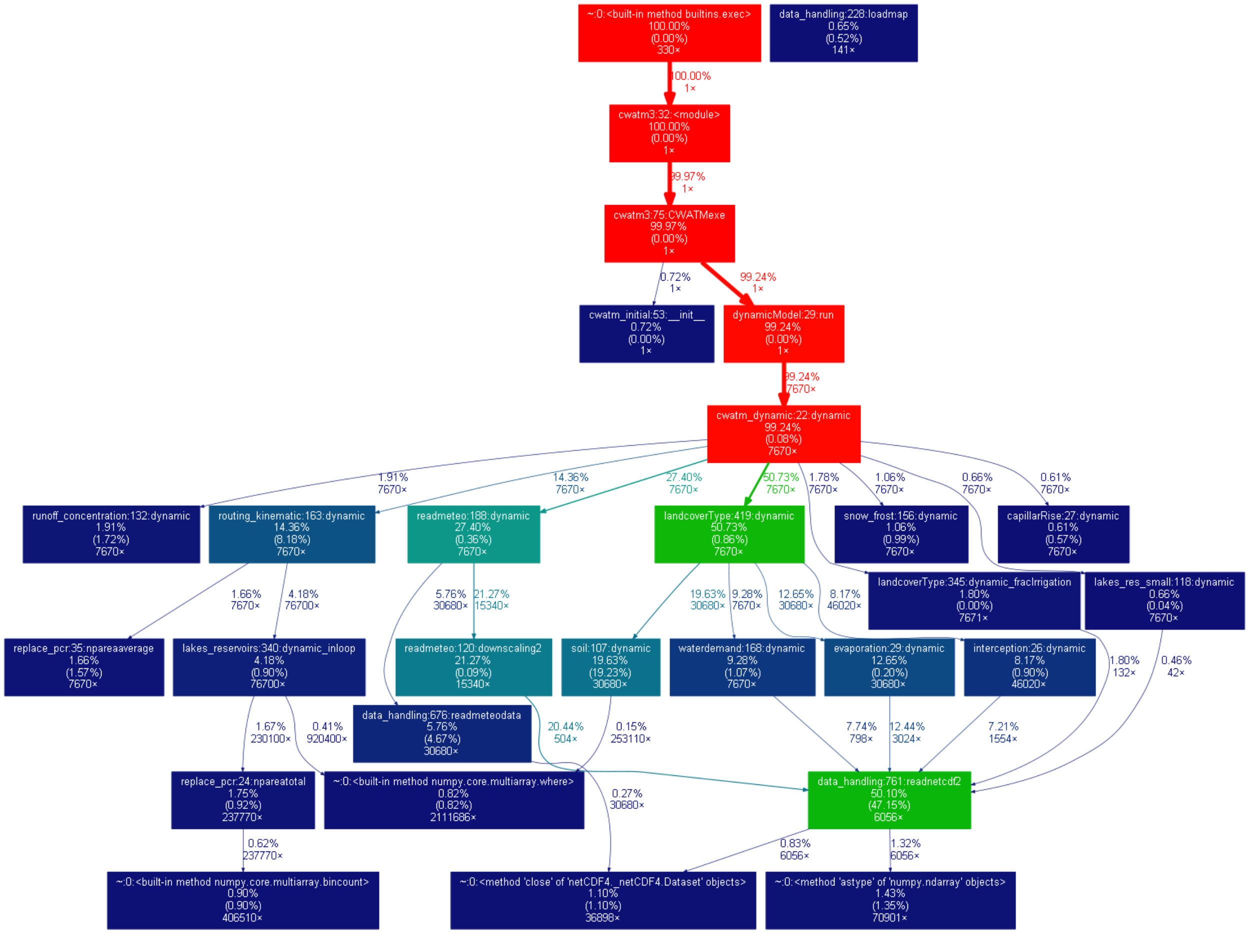


Figure 2: Graphical profile of CWatM run for Rhine catchment from 1/1/190-31/12/2010

Note:

Figure created with:
python -m cProfile -o l1.pstats run_cwatm.py settings1.ini -l
gprof2dot -f pstats l1.pstats | dot -T png -o callgraph.png

14.2.2 Source code description

run_cwatm

Note: Base module: base module to start CWatM: e.g. python run_cwatm.py settings.ini

```
----------------------------------------------------
#######  ##          ##   ####   ######   ##     ##
##       ##          ##  ##  ##    ##    ####   ####
##        ##        ##   ##  ##    ##    ## #### ##
##        ##   ##   ##  ########   ##   ##    ##   ##
##         ## #### ##   ##     ##   ##   ##         ##
```

(continues on next page)

(continued from previous page)

```
##          ####  #### ##      ## ## ##         ##
##########  ##     ##  ##      ## ## ##         ##

Community WATer Model
```


——————————————————

cwatm_model module

Note: Base module: run from root with settings file e.g. python run_cwatm.py settings.ini

class cwatm.cwatm_model.**CWATModel**

Bases: *cwatm.cwatm_initial.CWATModel_ini* (page 173), *cwatm.cwatm_dynamic.CWATModel_dyn* (page 172)

Initial and dynamic part of the CWATM model * initial part takes care of all the non temporal initialiation procedures * dynamic part loops over time

dynamic()

Dynamic part of CWATM calls the dynamic part of the hydrological modules Looping through time and space

Note: if flags set the output on the screen can be changed e.g.

- v: no output at all
- l: time and first gauge discharge
- t: timing of different processes at the end

i = 1

cwatm_dynamic module

class cwatm.cwatm_dynamic.**CWATModel_dyn**

Bases: *cwatm.management_modules.dynamicModel.DynamicModel* (page 218)

dynamic()

Dynamic part of CWATM calls the dynamic part of the hydrological modules Looping through time and space

Note: if flags set the output on the screen can be changed e.g.

- v: no output at all
- l: time and first gauge discharge

- t: timing of different processes at the end

i = 1

cwatm_initial module

class cwatm.cwatm_initial.**CWATModel_ini**
Bases: *cwatm.management_modules.dynamicModel.DynamicModel* (page 218)

CWATN initial part this part is to initialize the variables. It will call the initial part of the hydrological modules

i = 1

class cwatm.cwatm_initial.**Config**
Bases: object

class cwatm.cwatm_initial.**Variables**
Bases: object

load_initial(*name*, *default=0.0*, *number=None*)
First it is checked if the initial value is given in the settings file

- if it is <> None it is used directly
- if None it is loaded from the init netcdf file

Parameters

- **name** – Name of the init value
- **default** – default value -> default is 0.0
- **number** – in case of snow or runoff concentration several layers are included: number = no of the layer

Returns spatial map or value of initial condition

hydrological_modules package

Initialize

miscInitial module

Initializing some variables

class cwatm.hydrological_modules.miscInitial.**miscInitial**(*model*)
Bases: object

Miscellaneous repeatedly used expressions Definition if cell area comes from regular grid e.g. 5x5km or from irregular lat/lon Conversion factors between m3 and mm etc.

Note: Only used in the initial phase.

Global variables

Variable [self.var]	Description	Unit
DtSec	number of seconds per timestep (default = 86400)	s
twothird	2/3	–
MtoM3	Coefficient to change units	–
InvDtSec		
cellArea	Cell area [m^2] of each simulated mesh	
cellLength	length of a grid cell	m
InvCellArea	Inverse of cell area of each simulated mesh	m-1
DtDay	seconds in a timestep (default=86400)	s
InvDtDay	inverse seconds in a timestep (default=86400)	s-1
MMtoM	Coefficient to change units	–
MtoMM	Coefficient to change units	–
M3toM	Coefficient to change units	–
con_precipitation	conversion factor for precipitation	–
con_e	conversion factor for evaporation	–

Functions

initial()

Initialization of some basic parameters e.g. cellArea

- grid area, length definition
- conversion factors
- conversion factors for precipitation and pot evaporation

initcondition module

Load initial storage parameter maps

class `cwatm.hydrological_modules.initcondition.`**initcondition**(*model*)

Bases: `object`

READ/WRITE INITIAL CONDITIONS all initial condition can be stored at the end of a run to be used as a **warm** start for a following up run

Global variables

Variable [self.var]	Description	Unit
loadInit	Flag: if true initial conditions are loaded	–
initLoadFile	load file name of the initial condition data	–
saveInit	Flag: if true initial conditions are saved	–
saveInitFile	save file name of the initial condition data	–
coverTypes	land cover types - forest - grassland - irrPaddy - irrNonPaddy - water - sealed	–

Functions

dynamic()

Dynamic part of the initcondition module write initital conditions into a single netcdf file

Note: Several dates can be stored in different netcdf files

initial()

initial part of the initcondition module Puts all the variables which has to be stored in 2 lists:

- initCondVar: the name of the variable in the init netcdf file
- initCondVarValue: the variable as it can be read with the 'eval' command

Reads the parameter *save_initial* and *save_initial* to know if to save or load initial values

landcoverType module

Generate landcover types

class `cwatm.hydrological_modules.landcoverType.`**`landcoverType`**(*model*)

Bases: `object`

LAND COVER TYPE

runs the 6 land cover types through soil procedures

This routine calls the soil routine for each land cover type

Global variables

Variable [self.var]	Description	Unit
load_initial		
sum_gwRecharge	groundwater recharge	m
modflow	Flag: True if modflow_coupling = True in settings file	–
modflow_timestep	Chosen ModFlow model timestep (1day, 7days, 30days. . .)	
sumed_sum_gwRecharge		
GWVolumeVariation		
snowEvap	total evaporation from snow for a snow layers	m
maxGWCapRise	influence of capillary rise above groundwater level	m
minInterceptCap	Maximum interception read from file for forest and grassland land cover	m
interceptStor	simulated vegetation interception storage	m
dynamicLandcover		
landcoverSum		
act_SurfaceWaterAbst		
sum_interceptStor	Total of simulated vegetation interception storage including all landcover types	m
fracVegCover	Fraction of area covered by the corresponding landcover type	
minCropKC	minimum crop factor (default 0.2)	–
minTopWaterLayer		
rootFraction1		
maxRootDepth		
rootDepth		
KSat1		
KSat2		
KSat3		
alpha1		
alpha2		
alpha3		
lambda1		
lambda2		
lambda3		
thetas1		
thetas2		

Continued on next page

Table 1 – continued from previous page

Variable [self.var]	Description	Unit
thetas3		
thetar1		
thetar2		
thetar3		
genuM1		
genuM2		
genuM3		
genuInvM1		
genuInvM2		
genuInvM3		
genuInvN1		
genuInvN2		
genuInvN3		
invAlpha1		
invAlpha2		
invAlpha3		
ws1	Maximum storage capacity in layer 1	m
ws2	Maximum storage capacity in layer 2	m
ws3	Maximum storage capacity in layer 3	m
wres1	Residual storage capacity in layer 1	m
wres2	Residual storage capacity in layer 2	m
wres3	Residual storage capacity in layer 3	m
wrange1		
wrange2		
wrange3		
wfc1	Soil moisture at field capacity in layer 1	
wfc2	Soil moisture at field capacity in layer 2	
wfc3	Soil moisture at field capacity in layer 3	
wwp1	Soil moisture at wilting point in layer 1	
wwp2	Soil moisture at wilting point in layer 2	
wwp3	Soil moisture at wilting point in layer 3	
kUnSat3FC		
kunSatFC12		
kunSatFC23		
cropCoefficientNC_fi		
interceptCapNC_filen		
coverFractionNC_file		
sum_topwater	quantity of water on the soil (flooding) (weighted sum for all landcover types)	m
sum_soil		
sum_w1		
sum_w2		
sum_w3		
totalSto	Total soil,snow and vegetation storage for each cell including all landcover typ	m
arnoBetaOro		
arnoBeta		
adjRoot		
maxtopwater	maximum heigth of topwater	m
totAvlWater		
presumed_sum_gwRecha	Previous groundwater recharge [m/timestep] (used for the ModFlow version)	m

Continued on next page

Table 1 – continued from previous page

Variable [self.var]	Description	Unit
pretotalSto	Previous totalSto	m
sum_actBareSoilEvap		
sum_openWaterEvap		
addtoevapotrans		
sum_runoff	Runoff above the soil, more interflow, including all landcover types	m
sum_directRunoff		
sum_interflow		
sum_availWaterInfilt		
sum_capRiseFromGW	capillar rise from groundwater to 3rd soil layer (summed up for all land cover c	m
sum_act_irrConsumpti		
sum_perc3toGW	percolation from 3rd soil layer to groundwater (summed up for all land cover cla	m
sum_prefFlow	preferential flow from soil to groundwater (summed up for all land cover classes	m
act_irrWithdrawal		
act_nonIrrConsumptio		
returnFlow		
cellArea	Cell area [m^2] of each simulated mesh	
baseflow	simulated baseflow (= groundwater discharge to river)	m
Precipitation	Precipitation (input for the model)	m
coverTypes	land cover types - forest - grassland - irrPaddy - irrNonPaddy - water - sealed	–
Rain	Precipitation less snow	m
SnowMelt	total snow melt from all layers	m
SnowCover	snow cover (sum over all layers)	m
ElevationStD		
prevSnowCover	snow cover of previous day (only for water balance)	m
soilLayers	Number of soil layers	–
soildepth	Thickness of the first soil layer	m
soildepth12	Total thickness of layer 2 and 3	m
w1	Simulated water storage in the layer 1	m
w2	Simulated water storage in the layer 2	m
w3	Simulated water storage in the layer 3	m
topwater	quantity of water above the soil (flooding)	m
totalET	Total evapotranspiration for each cell including all landcover types	m
sum_actTransTotal		
sum_interceptEvap		

Functions

`dynamic()`

Dynamic part of the land cover type module

Calculating soil for each of the 6 land cover class

- calls evaporation_module.dynamic
- calls interception_module.dynamic
- calls soil_module.dynamic
- calls sealed_water_module.dynamic

And sums every thing up depending on the land cover type fraction

`dynamic_fracIrrigation`(*init=False*, *dynamic=True*)

Dynamic part of the land cover type module

Calculating fraction of land cover

- loads the fraction of landcover for each year from netcdf maps
- calculate the fraction of 6 land cover types based on the maps

Parameters

- **init** – (optional) True: set for the first time of a run
- **dynamic** – used in the dynmic run not in the initial phase

Returns

-

initial()

Initial part of the land cover type module Initialise the six land cover types

- Forest No.0
- Grasland/non irrigated land No.1
- Paddy irrigation No.2
- non-Paddy irrigation No.3
- Sealed area No.4
- Water covered area No.5

And initialize the soil variables

Hydrology I - from rain to soil

readmeteo module

Read meteorological input data

class cwatm.hydrological_modules.readmeteo.**readmeteo**(*model*)

Bases: object

READ METEOROLOGICAL DATA

reads all meteorological data from netcdf4 files

Global variables

Variable [self.var]	Description	Unit
modflow	Flag: True if modflow_coupling = True in settings file	–
modflowsteady	True if modflow_steadystate = True in settings file	–
DtDay	seconds in a timestep (default=86400)	s
con_precipitation	conversion factor for precipitation	–
con_e	conversion factor for evaporation	–
TMin	minimum air temperature	K
TMax	maximum air temperature	K
Psurf	Instantaneous surface pressure	Pa
Qair	specific humidity	kg/kg
Tavg	average air Temperature (input for the model)	K

Continued on next page

Table 2 – continued from previous page

Variable [self.var]	Description	Unit
Rsdl	long wave downward surface radiation fluxes	W/m2
Rsds	short wave downward surface radiation fluxes	W/m2
Wind	wind speed	m/s
ETRef	potential evapotranspiration rate from reference crop	m
EWRef	potential evaporation rate from water surface	m
Precipitation	Precipitation (input for the model)	m
meteomapsscale	if meteo maps have the same extend as the other spatial static maps -> meteomaps	–
meteodown	if meteo maps should be downscaled	–
preMaps	choose between steady state precipitation maps for steady state modflow or norma	–
tempMaps	choose between steady state temperature maps for steady state modflow or normal	–
evaTMaps	choose between steady state ETP water maps for steady state modflow or normal ma	–
eva0Maps	choose between steady state ETP reference maps for steady state modflow or norma	–
wc2_tavg	High resolution WorldClim map for average temperature	K
wc4_tavg	upscaled to low resolution WorldClim map for average temperature	K
wc2_tmin	High resolution WorldClim map for min temperature	K
wc4_tmin	upscaled to low resolution WorldClim map for min temperature	K
wc2_tmax	High resolution WorldClim map for max temperature	K
wc4_tmax	upscaled to low resolution WorldClim map for max temperature	K
wc2_prec	High resolution WorldClim map for precipitation	m
wc4_prec	upscaled to low resolution WorldClim map for precipitation	m
demAnomaly	digital elevation model anomaly (high resolution - low resolution)	m
demHigh	digital elevation model high resolution	m
prec	precipitation in m	m
temp	average temperature in Celsius deg	C°
Tmin	minimum temperature in Celsius deg	C°
Tmax	maximum temperature in celsius deg	C°
WtoMJ	Conversion factor from [W] to [MJ] for radiation: 86400 * 1E-6	–

Functions

downscaling2 (*input*, *downscaleName=''*, *wc2=0*, *wc4=0*, *downscale=0*)

Downscaling based on Delta method:

Note:

References

Moreno and Hasenauer 2015:

ftp://palantir.boku.ac.at/Public/ClimateData/Moreno_et_al-2015-International_Journal_of_Climatology.pdf

Mosier et al. 2018:

http://onlinelibrary.wiley.com/doi/10.1002/joc.5213/epdf

Parameters

- **input** – low input map
- **downscaleName** – High resolution monthly map from WorldClim
- **wc2** – High resolution WorldClim map
- **wc4** – upscaled to low resolution

- **downscale** – 0 for no change, 1: for temperature , 2 for pprecipitation, 3 for psurf

Returns input - downscaled input data

Returns wc2

Returns wc4

dynamic()

Dynamic part of the readmeteo module

Read meteo input maps from netcdf files

Note: If option *calc_evaporation* is False only precipitation, avg. temp., and 2 evaporation vlaues are read Otherwise all the variable needed for Penman-Monteith

Note: If option *TemperatureInKelvin* = True temperature is assumed to be Kelvin instead of Celsius!

initial()

Initial part of meteo

read multiple file of input

inflow module

Read river discharge time series as inflow data

class cwatm.hydrological_modules.inflow.**inflow**(*model*)

Bases: object

READ INFLOW HYDROGRAPHS (OPTIONAL) If option "inflow" is set to 1 the inflow hydrograph code is used otherwise dummy code is used

Global variables

Variable [self.var]	Description	Unit
sampleInflow	location of inflow point	lat/lon
noinflowpoints	number of inflow points	–
inflowTs	inflow time series data	m3/s
totalQInM3	total inflow over time (for mass balance calculation)	m3
inflowM3	inflow to basin	m3
DtSec	number of seconds per timestep (default = 86400)	s
QInM3Old	Inflow from previous day	m3

Functions

dynamic()

Dynamic part of the inflow module Use the inflow points to add inflow from time series file(s)

initial()

Initial part of the inflow module Get the inflow points

calls function hydrological_modules.getlocOutpoints() calls function hydrological_modules.join_struct_arrays2()

snow_frost module

Calculate snow and frost

class cwatm.hydrological_modules.snow_frost.**snow_frost**(*model*)

Bases: object

RAIN AND SNOW

Domain: snow calculations evaluated for center points of up to 7 sub-pixel snow zones 1 -7 which each occupy a part of the pixel surface

Variables *snow* and *rain* at end of this module are the pixel-average snowfall and rain

Global variables

Variable [self.var]	Description	Unit
load_initial		
DtDay	seconds in a timestep (default=86400)	s
Tavg	average air Temperature (input for the model)	K
Precipitation	Precipitation (input for the model)	m
Rain	Precipitation less snow	m
SnowMelt	total snow melt from all layers	m
SnowCover	snow cover (sum over all layers)	m
ElevationStD		
prevSnowCover	snow cover of previous day (only for water balance)	m
numberSnowLayersFloa		
numberSnowLayers	Number of snow layers (up to 10)	–
glaciertransportZone	Number of layers which can be mimiced as glacier transport zone	–
deltaInvNorm	Quantile of the normal distribution (for different numbers of snow layers)	–
DeltaTSnow	Temperature lapse rate x std. deviation of elevation	C°
SnowDayDegrees	day of the year to degrees: 360/365.25 = 0.9856	–
summerSeasonStart	day when summer season starts = 165	–
IceDayDegrees	days of summer (15th June-15th Sept.) to degree: 180/(259-165)	–
SnowSeason	seasonal melt factor	m C°-1 da
TempSnow	Average temperature at which snow melts	C°
SnowFactor	Multiplier applied to precipitation that falls as snow	–
SnowMeltCoef	Snow melt coefficient - default: 0.004	–
IceMeltCoef	Ice melt coefficnet - default 0.007	–
TempMelt	Average temperature at which snow melts	C°
SnowCoverS	snow cover for each layer	m
Kfrost	Snow depth reduction coefficient, (HH, p. 7.28)	m-1
Afrost	Daily decay coefficient, (Handbook of Hydrology, p. 7.28)	–
FrostIndexThreshold	Degree Days Frost Threshold (stops infiltration, percolation and capillary rise)	–
SnowWaterEquivalent	Snow water equivalent, (based on snow density of 450 kg/m3) (e.g. Tarboton and L	–
FrostIndex	FrostIndex - Molnau and Bissel (1983), A Continuous Frozen Ground Index for Floo	–
extfrostindex	Flag for second frostindex	–
FrostIndexThreshold2	FrostIndex2 - Molnau and Bissel (1983), A Continuous Frozen Ground Index for Flo	
frostInd1	forstindex 1	
frostInd2	frostindex 2	
frostindexS	array for frostindex	
Snow	Snow (equal to a part of Precipitation)	m

Functions

dynamic()

Dynamic part of the snow module

Distinguish between rain/snow and calculates snow melt and glacier melt The equation is a modification of:

References

Speers, D.D., Versteeg, J.D. (1979) Runoff forecasting for reservoir operations - the pastand the future. In: Proceedings 52nd Western Snow Conference, 149-156

Frost index in soil [degree days] based on:

References

Molnau and Bissel (1983, A Continuous Frozen Ground Index for Flood Forecasting. In: Maidment, Handbook of Hydrology, p. 7.28, 7.55)

Todo: calculate sinus shape function for the southern hemisspere

---

initial()

Initial part of the snow and frost module

- loads all the parameters for the day-degree approach for rain, snow and snowmelt
- loads the parameter for frost

evaporationPot module

Calculate potential Evaporation

class cwatm.hydrological_modules.evaporationPot.**evaporationPot**(*model*)

Bases: object

POTENTIAL REFERENCE EVAPO(TRANSPI)RATION Calculate potential evapotranspiration from climate data mainly based on FAO 56 and LISVAP Based on Penman Monteith

References

http://www.fao.org/docrep/X0490E/x0490e08.htm#penman%20monteith%20equation http://www.fao.org/docrep/X0490E/x0490e06.htm http://www.fao.org/docrep/X0490E/x0490e06.htm https://ec.europa.eu/jrc/en/publication/eur-scientific-and-technical-research-reports/lisvap-evaporation-pre-processor-lisflood-water-balance-and-flood-simulation-model

Global variables

Variable [self.var]	Description	Unit
cropCorrect	calibrated factor of crop KC factor	–
pet_modus	Flag: index which ETP approach is used e.g. 1 for Penman-Monteith	–
AlbedoCanopy	Albedo of vegetation canopy (FAO,1998) default =0.23	–
AlbedoSoil	Albedo of bare soil surface (Supit et. al. 1994) default = 0.15	–
AlbedoWater	Albedo of water surface (Supit et. al. 1994) default = 0.05	–
co2		
albedoLand	albedo from land surface (from GlobAlbedo database)	–
albedoOpenWater	albedo from open water surface (from GlobAlbedo database)	–
TMin	minimum air temperature	K
TMax	maximum air temperature	K
Psurf	Instantaneous surface pressure	Pa
Qair	specific humidity	kg/kg
Tavg	average air Temperature (input for the model)	K
Rsdl	long wave downward surface radiation fluxes	W/m2
Rsds	short wave downward surface radiation fluxes	W/m2
Wind	wind speed	m/s
ETRef	potential evapotranspiration rate from reference crop	m
EWRef	potential evaporation rate from water surface	m

Functions

dynamic()

Dynamic part of the potential evaporation module

Returns

- ETRef - potential reference evapotranspiration rate [m/day]
- EWRef - potential evaporation rate from water surface [m/day]

dynamic_1()

Dynamic part of the potential evaporation module Based on Penman Monteith - FAO 56

dynamic_2()

Dynamic part of the potential evaporation module 2: Milly and Dunne method P. C. D. Milly* and K. A. Dunne, 2016: Potential evapotranspiration and continental drying, Nature Climate Change, DOI: 10.1038/NCLIMATE3046 Energy only PET = 0.8(Rn â^'G) equation 8

dynamic_4()

Dynamic part of the potential evaporation module 4. Priestley-Taylor 1.26 * delat https://wetlandscapes.github.io/blog/blog/penman-monteith-and-priestley-taylor/

initial()

Initial part of evaporation type module Load inictial parameters

Note: Only run if *calc_evaporation* is True

initial_1()

Initial part of evaporation type module Load initial parameters 1: Penman Monteith 2: Milly and Dunne method P. C. D. Milly* and K. A. Dunne, 2016: Potential evapotranspiration and continental drying, Nature Climate Change, DOI: 10.1038/NCLIMATE3046 Energy only PET: ET=0.8(Rn â^'G) equation 8 3: Yang et al. Penman Montheith correction method Yang, Y., Roderick, M. L., Zhang, S., McVicar, T. R., and Donohue, R. J.: Hydrologic implications of vegetation response to elevated CO2 in climate projections, Nat. Clim. Change, 9, 44-48, 10.1038/s41558-018-0361-0, 2019. Equation 14: where the term accounts for changing [CO2] on rs

evaporation module

Calculate actual evapotranspiration

class `cwatm.hydrological_modules.evaporation.`**`evaporation`**(*model*)

Bases: `object`

Evaporation module Calculate potential evaporation and pot. transpiration

Global variables

Variable [self.var]	Description	Unit
cropKC	crop coefficient for each of the 4 different land cover types (forest, irrigated	–

Functions

`dynamic`(*coverType*, *No*)

Dynamic part of the soil module

calculating potential Evaporation for each land cover class with kc factor get crop coefficient, use potential ET, calculate potential bare soil evaporation and transpiration

Parameters

- **`coverType`** – Land cover type: forest, grassland . . .
- **`No`** – number of land cover type: forest = 0, grassland = 1 . . .

Returns potential evaporation from bare soil, potential transpiration

interception module

Calculate interception

class `cwatm.hydrological_modules.interception.`**`interception`**(*model*)

Bases: `object`

INTERCEPTION

Global variables

Variable [self.var]	Description	Unit
interceptCap	interception capacity of vegetation	m
availWaterInfiltrati	quantity of water reaching the soil after interception, more snowmelt	m
interceptEvap	simulated evaporation from water intercepted by vegetation	m
potTranspiration	Potential transpiration (after removing of evaporation)	m
snowEvap	total evaporation from snow for a snow layers	m
minInterceptCap	Maximum interception read from file for forest and grassland land cover	m
interceptStor	simulated vegetation interception storage	m
twothird	2/3	–
EWRef	potential evaporation rate from water surface	m
Rain	Precipitation less snow	m
SnowMelt	total snow melt from all layers	m
actualET	simulated evapotranspiration from soil, flooded area and vegetation	m

Functions

`dynamic` (*coverType*, *No*)
Dynamic part of the interception module calculating interception for each land cover class

Parameters

- **coverType** – Land cover type: forest, grassland . . .
- **No** – number of land cover type: forest = 0, grassland = 1 . . .

Returns interception evaporation, interception storage, reduced pot. transpiration

sealed_water module

Calculate water runoff from impermeable surface

class `cwatm.hydrological_modules.sealed_water.`**sealed_water**(*model*)
Bases: `object`

Sealed and open water runoff

calculated runoff from impermeable surface (sealed) and into water bodies

Global variables

Variable [self.var]	Description	Unit
modflow	Flag: True if modflow_coupling = True in settings file	–
availWaterInfiltrati	quantity of water reaching the soil after interception, more snowmelt	m
capillar	Simulated flow from groundwater to the third CWATM soil layer	m
EWRef	potential evaporation rate from water surface	m
actualET	simulated evapotranspiration from soil, flooded area and vegetation	m
directRunoff	Simulated surface runoff	m
openWaterEvap	Simulated evaporation from open areas	m
actTransTotal	Total actual transpiration from the three soil layers	m
actBareSoilEvap	Simulated evaporation from the first soil layer	m

Functions

`dynamic` (*coverType*, *No*)
Dynamic part of the sealed_water module

runoff calculation for open water and sealed areas

Parameters

- **coverType** – Land cover type: forest, grassland . . .
- **No** – number of land cover type: forest = 0, grassland = 1 . . .

Hydrology II - from soil to river

soil module

Calculate fluxes in 3 layer soil

class `cwatm.hydrological_modules.soil.`**`soil`**(*model*)

Bases: `object`

SOIL

Calculation vertical transfer of water based on Arno scheme

Global variables

Variable [self.var]	Description	Unit
capRiseFrac	fraction of a grid cell where capillar rise may happen	m
cropKC	crop coefficient for each of the 4 different land cover types (forest, irrigated	–
storGroundwater	simulated groundwater storage	m
modflow	Flag: True if modflow_coupling = True in settings file	–
availWaterInfiltrati	quantity of water reaching the soil after interception, more snowmelt	m
interceptEvap	simulated evaporation from water intercepted by vegetation	m
potTranspiration	Potential transpiration (after removing of evaporation)	m
snowEvap	total evaporation from snow for a snow layers	m
fracVegCover	Fraction of area covered by the corresponding landcover type	
rootDepth		
KSat1		
KSat2		
KSat3		
genuM1		
genuM2		
genuM3		
genuInvM1		
genuInvM2		
genuInvM3		
ws1	Maximum storage capacity in layer 1	m
ws2	Maximum storage capacity in layer 2	m
ws3	Maximum storage capacity in layer 3	m
wres1	Residual storage capacity in layer 1	m
wres2	Residual storage capacity in layer 2	m
wres3	Residual storage capacity in layer 3	m
wrange1		
wrange2		
wrange3		
wfc1	Soil moisture at field capacity in layer 1	
wfc2	Soil moisture at field capacity in layer 2	
wfc3	Soil moisture at field capacity in layer 3	
wwp1	Soil moisture at wilting point in layer 1	
wwp2	Soil moisture at wilting point in layer 2	
wwp3	Soil moisture at wilting point in layer 3	
kunSatFC12		
kunSatFC23		
arnoBeta		
adjRoot		
maxtopwater	maximum heigth of topwater	m
capillar	Simulated flow from groundwater to the third CWATM soil layer	m
EWRef	potential evaporation rate from water surface	m
FrostIndexThreshold	Degree Days Frost Threshold (stops infiltration, percolation and capillary rise)	–
FrostIndex	FrostIndex - Molnau and Bissel (1983), A Continuous Frozen Ground Index for Floo	–

Continued on next page

Table 4 – continued from previous page

Variable [self.var]	Description	Unit
actualET	simulated evapotranspiration from soil, flooded area and vegetation	m
soilLayers	Number of soil layers	–
soildepth	Thickness of the first soil layer	m
soildepth12	Total thickness of layer 2 and 3	m
w1	Simulated water storage in the layer 1	m
w2	Simulated water storage in the layer 2	m
w3	Simulated water storage in the layer 3	m
topwater	quantity of water above the soil (flooding)	m
directRunoff	Simulated surface runoff	m
interflow	Simulated flow reaching runoff instead of groundwater	m
openWaterEvap	Simulated evaporation from open areas	m
actTransTotal	Total actual transpiration from the three soil layers	m
actBareSoilEvap	Simulated evaporation from the first soil layer	m
percolationImp	Fraction of area covered by the corresponding landcover type	m
cropGroupNumber	soil water depletion fraction, Van Diepen et al., 1988: WOFOST 6.0, p.86, Dooren	–
cPrefFlow	Factor influencing preferential flow (flow from surface to GW)	–
act_irrConsumption	actual irrgation water consumption	m
potBareSoilEvap	potential bare soil evaporation (calculated with minus snow evaporation)	m
totalPotET	Potential evaporation per land use class	m
rws	Transpiration reduction factor (in case of water stress)	–
prefFlow	Flow going directly from rainfall to groundwater	m
infiltration	Water actually infiltrating the soil	m
capRiseFromGW	Simulated capillary rise from groundwater	m
NoSubSteps	Number of sub steps to calculate soil percolation	–
perc1to2	Simulated water flow from soil layer 1 to soil layer 2	m
perc2to3	Simulated water flow from soil layer 2 to soil layer 3	m
perc3toGW	Simulated water flow from soil layer 3 to groundwater	m
theta1	fraction of water in soil compartment 1 for each land use class	–
theta2	fraction of water in soil compartment 2 for each land use class	–
theta3	fraction of water in soil compartment 3 for each land use class	–
actTransTotal_forest		
actTransTotal_grassl		
actTransTotal_paddy		
actTransTotal_nonpad		
before		
gwRecharge	groundwater recharge	m

Functions

dynamic(*coverType*, *No*)

Dynamic part of the soil module

For each of the land cover classes the vertical water transport is simulated Distribution of water holding capiacity in 3 soil layers based on saturation excess overland flow, preferential flow Dependend on soil depth, soil hydraulic parameters

initial()

Initial part of the soil module

- Initialize all the hydraulic properties of soil
- Set soil depth

capillarRise module

Calculate capillar rise from groundwater

class cwatm.hydrological_modules.capillarRise.**capillarRise**(*model*)

Bases: object

CAPILLAR RISE calculate cell fraction influenced by capillary rise

Global variables

Variable [self.var]	Description	Unit
dzRel0100	map of relative elevation above flood plains (max elevation above plain)	m
dzRel0090	map of relative elevation above flood plains (90% elevation above plain)	m
dzRel0080	map of relative elevation above flood plains (80% elevation above plain)	m
dzRel0070	map of relative elevation above flood plains (70% elevation above plain)	m
dzRel0060	map of relative elevation above flood plains (60% elevation above plain)	m
dzRel0050	map of relative elevation above flood plains (median elevation above plain)	m
dzRel0040	map of relative elevation above flood plains (40% elevation above plain)	m
dzRel0030	map of relative elevation above flood plains (30% elevation above plain)	m
dzRel0020	map of relative elevation above flood plains (20% elevation above plain)	m
dzRel0010	map of relative elevation above flood plains (10% elevation above plain)	m
dzRel0005	map of relative elevation above flood plains (5% elevation above plain)	m
dzRel0001	map of relative elevation above flood plains (1% elevation above plain)	m
capRiseFrac	fraction of a grid cell where capillar rise may happen	m
storGroundwater	simulated groundwater storage	m
specificYield	groundwater reservoir parameters (if ModFlow is not used) used to compute ground	m
modflow	Flag: True if modflow_coupling = True in settings file	–
maxGWCapRise	influence of capillary rise above groundwater level	m

Functions

dynamic()

Dynamic part of the capillar Rise module calculate cell fraction influenced by capillary rise depending on appr. height of groundwater and relative elevation of grid cell

Returns capRiseFrac = cell fraction influenced by capillary rise

groundwater module

Calculate groundwater

class cwatm.hydrological_modules.groundwater.**groundwater**(*model*)

Bases: object

GROUNDWATER

Global variables

Variable [self.var]	Description	Unit
storGroundwater	simulated groundwater storage	m
specificYield	groundwater reservoir parameters (if ModFlow is not used) used to compute ground	m
recessionCoeff	groundwater storage times this coefficient gives baseflow	–
kSatAquifer	groundwater reservoir parameters (if ModFlow is not used), could be used to comp	m day-1
load_initial		
readAvlStorGroundwat	same as storGroundwater but equal to 0 when inferior to a treshold	m
pumping_actual		
prestorGroundwater	storGroundwater at the beginning of each step	m
sum_gwRecharge	groundwater recharge	m
modflow	Flag: True if modflow_coupling = True in settings file	–
gwstore		
capillar	Simulated flow from groundwater to the third CWATM soil layer	m
baseflow	simulated baseflow (= groundwater discharge to river)	m
nonFossilGroundwater	groundwater abstraction which is sustainable and not using fossil resources	m

Functions

dynamic()

Dynamic part of the groundwater module Calculate groundweater storage and baseflow

initial()

Initial part of the groundwater module

- load parameters from settings file
- initial groundwater storage

runoff_concentration module

Calculate runoff concentration - from grid cell to grid cell corner

class cwatm.hydrological_modules.runoff_concentration.**runoff_concentration**(*model*)

Bases: object

Runoff concentration

this is the part between runoff generation and routing for each gridcell and for each land cover class the generated runoff is concentrated at a corner of a gridcell this concentration needs some lag-time (and peak time) and leads to diffusion lag-time/ peak time is calculated using slope, length and land cover class diffusion is calculated using a triangular-weighting-function

$Q(t) = \sum_{i=0}^{max} c(i) * Q_{\mathrm{GW}}(t - i + 1)$

where $c(i) = \int_{i-1}^{i} \frac{2}{max} - |u - \frac{max}{2}| * \frac{4}{max^2} du$

see also:

http://stackoverflow.com/questions/24040984/transformation-using-triangular-weighting-function-in-python

Global variables

Variable [self.var]	Description	Unit
load_initial		
fracVegCover	Fraction of area covered by the corresponding landcover type	
sum_interflow		
baseflow	simulated baseflow (= groundwater discharge to river)	m
coverTypes	land cover types - forest - grassland - irrPaddy - irrNonPaddy - water - sealed	–
runoff		
runoff_peak	peak time of runoff in seconds for each land use class	s
tpeak_interflow	peak time of interflow	s
tpeak_baseflow	peak time of baseflow	s
maxtime_runoff_conc	maximum time till all flow is at the outlet	s
runoff_conc	runoff after concentration - triangular-weighting method	m
gridcell_storage		
sum_landSurfaceRunof	Runoff concentration above the soil more interflow including all landcover types	m
landSurfaceRunoff	Runoff concentration above the soil more interflow	m
directRunoff	Simulated surface runoff	m
interflow	Simulated flow reaching runoff instead of groundwater	m
prergridcell		

Functions

dynamic()

Dynamic part of the runoff concentration module

For surface runoff for each land cover class and for interflow and for baseflow the runoff concentration time is calculated

Note: the time demanding part is calculated in a c++ library

initial()

Initial part of the runoff concentration module

Setting the peak time for:

- surface runoff = 3
- interflow = 4
- baseflow = 5

based on the slope the concentration time for each land cover type is calculated

Note: only if option **includeRunoffConcentration** is TRUE

Hydrology III - Socio-economic - Water demand

water_demand.water_demand

Calculate water demand from different sectors

class cwatm.hydrological_modules.water_demand.water_demand.**water_demand**(*model*)

Bases: object

WATERDEMAND

calculating water demand - Industrial, domenstic based on precalculated maps Agricultural water demand based on water need by plants

Global variables

Variable [self.var]	Description	Unit
readAvlStorGroundwat	same as storGroundwater but equal to 0 when inferior to a treshold	m
nonFossilGroundwater	groundwater abstraction which is sustainable and not using fossil resources	m
waterbalance_module		
waterBodyID	lakes/reservoirs map with a single ID for each lake/reservoir	–
compress_LR	boolean map as mask map for compressing lake/reservoir	–
decompress_LR	boolean map as mask map for decompressing lake/reservoir	–
MtoM3C	conversion factor from m to m3 (compressed map)	–
MtoM3	Coefficient to change units	–
lakeVolumeM3C	compressed map of lake volume	m3
lakeStorageC		m3
reservoirStorageM3C		
lakeResStorageC		
lakeResStorage		
waterBodyTypCTemp		
InvDtSec		
cellArea	Cell area [m^2] of each simulated mesh	
smalllakeVolumeM3		
smalllakeStorage		
act_SurfaceWaterAbst		
fracVegCover	Fraction of area covered by the corresponding landcover type	
addtoevapotrans		
M3toM	Coefficient to change units	–
act_irrConsumption	actual irrgation water consumption	m
channelStorage		
act_bigLakeResAbst		
act_smallLakeResAbst		
returnFlow		
modflowPumpingM		
modflowTopography		
modflowDepth2		
leakageC		
domesticDemand		
pot_domesticConsumpt		
dom_efficiency		
demand_unit		
envFlow		
industryDemand		
pot_industryConsumpt		
ind_efficiency		
unmetDemandPaddy		
unmetDemandNonpaddy		
unmetDemand		
efficiencyPaddy		
efficiencyNonpaddy		

Continued on next page

Table 5 – continued from previous page

Variable [self.var]	Description	Unit
returnfractionIrr		
irrDemand		
totalIrrDemand		
livestockDemand		
pot_livestockConsump		
liv_efficiency		
allocSegments		
swAbstractionFractio		
modflowPumping		
leakage		
pumping		
nonIrrReturnFlowFrac		
nonIrruse		
act_indDemand		
act_domDemand		
act_livDemand		
nonIrrDemand		
totalWaterDemand		
act_irrWithdrawal		
act_nonIrrWithdrawal		
act_totalWaterWithdr		
act_indConsumption		
act_domConsumption		
act_livConsumption		
act_nonIrrConsumptio		
act_totalIrrConsumpt		
act_totalWaterConsum		
returnflowIrr		
pot_nonIrrConsumptio		
readAvlChannelStorag		
reservoir_command_ar		
leakageC_daily		
leakageC_daily_segme		
pot_GroundwaterAbstr		
renewableAvlWater		
act_irrNonpaddyWithd		
act_irrPaddyWithdraw		
act_irrPaddyDemand		
act_irrNonpaddyDeman		
act_indWithdrawal		
act_domWithdrawal		
act_livWithdrawal		
waterDemandLost		

Functions

dynamic()

Dynamic part of the water demand module

- calculate the fraction of water from surface water vs. groundwater
- get non-Irrigation water demand and its return flow fraction

initial()
Initial part of the water demand module

Set the water allocation

water_demand.domestic

Read water demand for domestic

class cwatm.hydrological_modules.water_demand.domestic.**waterdemand_domestic**(*model*)
Bases: object

WATERDEMAND domestic

calculating water demand - domenstic based on precalculated maps

Global variables

Variable [self.var]	Description	Unit
InvCellArea	Inverse of cell area of each simulated mesh	m-1
M3toM	Coefficient to change units	–
domesticTime		
domWithdrawalVar		
domConsumptionVar		
domesticDemand		
pot_domesticConsumpt		
dom_efficiency		
demand_unit		

Functions

dynamic()
Dynamic part of the water demand module - domestic read monthly (or yearly) water demand from netcdf and transform (if necessary) to [m/day]

initial()
Initial part of the water demand module

water_demand.industry

Read water demand for industry

class cwatm.hydrological_modules.water_demand.industry.**waterdemand_industry**(*model*)
Bases: object

WATERDEMAND domestic

calculating water demand - industry based on precalculated maps

Global variables

Variable [self.var]	Description	Unit
InvCellArea	Inverse of cell area of each simulated mesh	m-1
M3toM	Coefficient to change units	–
demand_unit		
industryTime		
indWithdrawalVar		
indConsumptionVar		
industryDemand		
pot_industryConsumpt		
ind_efficiency		

Functions

dynamic()
Dynamic part of the water demand module - industry read monthly (or yearly) water demand from netcdf and transform (if necessary) to [m/day]

initial()
Initial part of the water demand module - industry

water_demand.livestock

Read water demand for livestock

class cwatm.hydrological_modules.water_demand.livestock.**waterdemand_livestock**(*model*)
Bases: object

WATERDEMAND livestock

calculating water demand - livestock based on precalculated maps

Global variables

Variable [self.var]	Description	Unit
InvCellArea	Inverse of cell area of each simulated mesh	m-1
M3toM	Coefficient to change units	–
domesticTime		
demand_unit		
livestockTime		
livVar		
uselivestock		
livestockDemand		
pot_livestockConsump		
liv_efficiency		

Functions

dynamic()
Dynamic part of the water demand module - livestock read monthly (or yearly) water demand from netcdf and transform (if necessary) to [m/day]

initial()
Initial part of the water demand module - livestock

water_demand.irrigation

Calculate water demand for irrigation

class cwatm.hydrological_modules.water_demand.irrigation.**waterdemand_irrigation**(*model*)

Bases: object

WATERDEMAND

calculating water demand - irrigation Agricultural water demand based on water need by plants

Global variables

Variable [self.var]	Description	Unit
cropKC	crop coefficient for each of the 4 different land cover types (forest, irrigated	–
load_initial		
availWaterInfiltrati	quantity of water reaching the soil after interception, more snowmelt	m
fracVegCover	Fraction of area covered by the corresponding landcover type	
ws1	Maximum storage capacity in layer 1	m
ws2	Maximum storage capacity in layer 2	m
wfc1	Soil moisture at field capacity in layer 1	
wfc2	Soil moisture at field capacity in layer 2	
wwp1	Soil moisture at wilting point in layer 1	
wwp2	Soil moisture at wilting point in layer 2	
w1	Simulated water storage in the layer 1	m
w2	Simulated water storage in the layer 2	m
topwater	quantity of water above the soil (flooding)	m
arnoBeta		
maxtopwater	maximum heigth of topwater	m
totAvlWater		
InvCellArea	Inverse of cell area of each simulated mesh	m-1
totalPotET	Potential evaporation per land use class	m
unmetDemandPaddy		
unmetDemandNonpaddy		
unmetDemand		
efficiencyPaddy		
efficiencyNonpaddy		
returnfractionIrr		
alphaDepletion		
pot_irrConsumption		
irrDemand		
totalIrrDemand		

Functions

dynamic()

Dynamic part of the water demand module

- calculate the fraction of water from surface water vs. groundwater
- get non-Irrigation water demand and its return flow fraction

initial()

Initial part of the water demand module irrigation

water_demand.environmental_need

Read water for environmental need

class `cwatm.hydrological_modules.water_demand.environmental_need.`**`waterdemand_environmental_`**

Bases: `object`

WATERDEMAND environment_need

calculating water demand - environmental need based on precalculated maps done before in CWatM

Global variables

Variable [self.var]	Description	Unit
cut_ef_map	if TRUE calculated maps of environmental flow are cut to the extend of the area	–
M3toM	Coefficient to change units	–
chanLength		
channelAlpha		
use_environflow		
envFlowm3s		
envFlow		

Functions

`dynamic``()`

Dynamic part of the water demand module - environment read monthly (or yearly) water demand from netcdf and transform (if necessary) to [m/day]

`initial``()`

Initial part of the water demand module - environment

Hydrology IV - Lakes, reservoirs and river

lakes_reservoirs module

Calculate water retention in lakes

class `cwatm.hydrological_modules.lakes_reservoirs.`**`lakes_reservoirs`**`(`*model*`)`

Bases: `object`

LAKES AND RESERVOIRS

Note: Calculate water retention in lakes and reservoirs

Using the **Modified Puls approach** to calculate retention of a lake See also: LISFLOOD manual Annex 3 (Burek et al. 2013)

for Modified Puls Method the Q(inflow)1 has to be used. It is assumed that this is the same as Q(inflow)2 for the first timestep has to be checked if this works in forecasting mode!

Lake Routine using Modified Puls Method (see Maniak, p.331ff)

$$\frac{Qin1 + Qin2}{2} - \frac{Qout1 + Qout2}{2} = \frac{S2 - S1}{\delta time}$$

changed into:

$$\frac{S2}{time + Qout2/2} = \frac{S1}{dtime + Qout1/2} - Qout1 + \frac{Qin1 + Qin2}{2}$$

Outgoing discharge (Qout) are linked to storage (S) by elevation.

Now some assumption to make life easier:

1.) storage volume is increase proportional to elevation: S = A * H where: H: elevation, A: area of lake

2.) $Q_{\text{out}} = c * b * H^{2.0}$ (c: weir constant, b: width)

> 2.0 because it fits to a parabolic cross section see (Aigner 2008) (and it is much easier to calculate (that's the main reason)

c: for a perfect weir with mu=0.577 and Poleni: $\frac{2}{3}\mu * \sqrt{2 * g} = 1.7$

c: for a parabolic weir: around 1.8

because it is a imperfect weir: $C = c * 0.85 = 1.5$

results in formular: $Q = 1.5 * b * H^2 = a * H^2 - > H = \sqrt{Q/a}$

Solving the equation:

$\frac{S2}{dtime + Qout2/2} = \frac{S1}{dtime + Qout1/2} - Qout1 + \frac{Qin1 + Qin2}{2}$

$SI = \frac{S2}{dtime} + \frac{Qout2}{2} = \frac{A*H}{DtRouting} + \frac{Q}{2} = \frac{A}{DtRouting*\sqrt{a}*\sqrt{Q}} + \frac{Q}{2}$

-> replacement: $\frac{A}{DtSec*\sqrt{a}} = Lakefactor, Y = \sqrt{Q}$

$Y^2 + 2 * Lakefactor * Y - 2 * SI = 0$

solution of this quadratic equation:

$Q = (-LakeFactor + \sqrt{LakeFactor^2 + 2 * SI})^2$

Global variables

Variable [self.var]	Description	Unit
load_initial		
saveInit	Flag: if true initial conditions are saved	–
waterBodyID	lakes/reservoirs map with a single ID for each lake/reservoir	–
waterBodyOut	biggest outlet (biggest accumulation of ldd network) of a waterbody	–
dirUp	river network in upstream direction	–
ldd_LR	change river network (put pits in where lakes are)	–
lddCompress_LR	compressed river network lakes/reservoirs (without missing values)	–
dirUp_LR	river network direction upstream lake/reservoirs	–
dirupLen_LR	number of bifurcation upstream lake/reservoir	–
dirupID_LR	index river upstream lake/reservoir	–
downstruct_LR	river network downstream lake/reservoir	–
catchment_LR	catchments lake/reservoir	–
dirDown_LR	river network direktion downstream lake/reservoir	–
lendirDown_LR	number of river network connections lake/reservoir	–
compress_LR	boolean map as mask map for compressing lake/reservoir	–
decompress_LR	boolean map as mask map for decompressing lake/reservoir	–
waterBodyOutC	compressed map biggest outlet of each lake/reservoir	–
resYearC	compressed map of the year when the reservoirs is operating	–

Continued on next page

Table 6 – continued from previous page

Variable [self.var]	Description	Unit
waterBodyTypC	water body types 3 reservoirs and lakes (used as reservoirs but before the year	–
lakeArea	area of each lake/reservoir	m2
lakeAreaC	compressed map of the area of each lake/reservoir	m2
lakeDis0	compressed map average discharge at the outlet of a lake/reservoir	m3 s-1
lakeDis0C	average discharge at the outlet of a lake/reservoir	m3 s-1
lakeAC	compressed map of parameter of channel width, gravity and weir coefficient	–
resVolumeC	compressed map of reservoir volume	Million m
lakeEvaFactorC	compressed map of a factor which increases evaporation from lake because of wind	–
reslakeoutflow		m
lakeVolume	volume of lakes	m3
outLake	outflow from lakes	m
lakeStorage		
lakeInflow		
lakeOutflow		
reservoirStorage		
MtoM3C	conversion factor from m to m3 (compressed map)	–
EvapWaterBodyM		
lakeResInflowM		
lakeResOutflowM		
lakedaycorrect		
lakeFactor	factor for the Modified Puls approach to calculate retention of the lake	–
lakeFactorSqr	square root factor for the Modified Puls approach to calculate retention of the	–
lakeInflowOldC	inflow to the lake from previous days	m/3
lakeVolumeM3C	compressed map of lake volume	m3
lakeStorageC		m3
lakeOutflowC	compressed map of lake outflow	m3/s
lakeLevelC	compressed map of lake level	m
conLimitC		
normLimitC		
floodLimitC		
adjust_Normal_FloodC		
norm_floodLimitC		
minQC		
normQC		
nondmgQC		
deltaO		
deltaLN		
deltaLF		
deltaNFL		
reservoirFillC		
reservoirStorageM3C		
lakeResStorageC		
lakeResStorage		
resStorage		
waterBodyTypCTemp		
sumEvapWaterBodyC		
sumlakeResInflow		
sumlakeResOutflow		
lakeIn		

Continued on next page

Table 6 – continued from previous page

Variable [self.var]	Description	Unit
lakeEvapWaterBodyC		
resEvapWaterBodyC		
downstruct		
DtSec	number of seconds per timestep (default = 86400)	s
MtoM3	Coefficient to change units	–
InvDtSec		
cellArea	Cell area [m^2] of each simulated mesh	
UpArea1	upstream area of a grid cell	m2
lddCompress	compressed river network (without missing values)	–
lakeEvaFactor	a factor which increases evaporation from lake because of wind	–
dtRouting	number of seconds per routing timestep	s
evapWaterBodyC		
sumLakeEvapWaterBody		
noRoutingSteps		
sumResEvapWaterBodyC		
discharge	discharge	m3/s
prelakeResStorage		
runoff		

Functions

`dynamic``()`
Dynamic part set lakes and reservoirs for each year

`dynamic_inloop``(`*NoRoutingExecuted*`)`
Dynamic part to calculate outflow from lakes and reservoirs

- lakes with modified Puls approach
- reservoirs with special filling levels

Parameters **`NoRoutingExecuted`** – actual number of routing substep

Returns outLdd: outflow in m3 to the network

Note: outflow to adjected lakes and reservoirs is calculated separately

---

`initWaterbodies``()`
Initialize water bodies Read parameters from maps e.g area, location, initial average discharge, type 9reservoir or lake) etc.

Compress numpy array from mask map to the size of lakes+reservoirs (marked as capital C at the end of the variable name)

`initial_lakes``()`
Initial part of the lakes module Using the **Modified Puls approach** to calculate retention of a lake

`initial_reservoirs``()`
Initial part of the reservoir module Using the appraoch of LISFLOOD

See also:

LISFLOOD manual Annex 1: (Burek et al. 2013)

lakes_res_small module

Calculate water retention in small lakes

class `cwatm.hydrological_modules.lakes_res_small.`**`lakes_res_small`**(*model*)

Bases: `object`

Small LAKES AND RESERVOIRS

Note: Calculate water retention in lakes and reservoirs

Using the **Modified Puls approach** to calculate retention of a lake See also: LISFLOOD manual Annex 3 (Burek et al. 2013)

Global variables

Variable [self.var]	Description	Unit
load_initial		
smallpart		
smalllakeArea		
smalllakeDis0		
smalllakeA		
smalllakeFactor		
smalllakeFactorSqr		
smalllakeInflowOld		
smalllakeVolumeM3		
smalllakeOutflow		
smalllakeLevel		
smalllakeStorage		
minsmalllakeVolumeM3		
preSmalllakeStorage		
smallLakedaycorrect		
smallLakeIn		
smallevapWaterBody		
smallLakeout		
smallrunoffDiff		
DtSec	number of seconds per timestep (default = 86400)	s
InvDtSec		
cellArea	Cell area [m^2] of each simulated mesh	
EWRef	potential evaporation rate from water surface	m
lakeEvaFactor	a factor which increases evaporation from lake because of wind	–
runoff		

Functions

`dynamic`()

Dynamic part to calculate outflow from small lakes and reservoirs

- lakes with modified Puls approach
- reservoirs with special filling levels

Flow out of lake:

Returns outflow in m3 to the network

initial()

Initialize small lakes and reservoirs Read parameters from maps e.g area, location, initial average discharge, type: reservoir or lake) etc.

routing_reservoirs.routing_kinematic module

River routing - kinematic wave

class cwatm.hydrological_modules.routing_reservoirs.routing_kinematic.**routing_kinematic**(*mod*

Bases: object

ROUTING

routing using the kinematic wave

Global variables

Variable [self.var]	Description	Unit
load_initial		
inflowM3	inflow to basin	m3
waterBodyID	lakes/reservoirs map with a single ID for each lake/reservoir	–
dirUp	river network in upstream direction	–
dirupLen_LR	number of bifurcation upstream lake/reservoir	–
dirupID_LR	index river upstream lake/reservoir	–
dirDown_LR	river network direktion downstream lake/reservoir	–
lendirDown_LR	number of river network connections lake/reservoir	–
compress_LR	boolean map as mask map for compressing lake/reservoir	–
lakeArea	area of each lake/reservoir	m2
lakeEvaFactorC	compressed map of a factor which increases evaporation from lake because of wind	–
EvapWaterBodyM		
lakeResInflowM		
lakeResOutflowM		
lakeResStorage		
downstruct		
act_SurfaceWaterAbst		
fracVegCover	Fraction of area covered by the corresponding landcover type	
returnFlow		
DtSec	number of seconds per timestep (default = 86400)	s
cellArea	Cell area [m^2] of each simulated mesh	
EWRef	potential evaporation rate from water surface	m
QInM3Old	Inflow from previous day	m3
UpArea1	upstream area of a grid cell	m2
lddCompress	compressed river network (without missing values)	–
lakeEvaFactor	a factor which increases evaporation from lake because of wind	–
dtRouting	number of seconds per routing timestep	s
evapWaterBodyC		
sumLakeEvapWaterBody		
noRoutingSteps		
sumResEvapWaterBodyC		
discharge	discharge	m3/s
prelakeResStorage		
catchmentAll		

Continued on next page

Table 7 – continued from previous page

Variable [self.var]	Description	Unit
sumsideflow		
EvapoChannel		
prechannelStorage		
channelStorage		
chanLength		
totalCrossSectionAre		
dirupLen		
dirupID		
catchment		
dirDown		
lendirDown		
UpArea		
beta		
chanMan		
chanGrad		
chanWidth		
chanDepth		
invbeta		
invchanLength		
invdtRouting		
totalCrossSectionAre		
chanWettedPerimeterA		
alpPower		
channelAlpha		
invchannelAlpha		
riverbedExchange		
QDelta		
act_bigLakeResAbst		
act_smallLakeResAbst		
inflowDt		
dis_outlet		
runoff		
openWaterEvap	Simulated evaporation from open areas	m

Functions

`catchment`(*point*)

Get the catchment from "global" LDD and a point

- load and create a river network
- calculate catchment upstream of point

`dynamic`()

Dynamic part of the routing module

- calculate evaporation from channels
- calculate riverbed exchange between riverbed and groundwater
- if option **waterbodies** is true, calculate retention from water bodies
- calculate sideflow -> inflow to river
- calculate kinematic wave -> using C++ library for computational speed

initial()

Initial part of the routing module

- load and create a river network
- calculate river network parameter e.g. river length, width, depth, gradient etc.
- calculate initial filling
- calculate manning's roughness coefficient

routing_reservoirs.routing_sub module

Sub routines for river routing

cwatm.hydrological_modules.routing_reservoirs.routing_sub.**Compress**(*map*, *mask*)

compressing map from 2D to 1D without missing values

Parameters

- **map** – input map
- **mask** – mask map

Returns compressed map

cwatm.hydrological_modules.routing_reservoirs.routing_sub.**catchment1**(*dirUp*, *points*)

calculates all cells which belongs to a catchment from point onward

Parameters

- **dirUp** –
- **points** –

Returns subcatchment

cwatm.hydrological_modules.routing_reservoirs.routing_sub.**decompress1**(*map*)

Decompressing map from 1D to 2D with missing values

Parameters **map** – compressed map

Returns decompressed 2D map

cwatm.hydrological_modules.routing_reservoirs.routing_sub.**defLdd2**(*ldd*)

defines river network

Parameters **ldd** – river network

Returns ldd variables

cwatm.hydrological_modules.routing_reservoirs.routing_sub.**dirDownstream**(*dirUp*, *lddcomp*, *dirDown*)

runs the river network tree downstream - from source to outlet

Parameters

- **dirUp** –
- **lddcomp** –

- **dirDown** –

Returns direction downstream

`cwatm.hydrological_modules.routing_reservoirs.routing_sub.`**dirUpstream**(*dirshort*)

runs the network tree upstream from outlet to source

Parameters **dirshort** –

Returns direction upstream

`cwatm.hydrological_modules.routing_reservoirs.routing_sub.`**downstream1**(*dirUp*, *weight*)

calculated 1 cell downstream

Parameters

- **dirUp** –
- **weight** –

Returns dowmnstream 1 cell

`cwatm.hydrological_modules.routing_reservoirs.routing_sub.`**lddrepair**(*lddnp*, *lddOrder*)

repairs a river network

- eliminate unsound parts
- add pits at points with no connections

Parameters

- **lddnp** – rivernetwork as 1D array
- **lddOrder** –

Returns repaired ldd

`cwatm.hydrological_modules.routing_reservoirs.routing_sub.`**lddshort**(*lddnp*, *lddOrder*)

return short for calculating a catchment from a river network

Parameters

- **lddnp** – rivernetwork as 1D array
- **lddOrder** –

Returns short ldd

`cwatm.hydrological_modules.routing_reservoirs.routing_sub.`**postorder**(*dirUp*, *catchment*, *node*, *catch*, *dirDown*)

Routine to run a postorder tree traversal

Parameters

- **dirUp** –
- **catchment** –
- **node** –

- **catch** –
- **dirDown** –

Returns dirDown and catchment

`cwatm.hydrological_modules.routing_reservoirs.routing_sub.`**`subcatchment1`**(*dirUp*, *points*, *ups*)

calculates subcatchments of points

Parameters

- **dirUp** –
- **points** –
- **ups** –

Returns subcatchment

`cwatm.hydrological_modules.routing_reservoirs.routing_sub.`**`upstream1`**(*downstruct*, *weight*)

Calculates 1 cell upstream

Parameters

- **downstruct** –
- **weight** –

Returns upstream 1cell

`cwatm.hydrological_modules.routing_reservoirs.routing_sub.`**`upstreamArea`**(*dirDown*, *dirshort*, *area*)

calculates upstream area

Parameters

- **dirDown** – array which point from each cell to the next downstream cell
- **dirshort** –
- **area** – area in m2 for a single gridcell

Returns upstream area

Hydrology V - Water balance

waterbalance module

class `cwatm.hydrological_modules.waterbalance.`**`waterbalance`**(*model*)

Bases: `object`

WATER BALANCE

- check if water balnace per time step is ok (= 0)
- produce an annual overview - income, outcome storage

Global variables

Variable [self.var]	Description	Unit
storGroundwater	simulated groundwater storage	m
prestorGroundwater	storGroundwater at the beginning of each step	m
sum_gwRecharge	groundwater recharge	m
snowEvap	total evaporation from snow for a snow layers	m
smalllakeStorage		
preSmalllakeStorage		
smallLakedaycorrect		
smallLakeIn		
smallevapWaterBody		
smallLakeout		
lakeStorage		
EvapWaterBodyM		
lakeResInflowM		
lakeResOutflowM		
lakeResStorage		
resStorage		
act_SurfaceWaterAbst		
totalSto	Total soil,snow and vegetation storage for each cell including all landcover typ	m
pretotalSto	Previous totalSto	m
sum_actBareSoilEvap		
sum_openWaterEvap		
addtoevapotrans		
sum_directRunoff		
sum_interflow		
sum_capRiseFromGW	capillar rise from groundwater to 3rd soil layer (summed up for all land cover c	m
sum_act_irrConsumpti		
sum_perc3toGW	percolation from 3rd soil layer to groundwater (summed up for all land cover cla	m
sum_prefFlow	preferential flow from soil to groundwater (summed up for all land cover classes	m
act_irrWithdrawal		
act_nonIrrConsumptio		
returnFlow		
DtSec	number of seconds per timestep (default = 86400)	s
cellArea	Cell area [m^2] of each simulated mesh	
baseflow	simulated baseflow (= groundwater discharge to river)	m
Precipitation	Precipitation (input for the model)	m
lddCompress	compressed river network (without missing values)	–
discharge	discharge	m3/s
prelakeResStorage		
catchmentAll		
sumsideflow		
EvapoChannel		
prechannelStorage		
channelStorage		
runoff		
gridcell_storage		
nonFossilGroundwater	groundwater abstraction which is sustainable and not using fossil resources	m
totalET	Total evapotranspiration for each cell including all landcover types	m
sum_actTransTotal		
sum_interceptEvap		
prergridcell		

Continued on next page

Table 8 – continued from previous page

Variable [self.var]	Description	Unit
nonIrrReturnFlow		
localQW		
channelStorageBefore		
sumbalance		
sum_balanceStore		
sum_balanceFlux		
unmetDemand		
act_nonIrrWithdrawal		
returnflowIrr		
nonIrrReturnFlowFrac		
lakeReservoirStorage		
unmet_lost		

Functions

checkWaterSoilGround()

Check water balance of snow, vegetation, soil, groundwater

dynamic()

Dynamic part of the water balance module If option **sumWaterBalance** sum water balance for certain variables

initial()

Initial part of the water balance module

waterBalanceCheck(*fluxesIn*, *fluxesOut*, *preStorages*, *endStorages*, *processName*, *printTrue=False*)

Dynamic part of the water balance module

Returns the water balance for a list of input, output, and storage map files

Parameters

- **fluxesIn** – income
- **fluxesOut** – this goes out
- **preStorages** – this was in before
- **endStorages** – this was in afterwards
- **processName** – name of the process
- **printTrue** – calculate it?

Returns

-

waterBalanceCheckSum(*fluxesIn*, *fluxesOut*, *preStorages*, *endStorages*, *processName*, *printTrue=False*)

Returns the water balance for a list of input, output, and storage map files and sums it up for a catchment

Parameters

- **fluxesIn** – income
- **fluxesOut** – this goes out
- **preStorages** – this was in before
- **endStorages** – this was in afterwards

- **processName** – name of the process
- **printTrue** – calculate it?

Returns Water balance as output on the screen

management_modules package

Data management

data_handling module

Managing data and data handling

`cwatm.management_modules.data_handling.`**`cbinding`**`(`*inBinding*`)`

Check if variable in settings file has a counterpart in the source code

Parameters **inBinding** – parameter in settings file

`cwatm.management_modules.data_handling.`**`checkMeteo_Wordclim`**`(`*meteodata, wordclimdata*`)`

reads the map attributes of meteo dataset and wordclima dataset and compare if it has the same map extend

Parameters

- **nmeteodata** – name of the meteo netcdf file
- **wordlclimdata** – cname of the wordlclim netcdf file

Returns True if meteo and wordclim has the same mapextend

:raises if map extend is different `management_modules.messages.CWATMFileError()`

`cwatm.management_modules.data_handling.`**`checkOption`**`(`*inBinding*`)`

Check if option in settings file has a counterpart in the source code

Parameters **inBinding** – parameter in settings file

`cwatm.management_modules.data_handling.`**`compressArray`**`(`*map, name='None', zeros=0.0*`)`

Compress 2D array with missing values to 1D array without missing values

Parameters

- **map** – in map
- **name** – filename of the map
- **zeros** – add zeros (default= 0) if values of map are to big or too small

Returns Compressed 1D array

`cwatm.management_modules.data_handling.`**`decompress`**`(`*map*`)`

Decompress 1D array without missing values to 2D array with missing values

Parameters **map** – numpy 1D array as input

Returns 2D array for displaying

`cwatm.management_modules.data_handling.`**`divideValues`**`(`*x, y, default=0.0*`)`

returns the result of a division that possibly involves a zero

Parameters

- **x** –

- **y** – divisor
- **default** – return value if y =0

Returns result of x/y or default if y = 0

`cwatm.management_modules.data_handling.`**getmeta**(*key*, *varname*, *alternative*)
get the meta data information for the netcdf output from the global variable metaNetcdfVar

Parameters

- **key** – key
- **varname** – variable name e.g. self.var.Precipitation

Returns metadata information

`cwatm.management_modules.data_handling.`**loadmap**(*name*, *lddflag=False*, *compress=True*, *local=False*, *cut=True*)
load a static map either value or pc raster map or netcdf

Parameters

- **name** – name of map
- **lddflag** – if True the map is used as a ldd map
- **compress** – if True the return map will be compressed
- **local** – if True the map is local and will be not cut
- **cut** – if True the map will be not cut

Returns 1D numpy array of map

`cwatm.management_modules.data_handling.`**loadsetclone**(*self*, *name*)
load the maskmap and set as clone

Parameters **name** – name of mask map, can be a file or - row col cellsize xupleft yupleft -

Returns new mask map

`cwatm.management_modules.data_handling.`**mapattrNetCDF**(*name*, *check=True*)
get the 4 corners of a netcdf map to cut the map defines the rectangular of the mask map inside the netcdf map calls function `management_modules.data_handling.readCoord()`

Parameters

- **name** – name of the netcdf file
- **check** – checking if netcdffile exists

Returns cut1,cut2,cut3,cut4

Raises **if cell size is different** – `management_modules.messages.CWATMError()`

`cwatm.management_modules.data_handling.`**mapattrNetCDFMeteo**(*name*, *check=True*)
get the map attributes like col, row etc from a netcdf map and define the rectangular of the mask map inside the netcdf map calls function `management_modules.data_handling.readCoordNetCDF()`

Parameters

- **name** – name of the netcdf file
- **check** – checking if netcdffile exists

Returns cut0,cut1,cut2,cut3,cut4,cut5,cut6,cut7 `cwatm.management_modules.data_handling.`**`mapattrTiff`**`(`*nf2*`)`

map attributes of a geotiff file

Parameters **`nf2`** –

Returns cut0,cut1,cut2,cut3

`cwatm.management_modules.data_handling.`**`maskfrompoint`**`(`*mask2D*, *xleft*, *yup*`)`

load a static map either value or pc raster map or netcdf

Parameters

- **`mask2D`** – 2D array of new mask
- **`xleft`** – left lon coordinate
- **`yup`** – upper lat coordinate

Returns new mask map

`cwatm.management_modules.data_handling.`**`metaNetCDF`**`()`

get the map metadata from precipitation netcdf maps

`cwatm.management_modules.data_handling.`**`multinetdf`**`(`*meteomaps*, *startcheck='dateBegin'*`)`

Parameters

- **`meteomaps`** – list of meteomaps to define start and end time
- **`startcheck`** – date of beginning simulation

Returns

Raises **`if no map stack in meteo map folder`** – `management_modules.messages.CWATMFileError()`

`cwatm.management_modules.data_handling.`**`readCalendar`**`(`*name*`)`

`cwatm.management_modules.data_handling.`**`readCoord`**`(`*name*`)`

get the meta data information for the netcdf output from the global variable metaNetcdfVar

Parameters **`name`** – name of the netcdf file

Returns latitude, longitude, cell size, inverse cell size

`cwatm.management_modules.data_handling.`**`readCoordNetCDF`**`(`*name*, *check=True*`)`

reads the map attributes col, row etc from a netcdf map

Parameters

- **`name`** – name of the netcdf file
- **`check`** – checking if netcdffile exists

Returns latitude, longitude, cell size, inverse cell size

Raises **`if no netcdf map can be found`** – `management_modules.messages.CWATMFileError()`

`cwatm.management_modules.data_handling.`**`readmeteodata`**`(`*name*, *date*, *value='None'*, *addZeros=False*, *zeros=0.0*, *mapsscale=True*, *modflowSteady=False*`)`

load stack of maps 1 at each timestamp in netcdf format

Parameters

- **name** – file name
- **date** –
- **value** – if set the name of the parameter is defined
- **addZeros** –
- **zeros** – default value
- **mapsscale** – if meteo maps have the same extend as the other spatial static m

Returns Compressed 1D array of meteo data

Raises

- **if data is wrong** – management_modules.messages.CWATMError()
- **if meteo netcdf file cannot be opened** – management_modules.messages.CWATMFileError()

cwatm.management_modules.data_handling.**readnetcdf2**(*namebinding*, *date*, *useDaily='daily'*, *value='None'*, *addZeros=False*, *cut=True*, *zeros=0.0*, *meteo=False*, *usefilename=False*, *compress=True*)

load stack of maps 1 at each timestamp in netcdf format

Parameters

- **namebinding** – file name in settings file
- **date** –
- **useDaily** – if True daily values are used
- **value** – if set the name of the parameter is defined
- **addZeros** –
- **cut** – if True the map is clipped to mask map
- **zeros** – default value
- **meteo** – if map are meteo maps
- **usefilename** – if True filename is given False: filename is in settings file
- **compress** – True - compress data to 1D

Returns Compressed 1D array of netcdf stored data

Raises

- **if netcdf file cannot be opened** – management_modules.messages.CWATMFileError()
- **if netcdf file is not of the size of mask map** – management_modules.messages.CWATMWarning()

cwatm.management_modules.data_handling.**readnetcdfInitial**(*name*, *value*, *default=0.0*)

load initial condition from netcdf format

Parameters

- **name** – file name
- **value** – netcdf variable name

- **default** – (optional) if no variable is found a warning is given and value is set to default

Returns Compressed 1D array of netcdf stored data

Raises

- **if netcdf file is not of the size of mask map** – `management_modules.messages.CWATMError()`
- **if varibale name is not included in the netcdf file** – `management_modules.messages.CWATMWarning()`

`cwatm.management_modules.data_handling.`**readnetcdfWithoutTime**(*name*, *value='None'*)

load maps in netcdf format (has no time format)

Parameters

- **namebinding** – file name in settings file
- **value** – (optional) netcdf variable name. If not given -> last variable is taken

Returns Compressed 1D array of netcdf stored data

`cwatm.management_modules.data_handling.`**report**(*valueIn*, *name*, *compr=True*)

For debugging: Save the 2D array as .map or .tif

Parameters

- **name** – Filename of the map
- **valueIn** – 1D or 2D array in
- **compr** – (optional) array is 1D (default) or 2D

Returns

-

```
Example:
> report(c:/temp/ksat1.map, self_.var_.ksat1)
```

`cwatm.management_modules.data_handling.`**returnBool**(*inBinding*)

Test if parameter is a boolean and return an error message if not, and the boolean if everything is ok

Parameters **inBinding** – parameter in settings file

Returns boolean of inBinding

`cwatm.management_modules.data_handling.`**setmaskmapAttr**(*x*, *y*, *col*, *row*, *cell*)

Definition of cell size, coordinates of the meteo maps and maskmap

Parameters

- **x** – upper left corner x
- **y** – upper left corner y
- **col** – number of cols
- **row** – number of rows
- **cell** – cell size

Returns

-

`cwatm.management_modules.data_handling.`**`valuecell`**(*coordx*, *coordstr*, *returnmap=True*)

to put a value into a raster map -> invert of cellvalue, map is converted into a numpy array first

Parameters

- **`coordx`** – x,y or lon/lat coordinate
- **`coordstr`** – String of coordinates

Returns 1D array with new value

`cwatm.management_modules.data_handling.`**`writeIniNetcdf`**(*netfile*, *varlist*, *inputlist*)

write variables to netcdf init file

Parameters

- **`netfile`** – file name
- **`varlist`** – list of variable to be written in the netcdf file
- **`inputlist`** – stack of 1D arrays

Returns

-

`cwatm.management_modules.data_handling.`**`writenetcdf`**(*netfile*, *prename*, *addname*, *varunits*, *inputmap*, *timeStamp*, *posCnt*, *flag*, *flagTime*, *nrdays=None*, *dateunit='days'*)

write a netcdf stack

Parameters

- **`netfile`** – file name
- **`prename`** – 1st part of variable name with tell which variable e.g. discharge
- **`addname`** – part of the variable name with tells about the timestep e.g. daily, monthly
- **`varunits`** – unit of the variable
- **`inputmap`** – 1D array to be put as netcdf
- **`timeStamp`** – time
- **`posCnt`** – calculate nummer of the indece for time
- **`flag`** – to indicate if the file is new -> netcdf header has to be written,or simply appending data
- **`flagtime`** – to indicate the variable is time dependend (not a single array!)
- **`nrdays`** – (optional) if indicate number of days are set in the time variable (makes files smaller!)
- **`dateunit`** – (optional) dateunit indicate if the timestep in netcdf is days, month or years

Returns flag: to indicate if the file is set up

timestep module

Managing time

`cwatm.management_modules.timestep.`**`Calendar`**(*input*, *errorNo=0*)

Get the date from CalendarDayStart in the settings xml Reformatting the date till it fits to datetime

Parameters

- **input** – string from the settingsfile should be somehow a date
- **errorNo** – 0: check startdate, enddate 1: check startinit

Returns a datetime date

`cwatm.management_modules.timestep.`**`addmonths`**(*d*, *x*)

Adds months to a date

Parameters

- **d** – date
- **x** – month to add

Returns date with added months

`cwatm.management_modules.timestep.`**`checkifDate`**(*start*, *end*, *spinup*, *name*)

Checks if start date is earlier than end date etc And set some date variables

Parameters

- **start** – start date
- **end** – end date
- **spinup** – date till no output is generated = warming up time

Returns a list of date variable in: dateVar

`cwatm.management_modules.timestep.`**`ctbinding`**(*inBinding*)

Check if variable in settings file has a counterpart in source code

Parameters **x** – variable in settings file to be tested

Returns

-

Raises if variable is not found send an error: `management_modules.messages.CWATMError()`

`cwatm.management_modules.timestep.`**`date2indexNew`**(*date*, *nctime*, *calendar*, *select='nearest'*, *name=''*)

The original netCDF4 library cannot handle month and years Replace: date2index This one checks for days, month and years And set some date variables

Parameters

- **date** – date
- **nctime** – time unit of the netcdf file
- **select** – (optional) which date is selected, default: nearest
- **name** – (optional) name of th dataset

Returns index

`cwatm.management_modules.timestep.`**`date2str`**(*date*)

Convert date to string of date e.g. 27/12/2018 :param date: date as (datetime) :return: date string

`cwatm.management_modules.timestep.`**`datenum`**(*date*)

converts date to a int number based on the calender and unit of the netcdf file :param date: :return: number of the date `cwatm.management_modules.timestep.`**`datetoInt`**(*dateIn*, *begin*, *both=False*)

Calculates the integer of a date from a reference date

Parameters

- **`dateIn`** – date
- **`begin`** – reference date
- **`both`** – if set to True both the int and the string of the date are returned

Returns integer value of a date, starting from begin date

`cwatm.management_modules.timestep.`**`datetosaveInit`**(*initdates*, *begin*, *end*)

Calculates the save init dates

Parameters

- **`initdates`** – one or several dates
- **`begin`** – reference date
- **`end`** – end date

Returns integer value of a dates, starting from begin date

`cwatm.management_modules.timestep.`**`numdate`**(*num*, *add=0*)

converts int into date based on the calender and unit of the netcdf file :param num: number of the day :param add: addition to date in days :return: date

`cwatm.management_modules.timestep.`**`timemeasure`**(*name*, *loops=0*, *update=False*, *sample=1*)

Measuring of the time for each subroutine

Parameters

- **`name`** – name of the subroutine
- **`loops`** – if it it called several times this is added to the name
- **`update`** –
- **`sample`** –

Returns add a string to the time measure string: timeMesString

`cwatm.management_modules.timestep.`**`timestep_dynamic`**(*self*)

Dynamic part of setting the date Current date is increasing, checking if beginning of month, year

Returns a list of date variable in: dateVar

configuration module

Loading and parsing of the settings file

`class` `cwatm.management_modules.configuration.`**`ExtParser`**(**args*, ***kwargs*)

Bases: `configparser.ConfigParser`

addition to the parser to replace placeholders

Example

PathRoot = C:/work MaskMap = $(FILE_PATHS:PathRoot)/data/areamaps/area.tif

get (*section*, *option*, *raw=False*, *vars=None*, ***kwargs*)

def get(self, section, option, raw=False, vars=None placeholder replacement

Parameters

- **section** – section part of the settings file
- **option** – option part of the settings file
- **raw** –
- **vars** –

Returns

`cwatm.management_modules.configuration.`**parse_configuration**(*settingsFileName*)

Parse settings file

Parameters **settingsFileName** – name of the settings file

Returns parameters in list: binding, options in list: option

`cwatm.management_modules.configuration.`**read_metanetcdf**(*metaxml*, *name*)

Read the metadata for netcdf output files unit, long name, standard name and additional information

Parameters

- **metaxml** – file mit information for netcdf files (metadata)
- **name** – file name information

Returns List with metadata information: metaNetcdfVar

management_modules.messages module

Error handling - giving out messages

exception `cwatm.management_modules.messages.`**CWATMError**(*msg*)

Bases: `Warning`

The error handling class prints out an error

Parameters **Warning** – class CWATMError

Returns prints out a message about an error

exception `cwatm.management_modules.messages.`**CWATMFileError**(*filename*, *msg=''*, *sname=''*)

Bases: *cwatm.management_modules.messages.CWATMError* (page 216)

The error handling class prints out an error

Parameters **Warning** – class CWATMError

Returns prints out a message about file error

exception `cwatm.management_modules.messages.`**CWATMRunInfo**(*outputS*)

Bases: `Warning`

prints out an error

Parameters **Warning** – class warning

Returns prints out a message

Warning warning given with a header and a message from the subroutine

exception `cwatm.management_modules.messages.`**`CWATMWarning`**(*msg*)

Bases: `Warning`

the error handling class prints out an error

Parameters **`Warning`** – class warning

Returns prints out a message

Handling output of CWATM

management_modules.output module

class `cwatm.management_modules.output.`**`outputTssMap`**(*model*)

Bases: `object`

Output of time series and map

Global variables

Variable [self.var]	Description	Unit
dirUp	river network in upstream direction	–
cellArea	Cell area [m^2] of each simulated mesh	
sampleAdresses		
noOutpoints		
evalCatch		
catcharea		
firstout		
discharge	discharge	m3/s

Functions

`dynamic`(*ef=False*)

Dynamic part of the output module Output of maps and timeseries

Parameters **`ef`** – done with environmental flow

`initial`()

Initial part of the output module

management_modules.checks module

Checking maps if they fit in

`cwatm.management_modules.checks.`**`checkmap`**(**args*, ***kwargs*)

`cwatm.management_modules.checks.`**`counted`**(*fn*)

count number of times a subroutine is called

Parameters **`fn`** –

Returns number of times the subroutine is called

Program management

Global definition of variables

globals module

Global definition of variables

`cwatm.management_modules.globals.`**`globalFlags`**(*setting*, *arg*, *settingsfile*, *Flags*)

Read flags - according to the flags the output is adjusted quiet,veryquiet, loud, checkfiles, noheader,printtime, warranty

Parameters **`arg`** – argument from calling cwatm

`cwatm.management_modules.globals.`**`globalclear`**()

dynamicModel module

Framework of initial and dynamic modules

`class` `cwatm.management_modules.dynamicModel.`**`DynamicModel`**

Bases: `object`

`i = 1`

`class` `cwatm.management_modules.dynamicModel.`**`ModelFrame`**(*model*, *lastTimeStep=1*, *firstTimestep=1*)

Bases: `object`

Frame of the dynamic hydrological model

lastTimeStep: Last time step to run firstTimestep: Starting time step of the model

`initialize_run`()

`run`()

Run the dynamic part of the model

Returns

-

`step`()

replace_pcr module

Some pcr operation are done in numpy

`cwatm.management_modules.replace_pcr.`**`npareaaverage`**(*values*, *areaclass*)

numpy area average procedure

Parameters

- **`values`** –
- **`areaclass`** –

Returns calculates the average area of a class

`cwatm.management_modules.replace_pcr.`**`npareamajority`**`(`*values*, *areaclass*`)`
numpy area majority procedure

Parameters

- **`values`** –
- **`areaclass`** –

Returns calculates the majority of an area of a class

`cwatm.management_modules.replace_pcr.`**`npareamaximum`**`(`*values*, *areaclass*`)`
numpy area maximum procedure

Parameters

- **`values`** –
- **`areaclass`** –

Returns calculates the maximum of an area of a class

`cwatm.management_modules.replace_pcr.`**`npareatotal`**`(`*values*, *areaclass*`)`
numpy area total procedure

Parameters

- **`values`** –
- **`areaclass`** –

Returns calculates the total area of a class

14.3 Global dataset

If you are interested in obtaining the gloabl data set,
please send an email to wfas.info@iiasa.ac.at
We will give you access to our ftp server

14.4 Remarks

We as developers belief that CWatM should be utilize to encourage ideas and to advance hydrological, environmental science and stimulate integration into other science disciplines.
CWatM is based on existing knowledge of hydrology realized with Python and C++. Especially ideas from HBV, PCR-GLOBE, LISFLOOD, H08, MatSiro, WaterGAP are used for inspiration.

Your support is more then welcome and highly appreciated Have fun!

The developers of CWat Model

[74] http://www.iiasa.ac.at/cwatm