

---

# **SPHY Documentation**

***Release 2.2.1***

**Wilco Terink**

**Jun 16, 2019**



---

## Contents

---

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Documentation</b>	<b>5</b>
<b>3</b>	<b>Contents</b>	<b>7</b>
3.1	Concepts . . . . .	7
3.2	Applications . . . . .	10
3.3	Installation . . . . .	17
3.4	Building a model . . . . .	20
3.5	Running a model . . . . .	30
3.6	Releases . . . . .	31
3.7	FAQ . . . . .	41
3.8	References . . . . .	41
<b>4</b>	<b>Copyright</b>	<b>43</b>
	<b>Bibliography</b>	<b>45</b>





**Note:** The latest model release is version 2.2.1. Have a look at the *[Installation Section](#)* for easy model installation.

Alternatively, the source code and latest release can be downloaded from:

<https://github.com/WilcoTerink/SPHY>

All releases can be downloaded from:

<https://github.com/WilcoTerink/SPHY/releases>

---



# CHAPTER 1

---

## Introduction

---

This document describes the Spatial Processes in HYdrology (SPHY) model [31]. The Spatial Processes in HYdrology (SPHY) model is a hydrological modeling tool suitable for a wide range of water resource management applications. SPHY is a state-of-the-art, easy to use, robust tool, that can be applied for operational as well as strategic decision support. The SPHY model has been applied and tested in various studies ranging from real-time soil moisture predictions in flat lands, to operational reservoir inflow forecasting applications in mountainous catchments, irrigation scenarios in the Nile Basin, and detailed climate change impact studies in the snow- and glacier-melt dominated the Himalayan region. Example applications can be found [here](#).



## CHAPTER 2

---

### Documentation

---

The theory and concepts of SPHY model version 2.0.0 [31] are published in the peer-reviewed journal *Geoscientific Model Development*. A link to the pdf of the article can be found [here](#).

Differences between version 2.0.0 and later releases are described in the *Releases Section*.



### 3.1 Concepts

**Note:** The concepts described in this section correspond to the latest release (version 2.2.1).

Model concepts and processes of older releases may differ from the current release. It is therefore recommended to read the release notes of older versions if you plan to use an older release.

#### 3.1.1 Introduction

SPHY is a spatially distributed leaky bucket type of model, and is applied on a cell-by-cell basis. The main terrestrial hydrological processes are described in a conceptual way so that changes in storages and fluxes can be assessed adequately over time and space. SPHY is written in the Python programming language using the [PCRaster](#) ([14], [15], [13]) dynamic modeling framework.

SPHY is grid based and cell values represent averages over a cell. For glaciers, sub-grid variability is taken into account: a cell can be glacier-free, partially glacierized, or completely covered by glaciers. The cell fraction not covered by glaciers consists of either land covered with snow or land that is free of snow. Land that is free of snow can consist of vegetation, bare soil, or open water. The dynamic vegetation module accounts for a time-varying fractional vegetation coverage, which affects processes such as interception, effective precipitation, and potential evapotranspiration. [Figure 3.1](#) provides a schematic overview of the SPHY modeling concepts.

#### 3.1.2 Soil layers and fluxes

The soil column structure is similar to VIC ([17], [18]), with two upper soil storages and a third groundwater storage. Their corresponding drainage components are surface runoff, lateral flow and baseflow. SPHY simulates for each cell precipitation in the form of rain or snow. Precipitation that falls on land surfaces can be intercepted by vegetation and evaporated in part or whole. A part of the liquid precipitation is transformed in surface runoff, whereas the remainder infiltrates into the soil. The resulting soil moisture is subject to evapotranspiration, depending on the soil properties

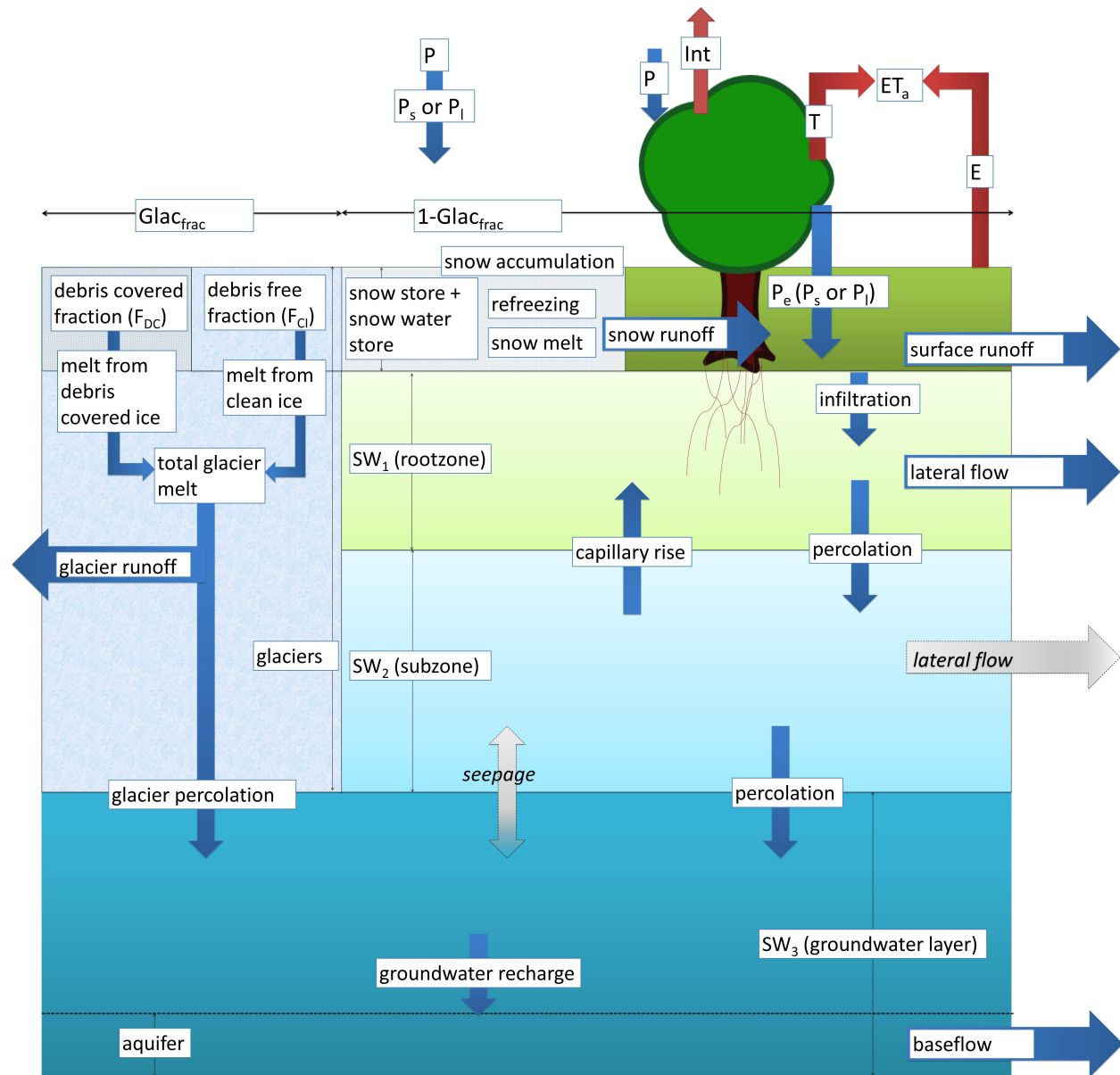


Figure 3.1: SPHY model concepts.



and fractional vegetation cover, while the remainder contributes to river discharge by means of lateral flow from the first soil layer, and baseflow from the groundwater layer.

### 3.1.3 Glaciers

Depending on the temperature and a critical temperature threshold, precipitation can either fall as rain or as snow. The snow storage is updated with snow accumulation and/or snowmelt. Melting of glacier ice contributes to the river discharge by means of a slow and fast component, being (i) percolation to the groundwater layer that eventually becomes baseflow, and (ii) direct runoff.

### 3.1.4 Runoff and routing

The cell-specific runoff, which becomes available for routing, is the sum of surface runoff, lateral flow, baseflow, snowmelt and glacier melt. If no lakes are present, then the user can choose a simple flow accumulation routing scheme: for each cell, the accumulated amount of water that flows out of the cell into its neighboring downstream cell is calculated. This accumulated amount is the amount of water in the cell itself plus the amount of water in upstream cells of the cell, and is calculated using the flow direction network. If lakes are present, then the fractional accumulation flux routing scheme is used; depending on the actual lake storage, a fraction of that storage becomes available for routing and is extracted from the lake, while the remaining part becomes the updated actual lake storage. The flux available for routing is routed in the same way as in the simple flow accumulation routing scheme.

### 3.1.5 Model input

As input, SPHY requires static data as well as dynamic data. For the static data, the most relevant are digital elevation model (DEM), land use type, glacier cover, lakes/reservoirs and soil characteristics. The main dynamic data consist of climate data, such as precipitation, temperature, and reference evapotranspiration. Since SPHY is grid based, optimal use of remote sensing data and global data sources can be made. For example, the Normalized Difference Vegetation Index (NDVI) ([33], [3], [24]) can be used to determine the leaf-area index (LAI) in order to estimate the growth stage of land cover. For setting up the model, streamflow data are not necessary. However, to undertake a proper calibration and validation procedure, flow data are required. The model could also be calibrated using actual evapotranspiration, soil moisture contents, and/or snow-covered area (SCA). An example application in which the SPHY model has been calibrated using MODIS snow cover images can be found [here](#).

### 3.1.6 Model output

The SPHY model provides a wealth of output variables that can be selected based on the preference of the user. Spatial output can be presented as maps of all the available hydrological processes, i.e., actual evapotranspiration, runoff generation (separated by its components), and groundwater recharge. These maps can be generated on a daily basis, but can also be aggregated at monthly or annual time periods. Time-series can be generated for each cell in the study area. Time-series often used are streamflow, actual evapotranspiration and recharge to the groundwater.

### 3.1.7 Modules

SPHY enables the user to turn on/off modules (processes) that are relevant/irrelevant for the area of interest. This concept is very useful if the user is studying hydrological processes in regions where not all hydrological processes are relevant. A user may for example be interested in studying irrigation water requirements in central Africa. For this region, glacier and snow melting processes are irrelevant, and can thus be switched off. The advantages of turning off irrelevant modules are two-fold: (i) decrease model run time, and (ii) decrease the number of required model input data.

## 3.2 Applications

This section describes a couple of applications in which the SPHY model has been applied. For each application it is indicated which **model release** has been used.

### 3.2.1 Spatial resolutions in areal rainfall estimation and their impact on hydrological simulations of a lowland catchment

#### Abstract

Many studies suggest that high-density rain gauge networks are required to capture the rainfall heterogeneities necessary to accurately describe the components of the hydrological cycle. However, equipping and maintaining a high-density rain gauge network will also involve high costs. Although many studies provided useful insights on the required accuracy of rainfall estimates to accurately describe the components of the hydrological cycle, most of these studies focused on streamflow simulations, large river basins or urban environments.

The objective of this study was therefore to evaluate the impact of uncertainties in areal rainfall, estimated at several spatial resolutions, on hydrological simulations of a small  $6.5\text{km}^2$  rural lowland catchment. For this study, the SPHY model was forced with rainfall retrieved from an X-band radar (Figure 3.2) and various synthetic rainfall products, calculated using bootstrap samples of a varying number of radar pixels, treated as virtual rain gauge locations within the catchment. This allowed us to determine the most appropriate resolution of rainfall data to accurately describe the hydrology of a small rural lowland catchment.

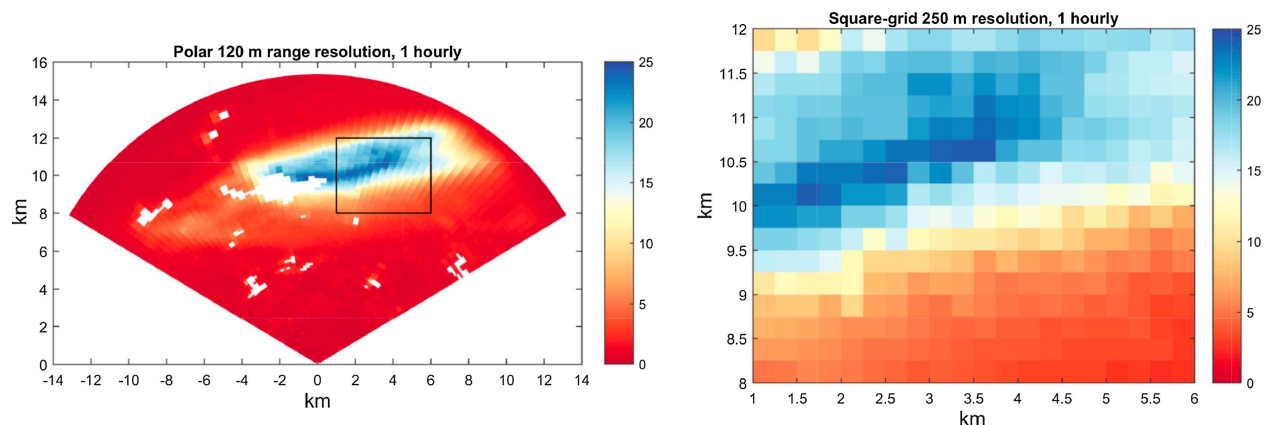


Figure 3.2: left: Example of an aggregated hourly X-band radar image with rain intensity in  $\text{mmh}^{-1}$  and range resolution of 120m. Right: Example of an aggregated hourly X-band radar image with rain intensity in  $\text{mmh}^{-1}$  at a 250m resolution square grid.

We found that the use of one rain gauge to estimate the catchment's areal rainfall may lead to a potential error of more than six times the average hourly rainfall. This may lead to uncertainties in simulated discharge that approach 60% of the average hourly discharge. More than 40 rain gauges are required to reduce the potential error in areal rainfall estimation to values  $<0.1\text{mmh}^{-1}$ . The associated uncertainty in discharge simulations is 20% if 10 rain gauges are used, and 10% if 40 rain gauges are used. The simulation of soil moisture contents and evapotranspiration rates are hardly affected by the number of rain gauges used to estimate the areal rainfall, which is due to the high saturated hydraulic conductivities of the top-soil. At least 12gaugesper $\text{km}^2$  are required to capture the spatial rainfall variation that is present in radar rainfall estimates. Analysis of an individual 18-h rainfall event revealed that the uncertainty in peak areal rainfall estimated using one rain gauge may range between 100% and 600%. The associated uncertainty in simulated discharge for this event ranges between 67 and 233%. With 25 rain gauges the uncertainty in simulated discharge is still in the range of 17 to 33%.

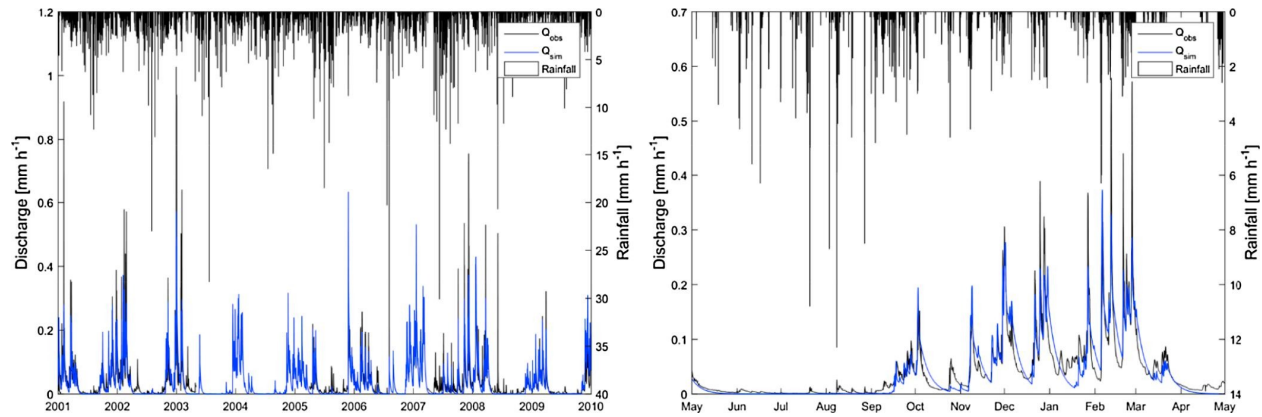


Figure 3.3: Left: observed ( $Q_{\text{obs}}$ ) and SPHY simulated discharge ( $Q_{\text{sim}}$ ) for the calibration period 2001–2009. Right: idem, but for the hydrological year May 2001–April 2002.

## Highlights

- Rain gauge sampling uncertainties impact hydrologic model output in small catchments.
- High-resolution radar data can be used to simulate rain gauge sampling uncertainties.
- A single rain gauge in a 6.5-km<sup>2</sup> area may sample 6 times the average hourly rainfall.
- Discharge uncertainties are 60%, 20%, and 10% for 1, 10, and 40 gauges, respectively.
- Uncertainties increase for larger rainfall intensities.

## Version and reference

This application was done using an unreleased version of SPHY, for which the SPHY model code was modified to allow for hourly model simulations.

A link to the peer-reviewed article [32] of this application can be found below:

- Wilco Terink, Hidde Leijnse, Gé van den Eertwegh, and Remko Uijlenhoet. Spatial resolutions in areal rainfall estimation and their impact on hydrological simulations of a lowland catchment. *Journal of Hydrology*, 563:319–335, aug 2018. URL: <https://www.sciencedirect.com/science/article/pii/S002216941830372X>, doi:10.1016/J.JHYDROL.2018.05.045

The pdf to this application can be found [here](#).

## 3.2.2 Irrigation management in lowland areas

### Introduction

As SPHY produces spatial outputs for the soil moisture content in the root zone and the potential and actual evapotranspiration (ET), it is a useful tool for application in agricultural water management decision support. By facilitating easy integration of remote sensing data, crop growth stages can be spatially assessed at different moments in time. The SPHY dynamic vegetation module ensures that all relevant soil water fluxes correspond to crop development stages throughout the growing season. Spatially distributed maps of root water content and ET deficit can be produced, enabling both the identification of locations where irrigation is required and a quantitative assessment of crop water stress. SPHY has been applied with the purpose of providing field-specific irrigation advice for a largescale farm in western Romania, comprising 380 individual fields and approximately ten different crops.

## Methodology

A high spatial resolution is very relevant for supporting decisions on variable-rate irrigation. The model has therefore been set up using a 30 m resolution, covering the 2013 and 2014 cropping seasons on a daily time-step. Optical satellite data from Landsat 8 [34] were used as input to the dynamic vegetation module. Soil properties were derived from the Harmonized World Soil Database [1], which for Romania contains data from the Soil Geographical Database for Europe [16]. Using the Van Genuchten equation [36], soil saturated water content, field capacity, and wilting point were determined for the HWSD classes occurring at the study site. Elevation data was obtained from the EU-DEM data set [4], and air temperature was measured by two on-farm weather stations.

For calibration purposes, field measurements of soil moisture and/or actual ET are desired. In this case study, one capacitance soil moisture sensor was installed in a soybean field to monitor root-zone water content shortly after 1 May 2014, which is the start of the soybean growing season. The sensor measures volumetric moisture content for every 10 cm of the soil profile up to a depth of 60 cm. It is also equipped with a rain gauge measuring the sum of rainfall and applied irrigation water, which was used as an input to SPHY. Soil moisture measured over the extent covered by the crop root depth was averaged and compared to simulated values.

## Results

In irrigation management applications like these, a model should be capable of simulating the moisture stress experienced by the crop due to insufficient soil moisture contents, which manifests itself by an evapotranspiration deficit (potential ET > actual ET). Figure 3.4 shows the spatial distribution of ET deficit, as simulated by the SPHY model for the entire farm on 3 April 2014. When SPHY is run in an operational setting, this spatial information can be included in a decision support system that aids the farmer in irrigation planning for the coming days.

Since this study was a demonstration project, only an initial model calibration was performed. The model was in this case most sensitive for the crop coefficient ( $K_c$ ), affecting the evaporative demand for water. As can be seen in Figure 3.5, the temporal patterns as measured by the soil moisture sensor are well simulated by the SPHY model. Based on daily soil moisture values, a Nash–Sutcliffe [25] model efficiency coefficient of 0.6 was found, indicating that the quality of prediction of the SPHY model is “good” [5].

Soil moisture simulations could be further improved by conducting a full model calibration, adjusting the soil physical parameters. Remotely sensed evapotranspiration can be used in the calibration process [9], although such data are often not available on these small scales as ET is a very complex variable to assess [26]. It should also be noted that soil moisture content is typically highly variable in space; a very high correlation between point measurements and gridcell simulations of soil moisture may therefore not always be feasible [2].

## Version

This study was carried out using SPHY model release 2.0.0.

### 3.2.3 Snow-fed and glacier-fed river basins

#### Introduction

SPHY is being used in large Asian river basins with significant contribution of glacier melt and snowmelt to the total flow ([11], [19], [22]). The major goals of these applications are two-fold:

- Assess the current hydrological regimes at high resolution; e.g., assess spatial differences in the contributions of glacier melt, snowmelt and rainfall–runoff to the total flow.
- Quantify the effects of climate change on the hydrological regimes in the future and how these affect the water availability.



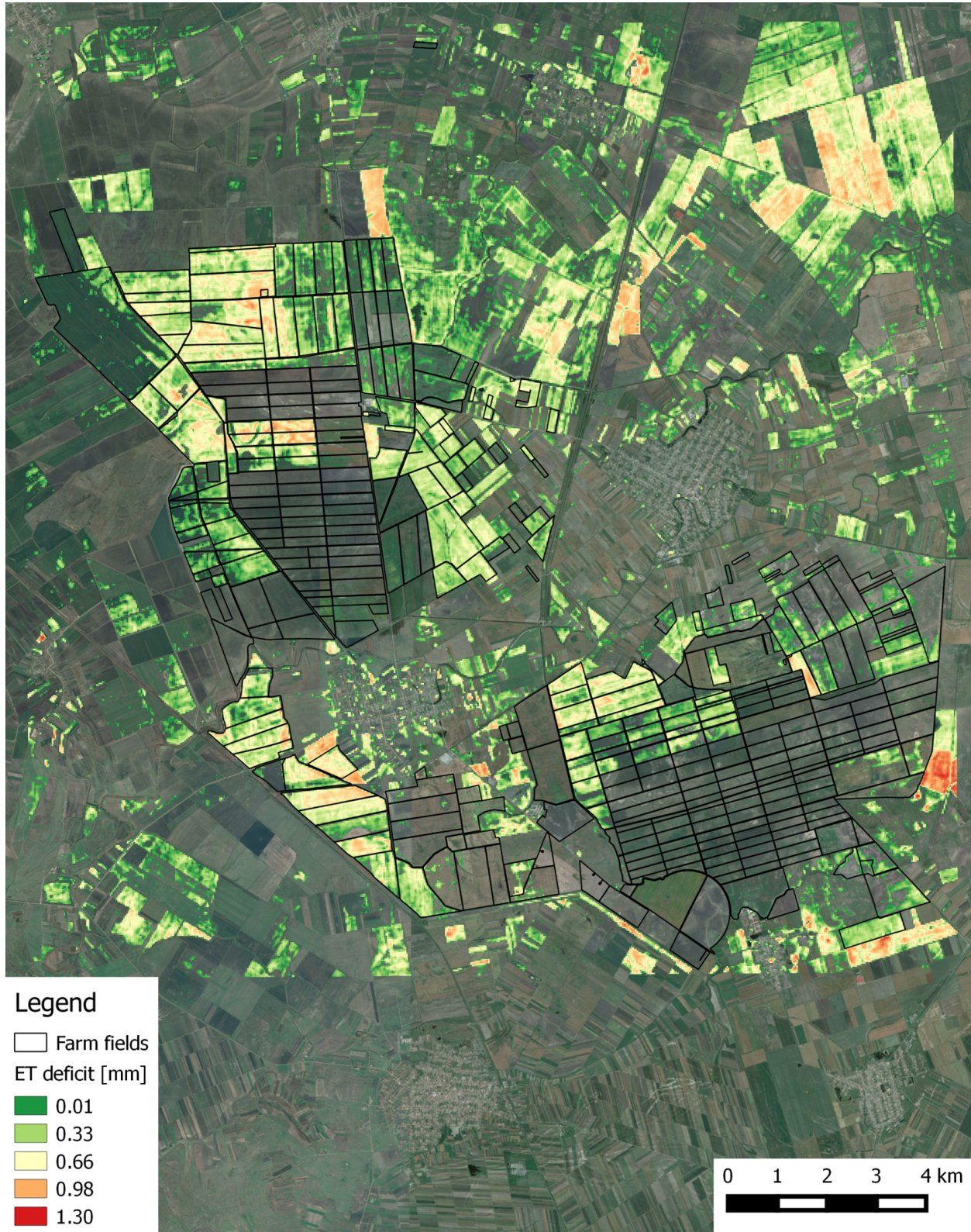


Figure 3.4: Spatial distribution of evapotranspiration (ET) deficit, as simulated by the SPHY model for a Romanian farm on 3 April 2014. Transparency means no ET deficit.

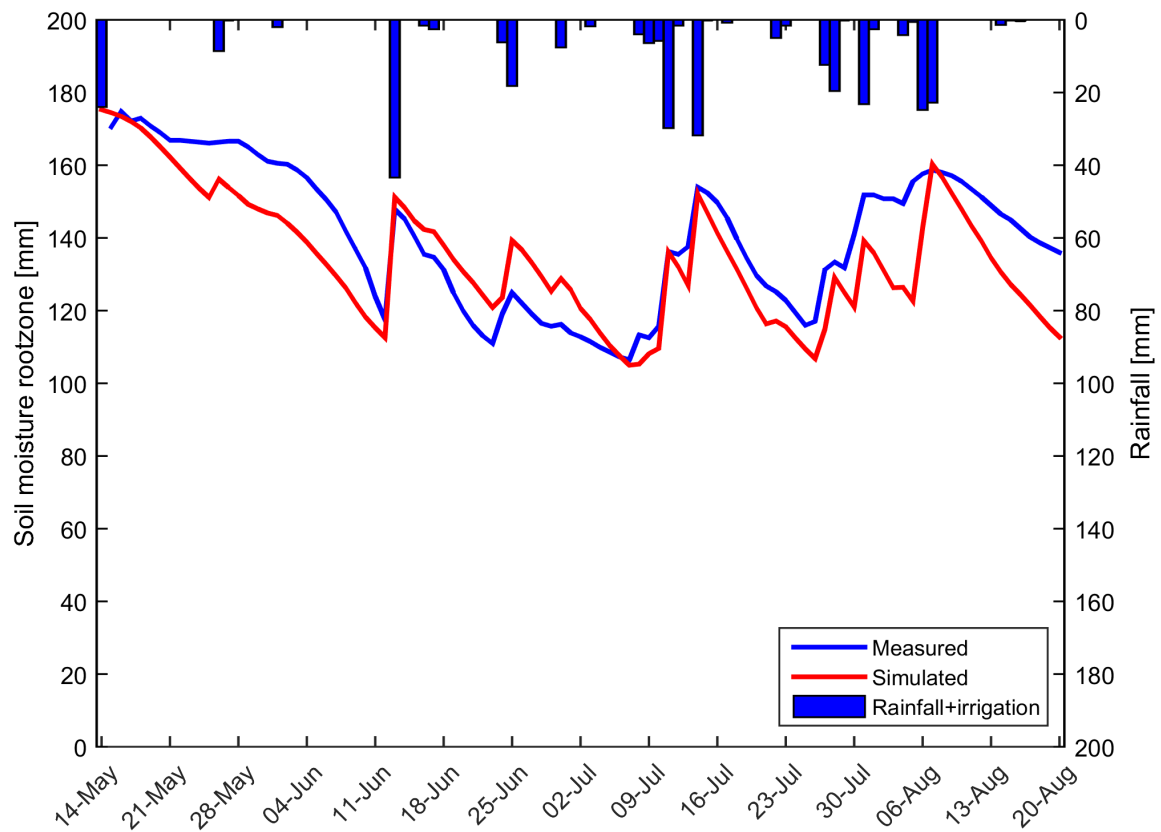


Figure 3.5: Measured and simulated daily root-zone soil moisture content during the 2014 growing season. Rainfall + irrigation has been measured by the rain gauge that was attached to the moisture sensor.

Rivers originating in the high mountains of Asia are considered to be the most meltwater dependent river systems on Earth [27]. In the regions surrounding the Himalayas and the Tibetan Plateau large human populations depend on the water supplied by these rivers [12]. However, the dependency on melt water differs strongly between river basins as a result of differences in climate and differences in basin hypsometry [8]. Only by using a distributed hydrological modelling approach that includes the simulation of key hydrological and cryospheric processes, and inclusion of transient changes in climate, snow cover, glaciers and runoff, appropriate adaptation and mitigation options can be developed for this region [29]. The SPHY model is very suitable for such goals, and has therefore been widely applied in the region.

## Methodology

In the Himalayan Climate Change Adaptation Programme (HICAP), led by the International Centre for Integrated Mountain Development (ICIMOD), SPHY has been successfully applied in the upstreambasins of the Indus, Ganges, Brahmaputra, Salween and Mekong rivers ([20], [22]).

For application in this region, SPHY was setup at a 1 km spatial resolution using a daily time-step, and forced with historical air temperature (Tavg, Tmax, Tmin) and precipitation data, obtained from global and regional datasets (e.g. APHRODITE [35], Princeton [28], TRMM [7]) or interpolated WMO station data from a historical reference period. For this historical reference period SPHY was calibrated and validated using observed streamflow. For the future period, SPHY was forced with downscaled climate change projections obtained from General Circulation Models (GCMs), as available through the Climate Model Intercomparison Projects (e.g. CMIP3 [23], CMIP5 [30]), which were used as basis for the Assessment Reports prepared by the Intergovernmental Panel on Climate Change (IPCC).

## Results

In this study the hydrological regimes of these five basins have been quantified and the calibrated and validated model (Figure 3.6) was forced with an ensemble of eight GCMs to create water availability scenarios until 2050.

Table 3.1 lists the calibration and validation results. Based on the validation results, we concluded that the model performs satisfactory given the large scale, complexity and heterogeneity of the modeled region and data scarcity [22]. We used one parameter set for the entire domain, which inherently means some stations perform better than others. In the particular case of the upper Indus, another possible explanation could be uncertainty in air temperature forcing in the highest parts of the upper Indus basin (locations Dainyor bridge, Besham Qila and Tarbela inflow in Table 3.1), since especially in this area, the used forcing datasets are based on very sparse observations.

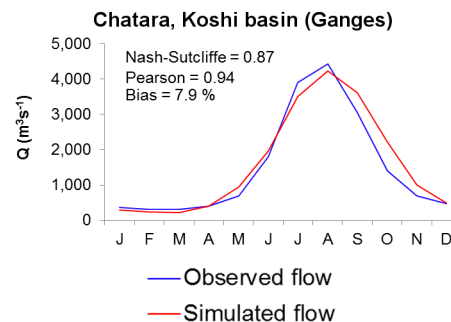


Figure 3.6: Average monthly observed and SPHY-simulated flow (1998-2007) for the Chatara major discharge measurement location in the Ganges basin [22]. Metrics are calculated based on monthly time steps.



Table 3.1: Station locations used for calibration and validation of the SPHY model in HICAP [Lutz2014]. Three stations were used for calibration for 1998-2007. Five stations were used for an independent validation for the same period. The Nash-Sutcliffe efficiency (NS) and bias metrics were calculated at a monthly time step.

Location	NS [-]	Bias [%]	Validation/Calibration
Dainyor bridge	0.39	58.2	validation
Besham Qila	0.66	24.7	validation
Tarbela Inflow	0.63	34.6	calibration
Marala Inflow	0.65	12.0	validation
Pachuwarghat	0.90	-1.6	validation
Rabuwa Bazar	0.65	-22.5	validation
Turkeghat	0.87	-5.4	calibration
Chatara	0.87	7.9	calibration

SPHY allowed the assessment of current contribution of glacier melt and snow melt to total flow (Figure 3.7), and how total flow volumes and the intra-annual distribution of river flow will change in the future [22].

### Calibration on MODIS snow cover

For basins with snow melt being an important contributor to the flow, besides calibration to observed flow, the snow-related parameters in the SPHY model can also be calibrated to observed snow cover. For the Upper Indus basin the snow-related parameters degree-day factor for snow (DDF) and snow water storage capacity (SSC) were calibrated independently using MODIS snow cover imagery [21]. The same MODIS dataset was used as in [10]. From the beginning of 2000 until halfway 2008, the snow cover imagery was averaged for 46 different periods of 8 days (5 days for the last period) to generate 46 different average snow cover maps. E.g. period 1 is the average snowcover for 1-8 January for 2000 until 2008, whereas period 2 is the average snow cover for 9-16 January for 2000 until 2008, etc.

The SPHY model was run for 2000-2007 at a daily time step and for each 1 x 1 km grid cell the average snow cover was calculated for the same 46 periods as in the MODIS observed snow cover dataset. Subsequently, these simulated snowcover maps were resampled to 0.05 degrees spatial resolution, which is the native resolution of the MODIS product.

Figure 3.8 shows the basin-average observed and simulated fractional snow cover for the 46 periods during 2000-2007 and Figure 3.9 shows the same at the 0.05 grid cell level. As a final step, the baseflow recession coefficient and routing coefficient (kx) were calibrated

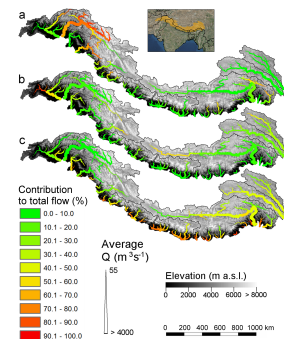
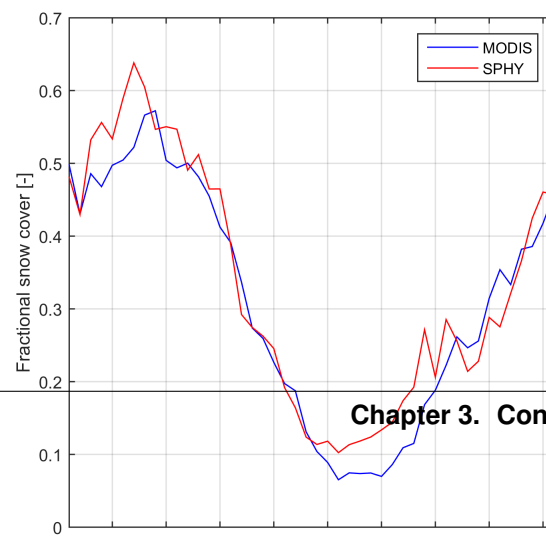


Figure 3.7: The contri-





to match the simulated streamflow with the observed streamflow.

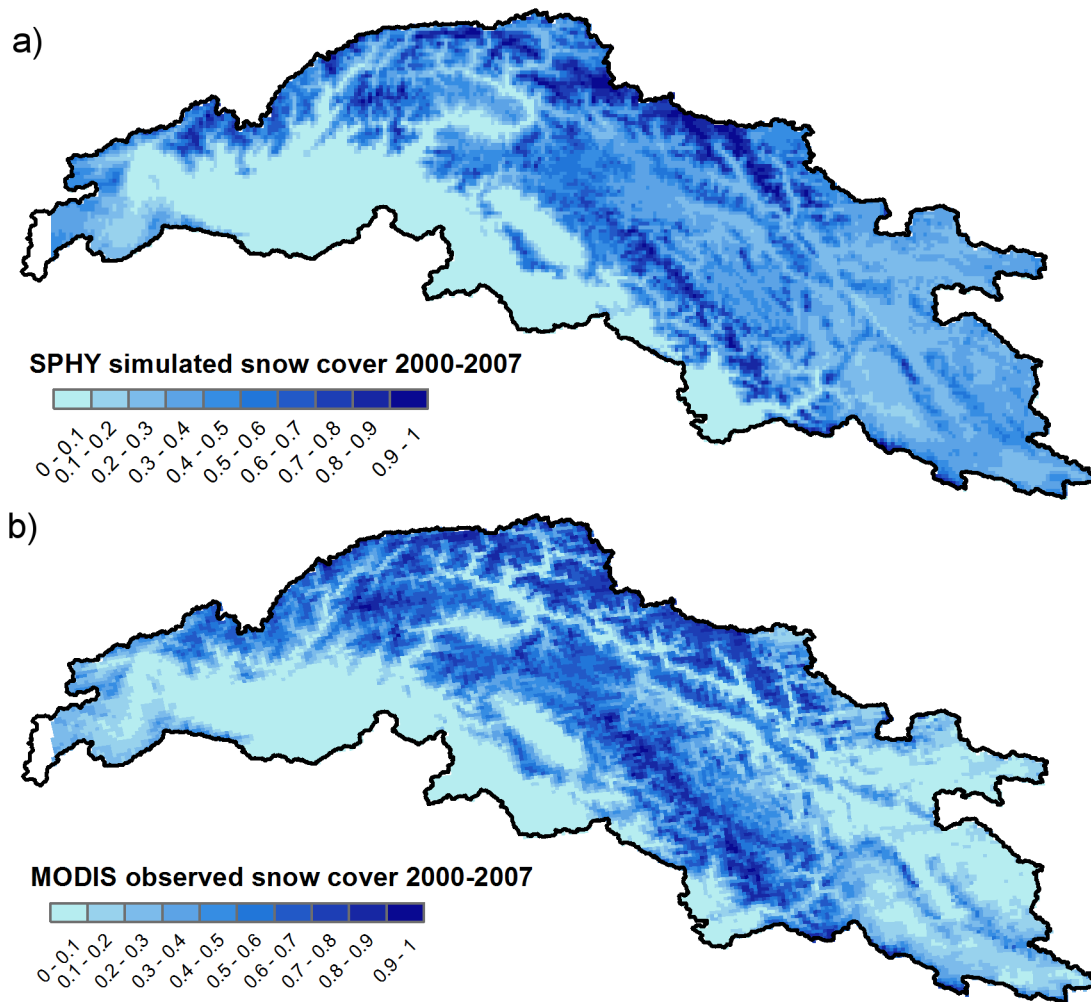


Figure 3.9: Observed and simulated average fractional snow cover in the upper Indus basin. The values represent the 9-year average for 46 (8-day) periods during 2000-2007.

### Version

This study was carried out using SPHY model release 2.0.0.

## 3.3 Installation

**Note:** The SPHY model requires the PCRaster dynamic modelling framework, which needs to be downloaded from: <http://pcraster.geo.uu.nl/>

The PCRaster version required for the SPHY model may vary between each SPHY model release. It is therefore

important to carefully read the [release notes](#) of the SPHY model release you intend to use.

---

This page describes how you can install the SPHY model. There are three different ways of installing the model:

1. Install via [Anaconda](#):

```
conda install -c WilcoTerink SPHY
```

2. Install via [pip](#):

```
pip install SPHY
```

3. Downloading the source code from the [SPHY model GitHub repository](#)

Each of these installation methods is described in detail in the sections below. The recommended installation is the **anaconda** method.

### 3.3.1 Anaconda

#### About Anaconda

Anaconda is the world's most popular Python/R data science platform. The open-source Anaconda distribution is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 11 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling individual data scientists to:

- Quickly download 1,500+ Python/R data science packages
- Manage libraries, dependencies, and environments with Conda
- Develop and train machine learning and deep learning models with scikit-learn, TensorFlow, and Theano
- Analyze data with scalability and performance with Dask, NumPy, pandas, and Numba
- Visualize results with Matplotlib, Bokeh, Datashader, and Holoviews

More information about the Anaconda distribution can be found [here](#).

#### Download Anaconda

It is strongly recommended to use the Anaconda Python distribution to install and manage all your Python packages. The reason for this is because Anaconda checks for dependencies, installs missing dependencies or updates old dependencies if required. Depending on the SPHY model release, a different Anaconda Python distribution may be required (see [Releases](#)). Use one of the links below to download the required Anaconda Python distribution:

- [Python 2.7 version](#)
- [Python 3.7 version](#)

#### Install SPHY using Anaconda

Once you have PCRaster and Anaconda installed you can easily install the SPHY model via the command prompt:

```
conda install -c WilcoTerink SPHY
```

The command above will always install the most recent SPHY model release. Installing a specific SPHY model release can be achieved by:

```
conda install -c WilcoTerink SPHY=2.1.1
```

which in this case installs SPHY model release 2.1.1.

Upgrading an already installed SPHY model distribution to the most recent release can be achieved by:

```
conda update -c WilcoTerink SPHY
```

After you have installed the SPHY model, the SPHY model package resides under:

```
PATH-TO-YOUR-ANACONDA-INSTALLATION\Lib\site-packages\SPHY
```

### 3.3.2 pip

#### About pip

pip is the package installer for Python. You can use pip to install packages from the [Python Package Index](https://pypi.org/) and other indexes. In contrast to Anaconda, which comes with a complete Python installation, pip itself does not contain a Python installation. Therefore, it is required to have a Python distribution already installed before you can use pip. See the link below for more information on how-to install pip:

<https://pip.pypa.io/en/stable/installing/>

#### Install SPHY using pip

Once you have PCRaster and pip installed you can easily install the SPHY model via the command prompt:

```
pip install SPHY
```

The command above will always install the most recent SPHY model release. Installing a specific SPHY model release can be achieved by:

```
pip install SPHY==2.1.1
```

which in this case installs SPHY model release 2.1.1.

Upgrading an already installed SPHY model distribution to the most recent release can be achieved by:

```
pip install SPHY --upgrade
```

After you have installed the SPHY model, the SPHY model package resides under:

```
PATH-TO-PYTHON-INSTALLATION\Lib\site-packages\SPHY
```

### 3.3.3 Download from GitHub repository

Alternatively, you can download the SPHY model from my GitHub repository:

<https://github.com/WilcoTerink/SPHY/releases>

You can download the release you want, and extract the contents to a folder on your hard drive. The SPHY model can then be run from inside this folder.

Installation by this method, however, is not recommended because it does not check for the dependencies that are required to run the model, whereas the *Anaconda* and *pip* installation methods do.

## 3.4 Building a model

---

**Note:** The SPHY model makes extensive use of the PCRaster functionalities. Many of the SPHY model input maps can therefore be created using command line based PCRaster commands. An overview of PCRaster functions can be found [here](#).

---

This page describes the steps that are required to create SPHY model input. There is no unique approach in creating SPHY model input, so the steps described in this section are just an example of how this can be done using QGIS.

### 3.4.1 Clone

The first and most important step in creating your model input is setting the Coordinate Reference System (CRS), the model extent, and the spatial resolution for the **clone** map. This is a map with `TRUE` for all cells that are part of your model domain, and `FALSE` for the cells not being part of your model domain.

---

**Note:** It is important to note that all the model input maps should have exactly the same CRS, extent (xmin, ymin, xmax, ymax), and spatial resolution (number of rows and columns) as the *clone* map.

---

### Coordinate Reference System (CRS)

Calculations in SPHY are metric, which means you need to make sure that all your model data is in a metric coordinate system. [Figure 3.10](#) shows the Universal Transverse Mercator (UTM) zones. You may use this image to pick the UTM zone for your area of interest. The link below provides a complete overview of the available Coordinate Reference Systems (CRSs):

- <http://spatialreference.org/ref/epsg/>

If we for example would like to build a model for an area in the east of Nepal, we would select UTM zone 45N, which is the WGS84 UTM Zone 45 North projection (EPSG:32645) in QGIS.

### Extent

The next step is to define the corners (xmin and ymax) for your area of interest. The easiest way to do this is to add a raster (e.g. a DEM) of your area of interest to QGIS. You need to make sure that this layer is in the same CRS as defined in the previous paragraph (e.g. EPSG:32645). If that is not the case, then you need to re-project ([QGIS warp](#)) the layer to match the defined CRS.

Right-click on the added layer and select properties and finally metadata. An example of how this may look like is shown in [Figure 3.11](#).

The example of [Figure 3.11](#) represents a raster with 271 columns (X) and 455 rows (Y). The four values at the bottom represent the layer extent in the order: xmin, ymin : xmax, ymax. It is not a problem if the added map is slightly larger than your area of interest. However, you need to make sure that the area for which you want to set-up

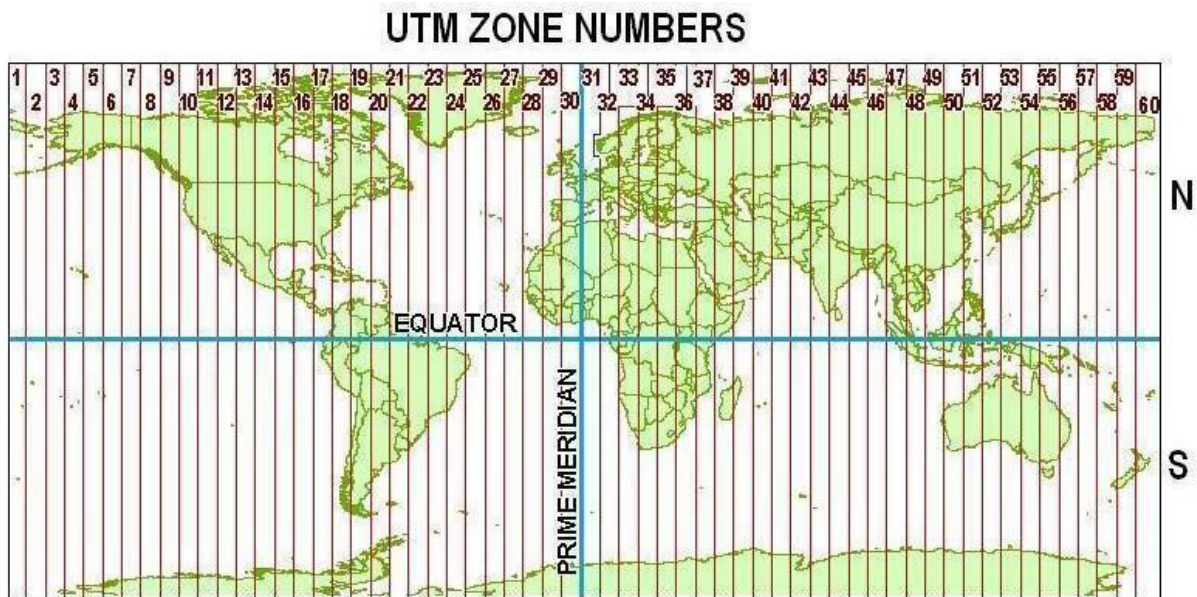


Figure 3.10: UTM zones.

**Dimensions**

X: 271 Y: 455 Bands: 1

**Origin**

391028,3.13376e+06

**Pixel Size**

250,-250

**No Data Value**

255

**Data Type**

Byte - Eight bit unsigned integer

**Pyramid overviews****Layer Spatial Reference System**

+proj=utm +zone=45 +datum=WGS84 +units=m +no\_defs

**Layer Extent (layer original source projection)**

391028.0000000000000000,3020010.0000000000000000 : 458778.0000000000000000,3133760.0000000000000000

Figure 3.11: QGIS metadata.

your model fits well within the extent of the added layer. Note that we can also see that the layer is in the correct CRS: the line below “Layer Spatial Reference System” shows that the layer is in the WGS84 UTM Zone 45.

Let’s refer to this layer as the **source** layer.

## Spatial resolution

Now we are going to define the spatial resolution for our model. The choice of resolution will be a trade-off between the resolution of your input data, computational resources, the number of runs you intend to do and the required detail for your modelling purpose.

Using the example from [Figure 3.11](#) we can see that `xmax` and `xmin` are 485,778 and 391,028, respectively. The total horizontal distance is therefore  $485,778 - 391,028 = 94,750$  m. The vertical distance is  $3,133,760 - 3,020,010 = 113,750$  m. If we would choose a spatial resolution of 250 m, then the number of columns would be 379, and the number of rows 455. Let’s decide we stay with the 250 m resolution. In that case we will have a model with  $379 \times 455 = 172,445$  grid cells.

## Creating the clone

### Source layer is already in the desired spatial resolution

Because the layer we have added to QGIS in the example above is already in the desired spatial resolution of 250 m (see “Pixel Size” in [Figure 3.11](#)), we can use this map as the base for our clone map. As mentioned before, the clone map is a boolean map with `TRUE` for all cells that are part of your model domain, and `FALSE` for the cells not being part of your model domain.

It is not unlikely that the map you have added to QGIS is a Digital Elevation Model (DEM) or other numerical map with floating numbers representing the value in each cell. Since the added map is not a boolean map, we need to convert this map from a float to a boolean map. Before we can do this, we need to make sure that the map is in the `PCRaster *.map` format. If your added map is in GeoTIFF format, then we need to convert the map from `*.tif` to `*.map` using [QGIS Translate](#).

Let’s say we have translated a map with the name `dem.tif` to `dem.map`. This can then easily be converted into a boolean clone map by:

```
pcrcalc clone.map = boolean(dem.map)
```

### Source layer is NOT in the desired spatial resolution

If your added QGIS map is NOT in the desired resolution, then you first need to resample the map to the chosen spatial resolution. One way of doing this is using the [SAGA resampling](#) module from the QGIS Processing Toolbox ([Figure 3.12](#)).

Within the SAGA resample tool ([Figure 3.13](#)) you need to select the “Grid” you like to resample (`dem2` in this example), the “Output extent”, and the “Cellsize” (250 m in this case). The fields for the “Interpolation Method” and output “Grid” can be left unchanged. It is important that the output extent is set correctly, which can easily be achieved by choosing the option “Use layer/canvas extent”. Select the same layer that you specified under “Grid” (because it has the correct extent, but not the correct resolution).

After clicking the “Run” button, a temporary grid is created and will be added to the QGIS canvas. This temporary grid can then be used as **source** layer, and the steps in the [previous paragraph](#) can be followed to create the **clone** map.

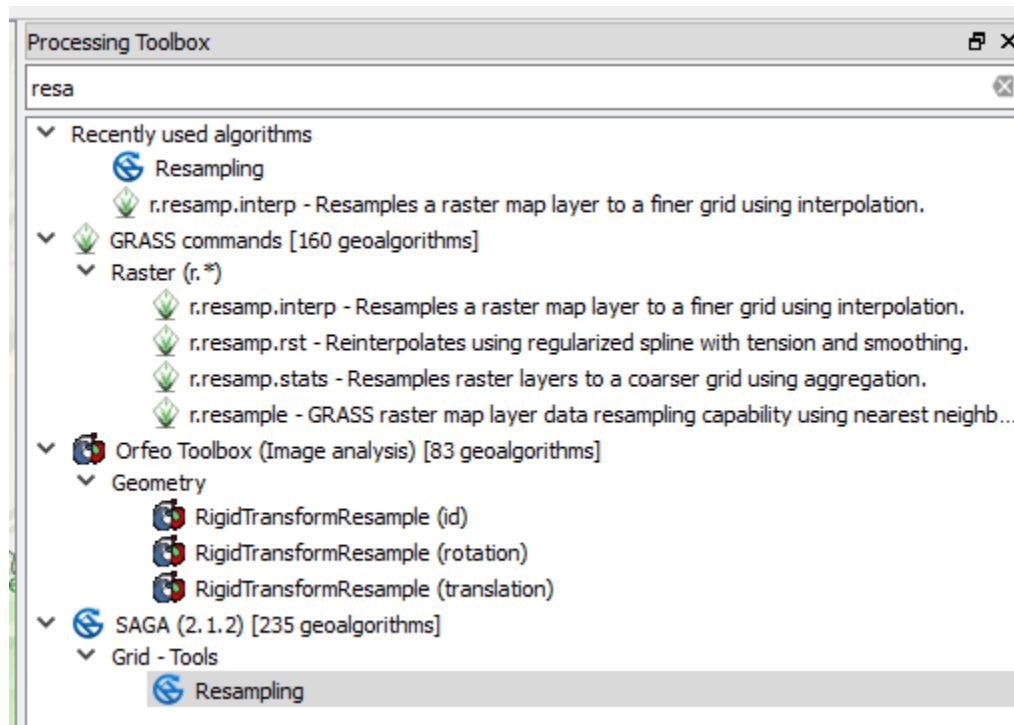


Figure 3.12: SAGA resample.

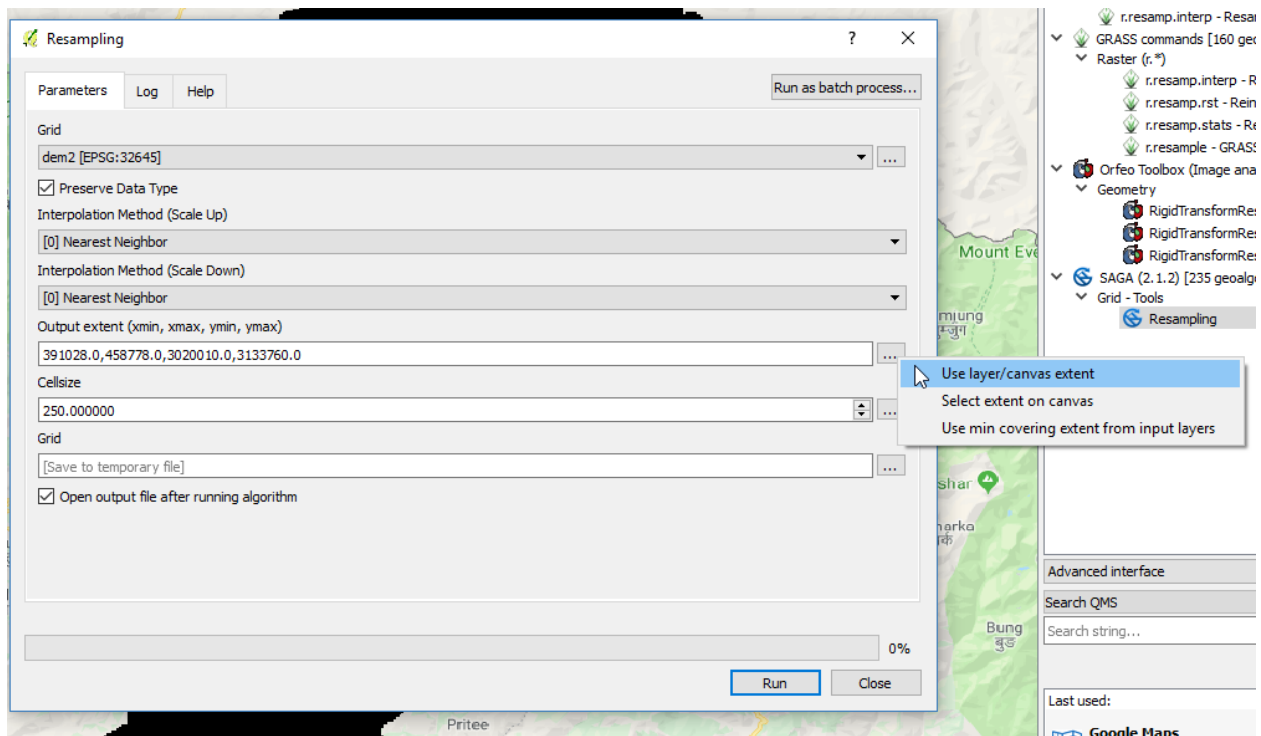


Figure 3.13: Setting the grid cell size and use an existing layer to set the extent.



### 3.4.2 Digital Elevation Model (DEM)

A DEM is a map with for each cell a floating number that represents the elevation in that cell. If your DEM is not in the same CRS as your clone, then you first need to re-project ([QGIS warp](#)) the DEM to the CRS of the clone map. Check the metadata (same as [Figure 3.11](#)) of your DEM map to see if the extent and spatial resolution are the same as the clone map. If that is not the case, then you need to resample the DEM using the same approach as illustrated in [Figure 3.12](#) and [Figure 3.13](#).

Once your DEM is in the same CRS, extent, and spatial resolution as your clone, you can convert the DEM to `dem.map` using [QGIS Translate](#).

If you want the DEM to only have values where your clone is `TRUE`, then you can achieve this by:

```
pcrcalc dem_new.map = if(clone.map, dem.map)
```

### 3.4.3 Slope

Once you have `dem.map` created, the `slope.map` can easily be created by:

```
pcrcalc slope.map = slope(dem.map)
```

### 3.4.4 Local Drain Direction (ldd)

#### ldd

An ldd is a local drain direction map with flow directions from each cell to its steepest downslope neighbour. To create a Local Drain Direction (ldd) map you need a [DEM](#). An ldd is then created using:

```
pcrcalc ldd.map = lddcreate(dem.map, 1e31, 1e31, 1e31, 1e31)
```

The `1e31` (default) arguments can be substituted depending on the orography of your catchment. More details about these arguments can be found [here](#). In most cases, the `1e31` value satisfies the creation of the ldd.

The ldd map in combination with the locations map can be used to create a [catchment or sub-catchment map](#).

#### accuflux

A test to verify if your ldd has been created correctly is to calculate for each cell the number of upstream cells. This can be achieved using the `accuflux` function:

```
pcrcalc accuflux.map = accuflux(ldd.map, 1)
```

You can validate the `accuflux.map` by dragging it into QGIS and compare it with a shapefile of your river network (if you have this).

More details about the `accuflux` function can be found [here](#).

#### lddrepair

Each cell on the local drain direction map must have a pit (value of 5) at the end of its downstream path. If this is not the case for one or more cells on a local drain direction map, the map is called unsound. An unsound local drain direction map can not be used as input expression for the operations with local drain direction maps. Sometimes the unsound ldd can be fixed using `lddrepair`:



```
pcrcalc lddsound.map = lddrepair(ldd.map)
```

More details about `lddrepair` can be found [here](#).

### 3.4.5 Locations

The locations map is required for reporting time-series of the various model fluxes (e.g. routed streamflow). As is shown in [Table 3.2](#), the `locations.map` is a nominal map. The easiest way to create a locations map is to have a point shapefile representing the various stations in your area of interest. For the locations map it is required to:

1. Give the stations a unique ID starting with 1 and the ID of the last station being equal to the total number of stations: e.g. if you have 4 stations for your model, you have to number them 1, 2, 3, and 4.
2. If you want to report routed fluxes, you need to make sure the stations are located on the delineated stream network. A good practice of doing this is to use *accuflux* map as a background layer to check if the position of your stations is correct (see [Figure 3.14](#)).

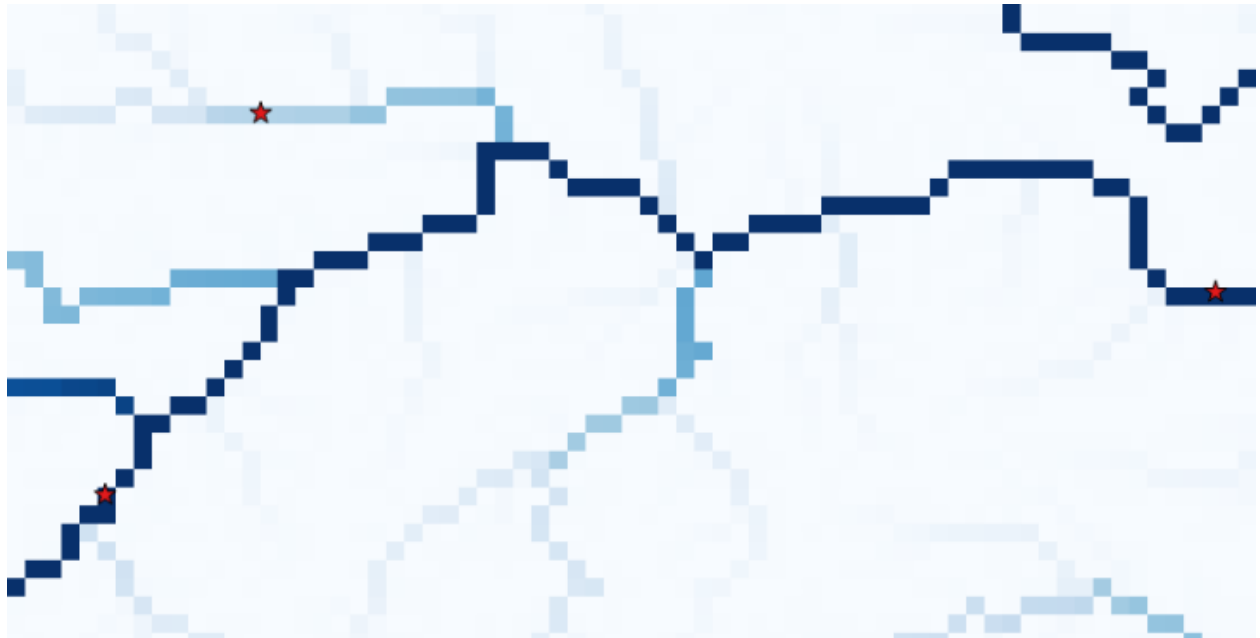


Figure 3.14: Adding stations using the *accuflux.map* as a background. The stations in the shapefile are represented by the red stars.

Once the stations are numbered correctly, and are in the correct location, you again need to make sure that the map is in the same CRS, extent, and spatial resolution as your `clone.map`. Finally you need to make sure the map is saved as a nominal map. The steps are summarized under “*Other static input maps*”.

### 3.4.6 Catchment and sub-catchment

The `catchment.map` and `subcatchment.map` can be created once the `ldd.map` and `locations.map` are created correctly. These two maps can be created using:

```
pcrcalc catchment.map = catchment(ldd.map, locations.map)
```

and:

```
pcrcalc subcatchment.map = subcatchment(ldd.map, locations.map)
```

Obviously, names of the input maps may be different if you have assigned different names in one of the preceding steps.

### 3.4.7 Glaciers

#### SPHY 2.2.0 and newer releases

SPHY 2.2.0 comes with an improved glacier module with glaciers being mass-conserving. With this module you can have a better representation of the glaciers using a higher resolution DEM than the DEM you use to run your model. More information regarding this module can be found under [version 2.2.0](#).

To run the model with this improved glacier module, you need to create the following input:

- **GlacTable:** a csv-file ([Figure 3.15](#)) with the header `U_ID, MOD_ID, GLAC_ID, MOD_H, GLAC_H, DEBRIS, FRAC_GLAC, ICE_DEPTH`. The columns are described below:
  1. `U_ID`: unique combination of model cell ID, glacier ID, glacier elevation, and debris/glacier free; i.e. each line in the csv-file should have a unique `U_ID`.
  2. `MOD_ID`: model cell ID a glacier cell belongs to. Different glaciers (multiple glacier IDs), or just one glacier but with different elevations (based on the higher resolution DEM) may occur within the same model cell ID, which means that in both cases that the model cell ID may occur on multiple lines in the csv-file.
  3. `GLAC_ID`: glacier cells with the same glacier should have the same glacier ID.
  4. `MOD_H`: DEM elevation of the model ID grid cell.
  5. `GLAC_H`: DEM elevation of the glacier ID grid cell.
  6. `DEBRIS`: Flag indicating whether the particular glacier grid cell has debris (`TRUE = 1`), or is debris-free (`FALSE = 0`).
  7. `FRAC_GLAC`: Fraction of the model grid cell ID that is covered with glacier grid cell ID. The total fraction of all glacier grid cell IDs belonging to the same model grid cell ID may never exceed 1.
  8. `ICE_DEPTH`: Thickness of the glacier ID grid cell.
- **ModelID:** Nominal map with a unique ID for each model grid cell (same CRS, extent, and spatial resolution as the clone map).
- **GlacID:** Nominal map with a unique ID for each glacier (same CRS, extent, and spatial resolution as the clone map).

An example of a few lines in the GlacTable is shown in [Figure 3.15](#). As you can see there are four different glaciers shown here, each with their unique `GLAC_ID`. Each line has a unique `U_ID` as is required. You can see that the glacier with `GLAC_ID` equals 17 is the largest of the four, covering ten unique model grid cells (`MOD_ID`). What you also may notice is that this glacier may occur twice within the same model ID grid cell (e.g. 62850). This is because part of the glacier in that model grid cell is debris-covered (with a glacier fraction of 0.92), and the rest of that glacier in that model cell is debris-free (fraction 0.08). This table also illustrates that the sum of all glacier fractions within the same model grid cell never exceeds one (it can be less). The elevation (`MOD_H`) for each model ID grid cell is unique for that particular model ID grid cell.

To be able to construct the required maps and the glacier table csv-file you need:

- A shapefile with the outlines of the glaciers within your area of interest. The [Randolph Glacier Inventory \(RGI\)](#) is a recommended source for this.

U_ID	MOD_ID	GLAC_ID	MOD_H	GLAC_H	DEBRIS	FRAC_GLAC	ICE_DEPTH
216	62825	23	5463.24	5380.17	0	0.24	10.59
217	62826	23	5380.88	5355.00	0	0.08	7.67
218	62831	21	5294.60	5294.77	0	0.88	9.03
219	62832	21	5270.48	5258.57	0	0.84	9.26
220	62833	21	5349.92	5243.80	0	0.20	3.85
221	62848	17	4888.24	4888.24	1	1.00	113.29
222	62849	17	4883.28	4883.28	1	1.00	172.44
223	62850	17	4916.88	4915.70	1	0.92	138.01
224	62850	17	4916.88	4930.50	0	0.08	125.14
225	62851	17	4936.56	4915.00	1	0.04	212.18
226	62851	17	4936.56	4937.46	0	0.96	172.62
227	62852	17	4958.88	4958.88	0	1.00	130.55
228	62853	17	5077.04	5077.04	0	1.00	38.56
229	62854	17	5280.64	5280.64	0	1.00	30.13
230	62855	17	5482.60	5482.60	0	1.00	36.15
231	62856	17	5614.60	5601.77	0	0.88	25.98
232	62857	17	5837.48	5716.00	0	0.04	6.71
233	63076	31	5140.44	5183.47	0	0.68	7.32
234	63077	31	5277.84	5276.96	0	0.96	11.16

Figure 3.15: Example of a few lines in the GlacTable csv-file. For easy reading the glaciers with the same ID are given the same background color.

- A classification of which areas are debris-free and which are debris-covered.
- A high-resolution DEM.

A methodology to estimate ice-thickness is using the **GlabTop2 model**, which is described in [6].

### Older releases

For older SPHY model versions, you need to create the `glacfrac.map`. This map can be calculated from a vector file with glacier outlines (e.g. the [RGI](#)). Follow the steps below to convert the glacier outlines to the `glacfrac.map`:

1. In QGIS from the Processing toolbox, select the “v.to.rast.value” tool ([Figure 3.16](#)).
2. Select your glacier outlines as vector input layer and convert it to raster at the same extent of the clone map. Set the cellsize at a lower value than your model resolution. For example, if your model cell size is 200 m, select 20 m for the converted raster. Set 1 as the raster value.
3. The “nodata” values need to be reclassified to zeros. To do this use SAGA’s Reclassify tool ([Figure 3.17](#)) from the Processing toolbox.
4. In the dialog box ([Figure 3.18](#)) set all values to 0.0, and set “replace no data values” to “Yes”, set “new value for no data values” to 0.0 and set “replace other values” to “No”. Select an output filename and click “Run”.
5. Now we aggregate the fine resolution grid with glaciers to the model resolution. This can be done using the “r.resamp.stats” toolbox ([Figure 3.19](#)). In the dialog box, set the fine resolution glacier grid (result step 4) as input raster layer and choose aggregation method “average”. Use the extent from the clone map in the “Grass region cellsize” field, and set the cell size to the model resolution (clone).
6. The last step involves converting the map from step 5 into a PCRaster map.

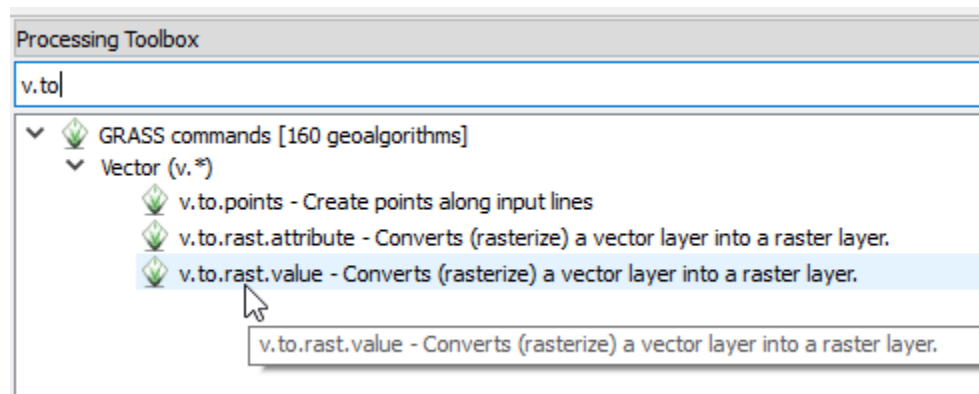


Figure 3.16: v.to.rast.value toolbox in QGIS.

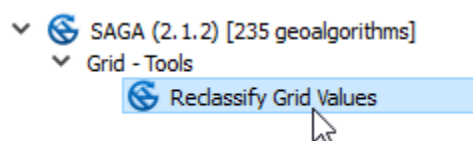


Figure 3.17: SAGA reclassify toolbox.

Reclassify Grid Values

Parameters Log Help Run as batch process...

Grid

Method

[0] single

old value (for single value change)

0.000000

new value (for single value change)

0.000000

operator (for single value change)

[0] =

minimum value (for range)

0.000000

maximum value (for range)

0.000000

new value(for range)

0.000000

operator (for range)

[0] <=

Lookup Table

Fixed table 3x3

operator (for table)

[0] min <= value < max

☒ replace no data values

new value for no data values

0.000000

☐ replace other values

new value for other values

0.000000

Reclassified Grid

[Save to temporary file]

☒ Open output file after running algorithm

0%

Run Close

Figure 3.18: Reclassify “nodata” values to zero using the SAGA reclassify toolbox.

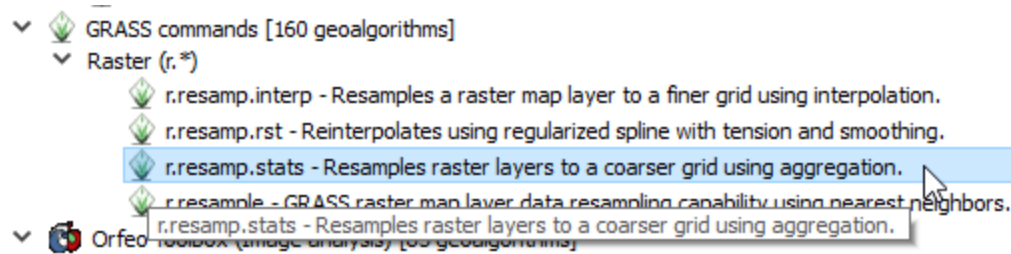


Figure 3.19: r.resamp.stats toolbox.

### 3.4.8 Other static input maps

Other static input maps (e.g. the soil maps and land use maps) can be created the same way as the DEM. The steps to create other static input maps are always the same:

1. Make sure that the CRS of your map in QGIS is the same as the CRS of your *clone map*. If this is not the case, then you need to re-project (QGIS warp) the map to the CRS of the clone map.
2. Check the metadata (same as Figure 3.11) of your map to see if the extent and spatial resolution are the same as the clone map. If that is not the case, then you need to resample the map using the same approach as illustrated in Figure 3.12 and Figure 3.13.
3. If your map is not in the PCRaster \*.map format, but for example in the \*.tif format, then you have to convert it using QGIS Translate.
4. Once your map is in the required PCRaster format, then the last step is to make sure the data type of the map matches the data type of the specific SPHY input map. For example, a DEM requires *scalar* data type, while a land use map should be *nominal*. Data types can easily be converted using the cmd line. A scalar map A.map can for example easily be converted to a nominal B.map using:

```
pcrcalc B.map = nominal(A.map)
```

An overview of the various SPHY model data types is shown in Table 3.2.

Table 3.2: Map data types used in the SPHY model.

Data type	Domain	Example
boolean	TRUE [1], FALSE [0]	clone.map
nominal	$[-2^{31}..2^{31}]$ integers	landuse.map, locations.map
scalar	$[-10^{37}..10^{37}]$ floats	dem.map, slope.map, sub_field.map, glacfrac.map
ldd	[1..9] integer of drain direction	ldd.map

## 3.5 Running a model

This section describes how you can run the SPHY model. How-to run the model depends on the way you have installed the SPHY model:

1. Installation via *Anaconda* or *pip*
2. Installation using the download from the *GitHub repository*

See ONE of the sections below for how-to run your SPHY model depending on the chosen installation method.

### 3.5.1 Running the model with an Anaconda or pip installation

If you have installed the SPHY model via Anaconda or pip, then you can run the SPHY model from any location on your PC; i.e. you do not need to refer to the physical address where the SPHY model source code resides. This is an advantage, because you do not need to copy all the model's source code files everytime you start a new project.

The steps that are needed to run the model are:

1. Create a folder somewhere on your hard drive, e.g. `c:\my_model`
2. After you have installed the model via Anaconda or pip, the `SPHY_config.cfg` configuration template can be found under:  
`PATH-TO-YOUR-PYTHON-INSTALLATION\Lib\site-packages\SPHY\SPHY_config.cfg`  
Copy `SPHY_config.cfg` to the folder you created under step 1
3. You can edit the `SPHY_config.cfg` and set all the variables in this configuration file
4. After you have edited and saved `SPHY_config.cfg`, you can run the model by:

```
python -m SPHY.main -i c:\my_model\SPHY_config.cfg
```

### 3.5.2 Running the model with the GitHub download installation

If you have downloaded the source code from the [SPHY model GitHub repository](#), then you always need to work from the folder where you have extracted the model's source code files; i.e. every time you start a new project you have to copy the source code files to the new project folder.

The steps that are needed to run the model are:

1. Create a folder somewhere on your hard drive, e.g. `c:\my_model`
2. Copy all the model source code files to the folder created under step 1
3. You can edit the `SPHY_config.cfg` and set all the variables in this configuration file
4. Save the `SPHY_config.cfg` file
5. You can now run the model via:

```
python c:\my_model\sphy.py -i c:\my_model\SPHY_config.cfg
```

## 3.6 Releases

### 3.6.1 2.2.1

#### About

The difference with respect to version 2.2.0 is that this version can be installed via *Anaconda* or *pip*.

#### Installation

Install via Anaconda:

```
conda install -c WilcoTerink SPHY=2.2.1
```

or install via pip:

```
pip install SPHY==2.2.1
```

## PCRaster

PCRaster 4.1.0 is required for this version of the SPHY model. PCRaster 4.1.0 can be downloaded [here](#).

### 3.6.2 2.2.0

#### About

#### Background

Glaciers in previous SPHY model releases were not mass conserving; i.e. they were implemented as a fixed mass generating glacier melt using a degree day factor. This means that:

- Rainfall on the glacier was not accounted for, and needs to be corrected for as a pre-processing step
- Snowfall, accumulation of snow, and snow melt on glacier grid cells was not taken into account
- Redistribution of ice from the accumulation to the ablation zone was not simulated; i.e. the dynamic retreat or advance of glaciers could not be simulated

Several methods exist to model glacier dynamics. E.g., [Bahr1997] performed a scaling analysis of the mass and momentum conservation equations and showed that glacier volumes can be related by a power law to more easily observed glacier surface areas. The disadvantage of this approach is that it is a lumped method and can therefore not be applied on a gridded surface. Another method is described by [Huss2010], who uses a  $\Delta h$ -parameterization to describe the spatial distribution of the glacier surface elevation change in response to a change in mass balance. It is an empirical glacier-specific function derived from observations in the past that can easily be applied to large samples of glaciers. However, a disadvantage of this methodology is that a large number of observations (multiple DEMs of the same glacier) from the past is required. [Weertman1957] and [Immerzeel2012b] described another method, known as Weertman sliding [Weertman1957]. The disadvantage of this method is that it is only feasible at high spatial model resolutions (<100 m), and it ignores viscous deformation. A good overview of the methods that are available for estimating ice volumes is well-described in [6].

#### Improved glacier module

The improved glacier module accounts for rainfall and snowfall onto the glacier, accumulation and melt of snow, and redistribution of ice from the accumulation to the ablation zone. These modifications allow glaciers to retreat over time if the melt rate is higher than the accumulation rate. The only data required for the improved module is:

1. A high-resolution DEM (higher resolution than the spatial resolution of your model)
2. A shapefile with the outlines of the glaciers within your area of interest. The [Randolph Glacier Inventory \(RGI\)](#) is a recommended source for this
3. A classification of which areas are debris-free and which are debris-covered

The new module uses the strength of combining PCRaster operations with [NumPy](#) and [pandas dataframe](#) functionalities. Instead of specifying the spatial distribution of glaciers on a PCRaster model grid, the glacier characteristics are now specified in a csv-file, and the glacier model grids are initialized from this csv-file. Each line in the csv-file





Figure 3.20: A glacier in New Zealand.

describes the glacier properties for a UID, which is a unique combination of model cell ID, glacier ID, glacier elevation, and a boolean describing if that UID is clean ice or debris-covered. The contents of this csv-file and the creation thereof are described in [here](#).

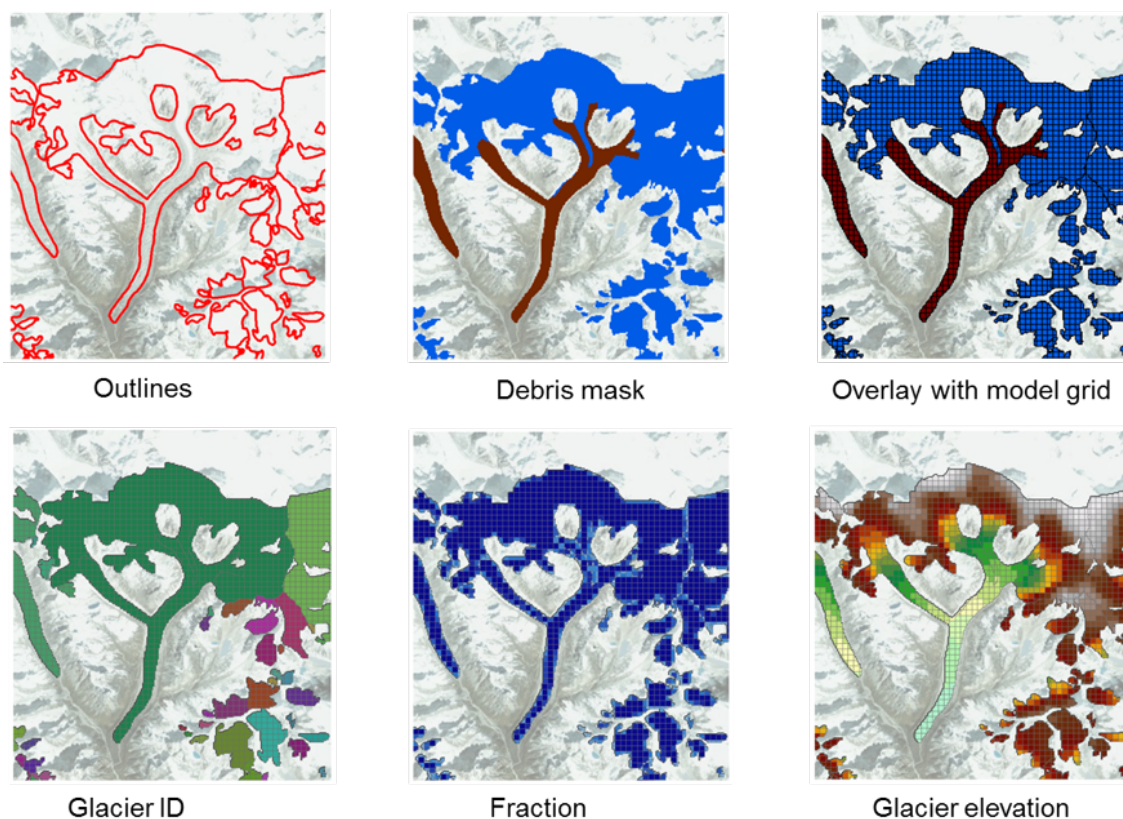


Figure 3.21: Illustration of steps to prepare input for the improved glacier module.

The new module basically comes down to overlaying your model grid with a higher resolution DEM to allow for the differentiation of multiple glacier fractions and associated elevations within one model grid cell. A more accurate representation of multiple glacier elevations within one model grid cell allows for a more accurate calculation of temperature, and thus melting rates. The more accurate temperature is calculated by lapsing the temperature from the model cell elevation to the elevation of the glacier fraction within that model grid cell. Snow and rain on the glacier are now accounted for, and melt rates adapt based on the presence of a dynamic snow pack; snow is melted before the glacier start melting.

Although the movement of glaciers is a continuous process, modelling this on a daily time-step is not feasible given the level of detailed geophysical information required for this. Glacier movement is the result of ice-redistribution. This is modelled by calculating the difference between the accumulation of snow and the melting of glacier ice on an annual basis (end of the hydrological year). Glaciers have an Equilibrium Line Altitude (ELA) above which the net accumulation of ice occurs and net melt (ablation) occurs below the ELA (Figure 3.22). If the annual glacier's mass balance is negative (more ablation than accumulation), then the accumulated mass (snow in accumulation zone) is redistributed over the ablation UIDs according to the ice volume distribution of the cells in the ablation zone. This process may eventually cause glaciers to disappear if the net ablation in a year is larger than the amount of ice available in the accumulation zone.

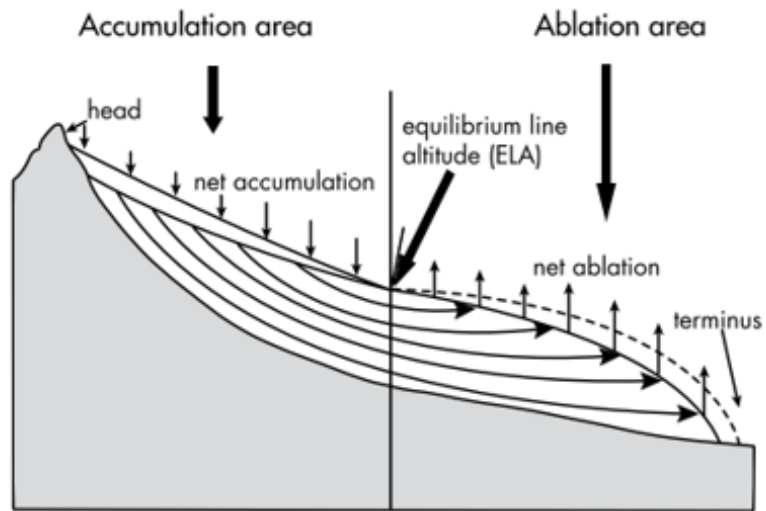


Figure 3.22: The Equilibrium Line Altitude (ELA), accumulation area, and ablation area of a glacier.

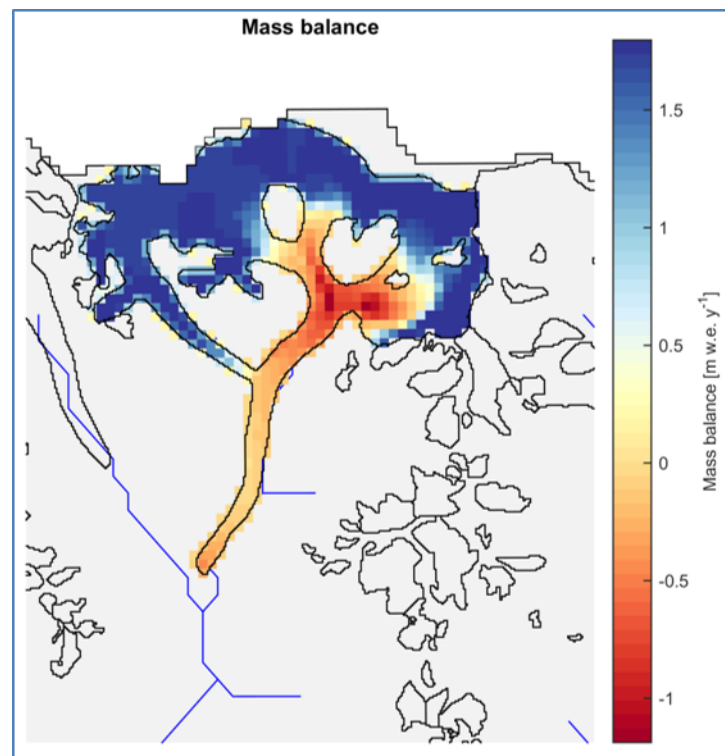


Figure 3.23: Example of the simulation of the glacier's average annual mass balance.

### Installation

This version can only be installed by downloading it from the [SPHY model GitHub repository](#). A direct link to this release can be found [here](#).

### PCRaster

PCRaster 4.1.0 is required for this version of the SPHY model. PCRaster 4.1.0 can be downloaded [here](#).

## 3.6.3 2.1.1

### About

The difference with respect to version 2.1.0 is that this version can be installed via *Anaconda* or *pip*.

### Installation

Install via Anaconda:

```
conda install -c WilcoTerink SPHY=2.1.1
```

or install via pip:

```
pip install SPHY==2.1.1
```

### PCRaster

PCRaster 4.1.0 is required for this version of the SPHY model. PCRaster 4.1.0 can be downloaded [here](#).

## 3.6.4 2.1.0

### About

### Background

Although most dominant hydrological processes were integrated in previous SPHY release versions, the model was not yet capable of simulating reservoir inflow and outflow. In version 2.1.0, a reservoir module was developed for the SPHY model.

Depending on the availability and quality of reservoir data, the user has two options for simulating reservoir in- and outflow in the SPHY model, being:

1. A *simple* reservoir scheme
2. An *advanced* reservoir scheme

The advanced reservoir scheme focuses on a target release, while the simple scheme calculates the outflow based on the actual and maximum reservoir storage. The reservoir schemes can be specified for each individual reservoir, meaning that simple and advanced reservoirs can exist independently from each other. The concepts of these schemes are described in the following two sections.



### Simple reservoir scheme

Many studies have implemented reservoir operation schemes into large-scale hydrological models. [Meigh1999] for example used a grid-based model and incorporated reservoir outflow using the following assumptions. For outflow regulation reservoirs, the outflow ( $Q$ ) was proportional to the current storage ( $S$ ) raised to the power of 1.5. For water storage reservoirs, any water released was used to meet demands and once the reservoir was filled, the outflow equaled the amount spilled [Hanasaki2006]. A study by [Doll2003] accounted for reservoir flow regulation [Hanasaki2006], and modified the equation of [Meigh1999] to:

$$Q_{out} = k_r \cdot S_{act} \cdot \left( \frac{S_{act}}{S_{max}} \right)^{1.5}$$

with:

$$Q_{out} = \text{reservoir outflow } [m^3 d^{-1}] \quad (3.1)$$

$$k_r = \text{outflow coefficient } [d^{-1}] \quad (3.2)$$

$$S_{act} = \text{actual reservoir storage } [m^3] \quad (3.3)$$

$$S_{max} = \text{maximum reservoir storage } [m^3]$$

(3.4)

This parametrization was developed for global lakes, but is also applied for reservoirs because of a lack of information on their management [Hanasaki2006]. For this reason this reservoir outflow equation was implemented in the SPHY model, and is referred to as the simple reservoir scheme hereafter. The outflow coefficient ( $k$ ) is generally used as a calibration parameter. An example of the reservoir inflow, outflow and storage is illustrated in [Figure 3.24](#) for the simple reservoir scheme.

### Advanced reservoir scheme

If reservoir management information is available, then it is recommended to use the advanced reservoir scheme. This scheme uses the same approach as is implemented in the SWAT model [Neitsch\_2000]. Within the advanced reservoir scheme the user can define a target reservoir release volume, which is season dependent. The user can define two different seasons per year, being a flood and dry season respectively. The target releases that need to be defined are the *MAX\_FLOW* (maximum flow) for the flood season, and the *DEM\_FLOW* (demand flow) for the dry season.

In case of floods it is important to have a large storage capacity available. Therefore, the model tries to release the maximum amount of water from the reservoir during the flood season. The maximum amount of water that can be released depends on the user defined *MAX\_FLOW*, the actual reservoir storage, and the maximum amount of water that is available between the emergency spillway and principal spillway. During the dry season, the model aims to keep the reservoir filled in order to have water available for its water users downstream. During this season, the reservoir outflow depends on the user defined *DEM\_FLOW* (demand flow), the actual reservoir storage, and the maximum amount of water that is available between the emergency spillway and principal spillway. Reservoir outflow for both seasons is thus calculated as:

$$Q_{out} = F \cdot \left( \frac{S_{act}}{RES_{evol} - RES_{pvol}} \right)$$

with:

$$Q_{out} = \text{reservoir outflow } [m^3 d^{-1}] \quad (3.5)$$

$$F = \text{target release, being either } MAX\_FLOW \text{ or } DEM\_FLOW [m^3 d^{-1}] \quad (3.6)$$

$$S_{act} = \text{actual reservoir storage } [m^3] \quad (3.7)$$

$$RES_{evol} = \text{reservoir storage capacity at emergency spillway } [m^3] \quad (3.8)$$

$$RES_{pvol} = \text{reservoir storage capacity at principal spillway } [m^3] \quad (3.9)$$

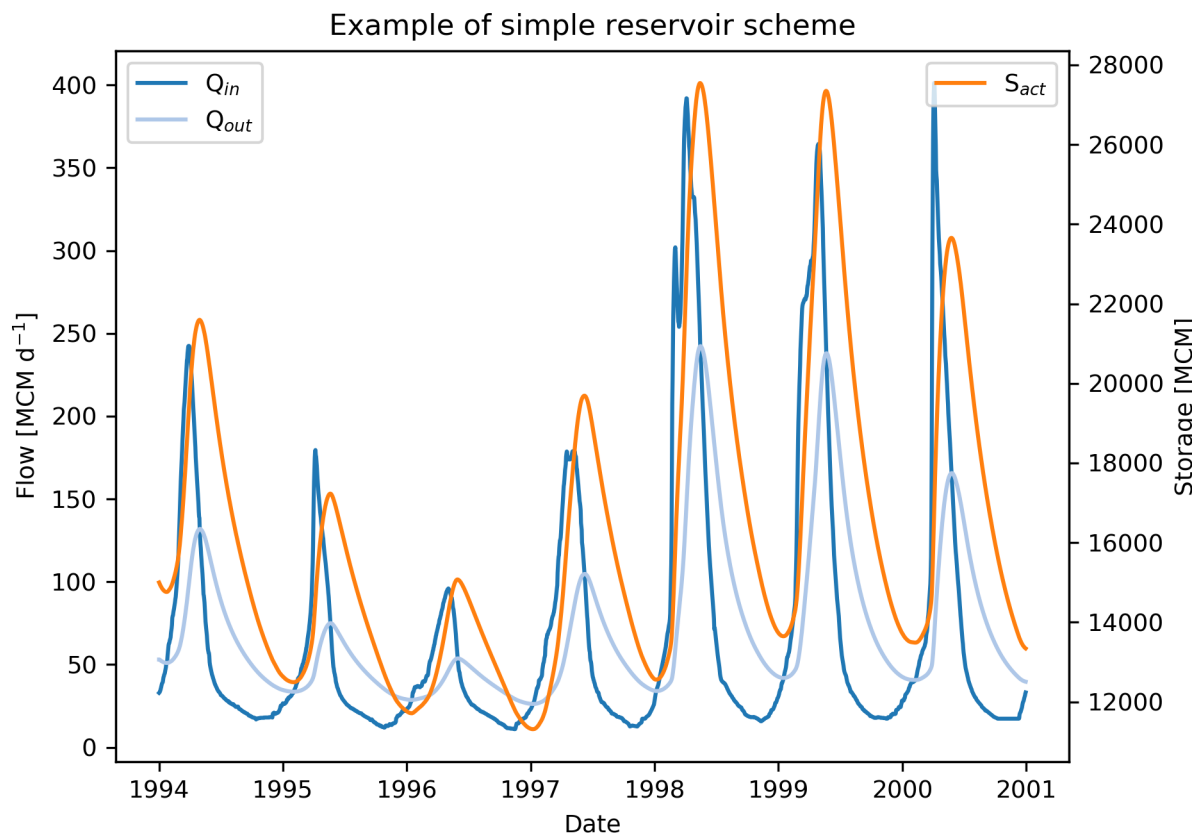


Figure 3.24: Example of reservoir inflow ( $Q_{in}$ ), outflow ( $Q_{out}$ ), and storage ( $S_{act}$ ) as modelled by the simple reservoir scheme.

An example of the reservoir inflow, outflow and storage is illustrated in [Figure 3.25](#) for the advanced reservoir scheme.

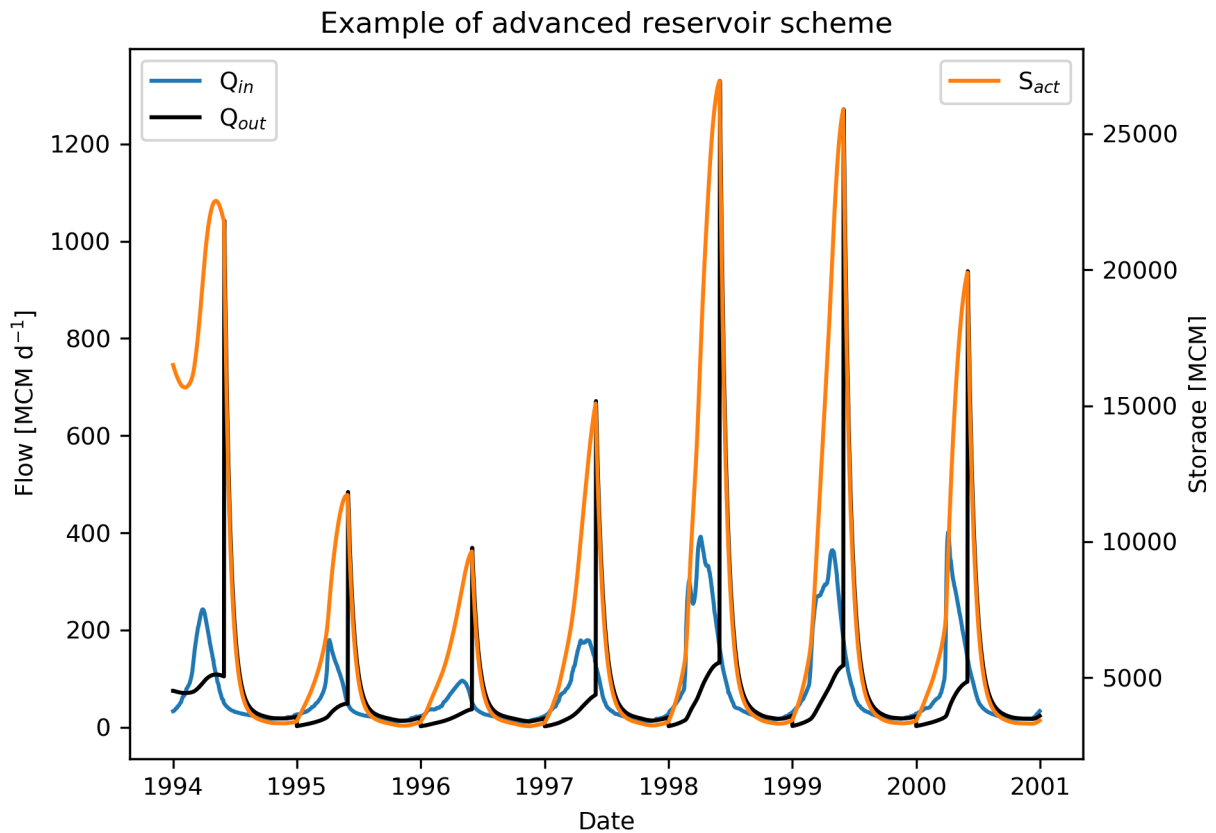


Figure 3.25: Example of reservoir inflow ( $Q_{in}$ ), outflow ( $Q_{out}$ ), and storage ( $S_{act}$ ) as modelled by the advanced reservoir scheme. In this example the flood season starts the first of June, and ends by the end of December.

## Installation

This version can only be installed by downloading it from the [SPHY model GitHub repository](#). A direct link to this release can be found [here](#).

## PCRaster

PCRaster 4.1.0 is required for this version of the SPHY model. PCRaster 4.1.0 can be downloaded [here](#).

## 3.6.5 2.0.4

### About

The difference with respect to version 2.0.3 is that this version can be installed via [Anaconda](#) or [pip](#).

### Installation

Install via Anaconda:

```
conda install -c WilcoTerink SPHY=2.0.4
```

or install via pip:

```
pip install SPHY==2.0.4
```

### PCRaster

PCRaster 4.1.0 is required for this version of the SPHY model. PCRaster 4.1.0 can be downloaded [here](#).

## 3.6.6 2.0.3

### About

Bug fix in the reporting function and in the calculation of snow storage and melt water stored in snow pack.

### Installation

This version can only be installed by downloading it from the [SPHY model GitHub repository](#). A direct link to this release can be found [here](#).

### PCRaster

PCRaster 4.1.0 is required for this version of the SPHY model. PCRaster 4.1.0 can be downloaded [here](#).

## 3.6.7 2.0.2

### About

Fixed critical error in the calculation of lateral flow. Lateral flow was originally calculated as:

$$(rootlat + rootdrain) \cdot \left(1 - \exp\frac{-1}{rootTT}\right)$$

This resulted in a continuous outflow of lateral flow with small values for  $rootTT$ , even if there was no lateral flow generated in the current timestep. This equation has been fixed to:

$$rootlat \cdot \left(1 - \exp\frac{-1}{rootTT}\right) + rootdrain \cdot \exp\frac{-1}{rootTT}$$

### Installation

This version can only be installed by downloading it from the [SPHY model GitHub repository](#). A direct link to this release can be found [here](#).



## PCRaster

PCRaster 4.1.0 is required for this version of the SPHY model. PCRaster 4.1.0 can be downloaded [here](#).

### 3.6.8 2.0.1

#### About

There was an error in the calculation of the fractional rain. In *version 2.0.0* the fractional rain was calculated by multiplying the rain with the rain fraction. This has been fixed by multiplying the rain with  $(1 - \textit{glacfrac})$ .

#### Installation

This version can only be installed by downloading it from the [SPHY model GitHub repository](#). A direct link to this release can be found [here](#).

## PCRaster

PCRaster 4.1.0 is required for this version of the SPHY model. PCRaster 4.1.0 can be downloaded [here](#).

### 3.6.9 2.0.0

#### About

SPHY version 2.0.0 [31] is the first official release of the SPHY model, and was published in [Geoscientific Model Development](#). All model concepts and processes are described in this [PDF](#).

#### Installation

This version can only be installed by downloading it from the [SPHY model GitHub repository](#). A direct link to this release can be found [here](#).

## PCRaster

PCRaster 4.1.0 is required for this version of the SPHY model. PCRaster 4.1.0 can be downloaded [here](#).

## 3.7 FAQ

A Google user group for the SPHY model is available [here](#). You can use this group to post questions and answers related to the source code, available plugins, input and output formats, calibration, applications, and suggestions for improvements.

## 3.8 References



## CHAPTER 4

---

### Copyright

---

Copyright (C) 2013 Wilco Terink. The Spatial Processes in HYdrology (SPHY) model is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.



---

## Bibliography

---

- [1] N Batjes, K Dijkshoorn, V van Engelen, G Fischer, A Jones, L Montanarella, M Petri, S Prieler, E Teixeira, D Wiberg, and X Shi. Harmonized World Soil Database (version 1.2). Technical Report, FAO and IIASA, Rome, Italy and Laxenburg, Austria, 2012.
- [2] Lisa M Bramer, Brian K Hornbuckle, and Petruța C Caragea. How Many Measurements of Soil Moisture within the Footprint of a Ground-Based Microwave Radiometer Are Required to Account for Meter-Scale Spatial Variability? *Vadose Zone Journal*, 2013. URL: <https://www.soils.org/publications/vzj/abstracts/12/3/vzj2012.0100>, doi:10.2136/vzj2012.0100.
- [3] Toby N Carlson and David A Ripley. On the relation between NDVI, fractional vegetation cover, and leaf area index. *Remote Sensing of Environment*, 62(3):241–252, dec 1997. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0034425797001041>, doi:10.1016/S0034-4257(97)00104-1.
- [4] EEA. EU-DEM layers, Copernicus data and information funded by the European Union, European Environmental Agency. Technical Report, European Environmental Agency, 2014.
- [5] L Foglia, M C Hill, S W Mehl, and P Burlando. Sensitivity analysis, calibration, and testing of a distributed hydrological model using error-based weighting and one objective function. *Water Resources Research*, 45(6):n/a—n/a, jun 2009. URL: <http://doi.wiley.com/10.1029/2008WR007255>, doi:10.1029/2008WR007255.
- [6] H Frey, H Machguth, M Huss, C Huggel, S Bajracharya, T Bolch, A Kulkarni, A Linsbauer, N Salzmann, and M Stoffel. Estimating the volume of glaciers in the Himalayan-Karakoram region using different methods. *The Cryosphere*, 8(6):2313–2333, 2014. doi:10.5194/tc-8-2313-2014.
- [7] Kaushik Gopalan, Nai-Yu Wang, Ralph Ferraro, and Chuntao Liu. Status of the TRMM 2A12 Land Precipitation Algorithm. *Journal of Atmospheric and Oceanic Technology*, 27(8):1343–1354, aug 2010. URL: <http://journals.ametsoc.org/doi/abs/10.1175/2010JTECHA1454.1>, doi:10.1175/2010JTECHA1454.1.
- [8] W W Immerzeel and M F P Bierkens. Asia’s water balance. *Nature Geoscience*, 5(12):841–842, nov 2012. URL: <http://www.nature.com/doi/abs/10.1038/ngeo1643>, doi:10.1038/ngeo1643.
- [9] W W Immerzeel and P Droogers. Calibration of a distributed hydrological model based on satellite evapotranspiration. *Journal of Hydrology*, 349(3-4):411–424, feb 2008. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0022169407006944>, doi:10.1016/j.jhydrol.2007.11.017.
- [10] W W Immerzeel, P Droogers, S M de Jong, and M F P Bierkens. Large-scale monitoring of snow cover and runoff simulation in Himalayan river basins using remote sensing. *Remote Sensing of Environment*, 113(1):40–49, 2009. doi:10.1016/j.rse.2008.08.010.

- [11] W W Immerzeel, A F Lutz, and P Droogers. Climate Change Impacts on the Upstream Water Resources of the Amu and Syr Darya River Basins. Technical Report, FutureWater, Wageningen, 2012.
- [12] Walter W Immerzeel, Ludovicus P H van Beek, and Marc F P Bierkens. Climate change will affect the Asian water towers. *Science (New York, N.Y.)*, 328(5984):1382–1385, jun 2010. URL: <http://www.ncbi.nlm.nih.gov/pubmed/20538947>, doi:10.1126/science.1183188.
- [13] Derek Karssenberg. The value of environmental modelling languages for building distributed hydrological models. *Hydrological Processes*, 16(14):2751–2766, 2002. URL: <http://doi.wiley.com/10.1002/hyp.1068>, doi:10.1002/hyp.1068.
- [14] Derek Karssenberg, Peter A Burrough, Raymond Sluiter, and Kor de Jong. The PCRaster Software and Course Materials for Teaching Numerical Modelling in the Environmental Sciences. *Transactions in GIS*, 5(2):99–110, 2001. URL: <http://doi.wiley.com/10.1111/1467-9671.00070>, doi:10.1111/1467-9671.00070.
- [15] Derek Karssenberg, Oliver Schmitz, Peter Salamon, Kor de Jong, and Marc F P Bierkens. A software framework for construction of process-based stochastic spatio-temporal models and data assimilation. *Environmental Modelling & Software*, 25(4):489–502, apr 2010. URL: <http://linkinghub.elsevier.com/retrieve/pii/S1364815209002643>, doi:10.1016/j.envsoft.2009.10.004.
- [16] J J Lambert, J Daroussin, M Eimberck, C Le Bas, M Jamagne, D King, and L Montanarella. Soil Geographical Database for Eurasia & The Mediterranean. Instructions Guide for Elaboration at scale 1:1,000,000 version 4.0. EUR 20422 EN. Technical Report, JRC, Ispra, Italy, 2003.
- [17] Xu Liang, Dennis P Lettenmaier, Eric F Wood, and Stephen J Burges. A simple hydrologically based model of land surface water and energy fluxes for general circulation models. 1994. URL: <http://doi.wiley.com/10.1029/94JD00483>, doi:10.1029/94JD00483.
- [18] Xu Liang, Eric F Wood, and Dennis P Lettenmaier. Surface soil moisture parameterization of the VIC-2L model: Evaluation and modification. *Global and Planetary Change*, 13(1-4):195–206, 1996. URL: <http://www.sciencedirect.com.proxy.library.uu.nl/science/article/pii/0921818195000461>, doi:10.1016/0921-8181(95)00046-1.
- [19] A F Lutz, P Droogers, and W W Immerzeel. Climate Change Impact and Adaptation on the Water Resources in the Amu Darya and Syr Darya River Basins. Technical Report, FutureWater, Wageningen, 2012.
- [20] A F Lutz, W W Immerzeel, A Gobiet, F Pellicciotti, and M F P Bierkens. Comparison of climate change signals in CMIP3 and CMIP5 multi-model ensembles and implications for Central Asian glaciers. *Hydrology and Earth System Sciences*, 17(9):3661–3677, sep 2013. URL: <http://www.hydrol-earth-syst-sci.net/17/3661/2013/>, doi:10.5194/hess-17-3661-2013.
- [21] A F Lutz, W W Immerzeel, and P D A Kraaijenbrink. Gridded Meteorological Datasets and Hydrological Modelling in the Upper Indus Basin. FutureWater Report 130. Technical Report, FutureWater, Wageningen, The Netherlands, 2014.
- [22] A F Lutz, W W Immerzeel, A B Shrestha, and M F P Bierkens. Consistent increase in High Asia’s runoff due to increasing glacier melt and precipitation. *Nature Climate Change*, 4(7):587–592, jun 2014. URL: <http://www.nature.com/doifinder/10.1038/nclimate2237>, doi:10.1038/nclimate2237.
- [23] Gerald A Meehl, Curt Covey, Karl E Taylor, Thomas Delworth, Ronald J Stouffer, Mojib Latif, Bryant McAvaney, and John F B Mitchell. THE WCRP CMIP3 Multimodel Dataset: A New Era in Climate Change Research. *Bulletin of the American Meteorological Society*, 88(9):1383–1394, sep 2007. URL: <http://journals.ametsoc.org/doi/abs/10.1175/BAMS-88-9-1383>, doi:10.1175/BAMS-88-9-1383.
- [24] R B Myneni and D L Williams. On the relationship between FAPAR and NDVI. *Remote Sensing of Environment*, 49(3):200–211, sep 1994. URL: <http://linkinghub.elsevier.com/retrieve/pii/0034425794900167>, doi:10.1016/0034-4257(94)90016-7.
- [25] J E Nash and J V Sutcliffe. River flow forecasting through conceptual models. Part I - A discussion of principles. *Journal of Hydrology*, 10(3):282–290, apr 1970. URL: <http://linkinghub.elsevier.com/retrieve/pii/0022169470902556>, doi:10.1016/0022-1694(70)90255-6.

- [26] B Samain, G W H Simons, M P Voogt, W Defloor, N.-J. Bink, and V R N Pauwels. Consistency between hydrological model, large aperture scintillometer and remote sensing based evapotranspiration estimates for a heterogeneous catchment. *Hydrology and Earth System Sciences*, 16(7):2095–2107, jul 2012. URL: <http://www.hydrol-earth-syst-sci.net/16/2095/2012/>, doi:10.5194/hess-16-2095-2012.
- [27] Neil Schaner, Nathalie Voisin, Bart Nijssen, and Dennis P Lettenmaier. The contribution of glacier melt to streamflow. *Environmental Research Letters*, 7(3):34029, sep 2012. URL: <http://stacks.iop.org/1748-9326/7/i=3/a=034029?key=crossref.1d60a5b89febe92ea66fea367de538be>, doi:10.1088/1748-9326/7/3/034029.
- [28] Justin Sheffield, Gopi Goteti, and Eric F Wood. Development of a 50-Year High-Resolution Global Dataset of Meteorological Forcings for Land Surface Modeling. *Journal of Climate*, 19(13):3088–3111, jul 2006. URL: <http://journals.ametsoc.org/doi/abs/10.1175/JCLI3790.1>, doi:10.1175/JCLI3790.1.
- [29] Annina Sorg, Tobias Bolch, Markus Stoffel, Olga Solomina, and Martin Beniston. Climate change impacts on glaciers and runoff in Tien Shan (Central Asia). *Nature Climate Change*, 2(10):725–731, jul 2012. URL: <http://www.nature.com/doifinder/10.1038/nclimate1592>, doi:10.1038/nclimate1592.
- [30] Karl E Taylor, Ronald J Stouffer, and Gerald A Meehl. An Overview of CMIP5 and the Experiment Design. *Bulletin of the American Meteorological Society*, 93(4):485–498, apr 2012. URL: <http://journals.ametsoc.org/doi/abs/10.1175/BAMS-D-11-00094.1>, doi:10.1175/BAMS-D-11-00094.1.
- [31] W Terink, A F Lutz, G W H Simons, W W Immerzeel, and P Droogers. SPHY v2.0: Spatial Processes in HYdrology. *Geoscientific Model Development*, 8(7):2009–2034, jul 2015. URL: <http://www.geosci-model-dev.net/8/2009/2015/>, doi:10.5194/gmd-8-2009-2015.
- [32] Wilco Terink, Hidde Leijnse, Gé van den Eertwegh, and Remko Uijlenhoet. Spatial resolutions in areal rainfall estimation and their impact on hydrological simulations of a lowland catchment. *Journal of Hydrology*, 563:319–335, aug 2018. URL: <https://www.sciencedirect.com/science/article/pii/S002216941830372X>, doi:10.1016/J.JHYDROL.2018.05.045.
- [33] Compton J Tucker. Red and photographic infrared linear combinations for monitoring vegetation. *Remote Sensing of Environment*, 8(2):127–150, 1979. URL: <http://linkinghub.elsevier.com/retrieve/pii/0034425779900130>, doi:10.1016/0034-4257(79)90013-0.
- [34] USGS. Landsat 8: U.S. Geological Survey Fact Sheet 2013–3060. Technical Report, USGS, 2013. URL: <http://pubs.usgs.gov/fs/2013/3060/>.
- [35] Akiyo Yatagai, Kenji Kamiguchi, Osamu Arakawa, Atsushi Hamada, Natsuko Yasutomi, and Akio Kitoh. APHRODITE: Constructing a Long-Term Daily Gridded Precipitation Dataset for Asia Based on a Dense Network of Rain Gauges. *Bulletin of the American Meteorological Society*, 93(9):1401–1415, sep 2012. URL: <http://journals.ametsoc.org/doi/abs/10.1175/BAMS-D-11-00122.1>, doi:10.1175/BAMS-D-11-00122.1.
- [36] M Van Genuchten. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil science society of America journal*, 44(5):892–898, 1980.