NTNU
Norwegian University of
Science and Technology

Evaluation of snow simulations in SHyFT

Joseba Ullibarri Lombraña

Hydropower Development
Submission date: June 2017
Supervisor: Knut Alfredsen, IBM
Co-supervisor: Oddbjørn Bruland, IBM
Yisak Sultan Abdella, Statkraft

Norwegian University of Science and Technology
Department of Civil and Environmental Engineering

PREFACE

This Master's Thesis entitled "Evaluation of snow simulations in SHyFT" is submitted to the Department of Hydraulic and Environmental Engineering at the Norwegian University of Science and Technology, NTNU, Trondheim, Norway, as a partial fulfilment of the M.Sc. degree requirements in Hydropower Development.

This report is carried out as an outcome of the work developed from January to June 2017 at NTNU under the supervision of Prof. Knut Aldredsen, together with co-supervisor and other colleagues.

Finally, I hereby declare that the work presented here is my own and all outside contributions have been properly acknowledged.

Joseba Ullibarri Lombraña

June 2017

Trondheim, Norway

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor, Professor Knut Alfredsen at the Department of Hydraulic and Environmental Engineering, NTNU, first of all, for helping me finding a Master's Thesis in collaboration with Statkraft. Thank you for offering me the opportunity to work closer to the real world outside the pure academic and also for the effort put in the preparation of this thesis. I appreciate the commitment throughout the semester and your availability to arrange meetings for related discussions.

I would like to acknowledge my co-supervisor, Professor Oddbjørn Bruland, for sharing his unlimited-seeming knowledge and experience on the topic and his support at the meetings and during the semester.

Next, I am deeply grateful to Dr. Yisak Sultan Abdella from Statkraft and Mr. Kuganesan Sivasubramaniam, PhD at the Department, for their assistance while using SHyFT. Thank you, Kuganesan, for your help and infinite patience; I wish you the best during your stay in Australia.

In addition, I would like to thank Mr. Abebe Girmay Adera, current Research Assistant at the Department, for his support and interest showed on my work.

Last but not least, I would like to say thank you to my parents, who have and will always be there for me, for better or worse, supporting me no matter what. This one is for you, Aita y Ama. I love you.

NTNU
Norwegian University of
Science and Technology

Faculty of Engineering
Science and Technology
Department of Hydraulic and
Environmental Engineering

M.Sc. THESIS IN

HYDROPOWER DEVELOPMENT

Candidate: **Joseba Ullibarri Lombraña**

Title: **Evaluation of snow simulations in SHyFT.**

1 BACKGROUND

Snow is a very important component in the hydrological cycle in Norway and crucial for determining reservoir operation during the spring flood to ensure full reservoir and as little flood spill as possible. The Statkraft Hydrological Forecasting Toolbox (SHyFT) is a newly developed hydrological toolbox that is used for forecasting inflow in the Statkraft system. This is a flexible system in which model can be custom designed for various purposes. The SHyFT toolbox currently have three different methods for simulating snow accumulation and storage, and these are not yet evaluated with snow data. The purpose of this master thesis is to evaluate the SHyFT snow routines against observed snow data from satellite images and snow measurements in the field.

2 MAIN QUESTIONS FOR THE THESIS

The main questions for the thesis can be stated as follows:

1. Prepare the data needed to calibrate the SHyFT model for the Nea-Nidelva catchment. This includes climatic data from observation sites in the catchment and other climatic data derived from other stations. Collect the data needed for evaluating the snow simulations, including both satellite and from measurement campaigns in the field. Decide on the periods that should be used for calibration and evaluation based on the available data.

2. Calibrate SHyFT for Nidelva for all three snow routines. Compare the calibrations and evaluate their goodness. Evaluation should be done on standard statistical parameters measuring runoff distribution and also parameters measuring runoff volume.

3. Compare simulated snow from the three setups from 2) against each other and against observed snow data. Perform a statistical analysis to evaluate both the temporal and spatial accuracy of the simulated snow. Measures of goodness of fit both for temporal and spatial variation should be decided and used in this task. Discrepancies should be quantified and evaluations should be done to try to identify reasons for any differences between observed and simulated snow cover and water equivalent such as autumn snow start errors or errors in snow volume over the winter.

3 SUPERVISION, DATA AND INFORMATION INPUT

Professor Knut Alfredsen, Professor Oddbjørn Bruland, NTNU and Dr. Yisak Sultan Abdella, Statkraft will be advisors on the project. Professor Knut Alfredsen will handle the formalities related to the supervision.

Discussion with and input from colleagues and other research or engineering staff at NTNU, SINTEF, power companies or consultants are recommended. Significant inputs from others shall, however, be referenced in a convenient manner.

The research and engineering work carried out by the candidate in connection with this thesis shall remain within an educational context. The candidate and the supervisors are therefore free to introduce assumptions and limitations, which may be considered unrealistic or inappropriate in a contract research or a professional engineering context.

4 REPORT FORMAT AND REFERENCE STATEMENT

The thesis report shall be in the format A4. It shall be typed by a word processor and figures, tables, photos etc. shall be of good report quality. The report shall include a summary, a table of content, a list of literature formatted according to a common standard and other relevant references. A signed statement where the candidate states that the presented work is his own and that significant outside input is identified should be included.

The report shall have a professional structure, assuming professional senior engineers (not in teaching or research) and decision makers as the main target group.

The thesis shall be submitted no later than 10th of June 2017.

Trondheim 10th of January 2017

____

Knut Alfredsen

Professor

ABSTRACT

Snow is a very important component in the hydrological cycle in Norway and crucial for determining reservoir operation during the spring flood to ensure full reservoir and as little flood spill as possible. The Statkraft Hydrological Forecasting Toolbox (SHyFT) is a newly developed hydrological toolbox that is used for forecasting inflow in the Statkraft system. This is a flexible system in which model can be custom designed for various purposes.

This study calibrates, runs and evaluates the three snow routines implemented in SHyFT against field observed snow data provided by Statkraft. The three routines are Gamma Snow, HBV Snow and Skaugen Snow.

Part of the study is based on the process of finding the closest and most representative cells of the grid in the catchment of the study area (the Nea-Nidelva river basin), in order to later make the comparison between observed and simulated snow data. To do that, the snow transects where Statkraft made the snow field measurements are analysed.

Next, the calibration of the model and their validations are performed by simulating runoff and comparing them to the unregulated observed discharge data from Aune gauging station (SeNorge). The calibration period, where the calibration of the model takes place, is from 01/09/2012 to 01/09/2014 and the validation, where the runoff is simulated without having calibrated the model for that period, is from 01/09/2014 to 01/09/2015. The calibration results showed a Nash-Shutcliffe efficiency criteria (R^2) of 0.733 for Gamma Snow, 0.755 for HBV Snow and 0.784 for Skaugen Snow. However, the results show that Gamma Snow performs better simulations (or closer to the observed data) that HBV Snow and Skaugen Snow.

Regarding the snow results, SHyFT codes are used to extract the SWE from the specific grid cells. All the models show both similarities and discrepancies between the observed SWE data and the simulated results. The comparison was carried out by observing the percentage of the difference between observed and simulated discharge per transect, and it is concluded that all of the three models present flaws and that the simulations were sometimes poor. Some other reasons why these simulations were not too good are commented, such as redistribution of the snow by wind, elevation, or orientation.

The models behaved in a way in 2013 and 2015 that they presented close difference values of SWE, whereas in 2014, the results of each model differ significantly from each other.

It can be said that the calibration of the model was successful and the R2 values were good, the runoff simulations were acceptable and the snow simulations were somehow poor.

Table of contents

List of Appendix

List of figures

List of tables

Abbreviations

API	Application Programming Interface
GIS	Geographic Information System
HBV	Hydrologiska Byråns Vattenbalansavdelning
masl	meters above sea level
NSE	Nash Sutcliffe Efficiency
NetCDF	Network Common Data Frame
NVE	Norges vassdrags- og energidirektorat
PDF	Probability Density Function
ptgsk	Priestley Taylor Gamma Snow Kirchner
pthsk	Priestley Taylor HBV Snow Kirchner
ptssk	Priestley Taylor Skaugen Snow Kirchner
SCA	Snow Covered Area
SDC	Snow Depletion Curve
SMHI	Sveriges Meteorologiska och Hydrologiska Institut
SWE	Snow Water Equivalent
YAML	Yet Another Markup Language

1. INTRODUCTION

1.1. Background

Snow is an important and non-easy to handle element in hydrological modelling. In many high-altitude regions of the Earth, snow is a fundamental parameter of the hydrological cycle; in Norway, approximately 30% of the annual precipitation falls as snow. Snow plays hence an essential role on reservoirs management and operational strategies. Furthermore, snow is not only relevant for water resources management (water supply, hydropower production, irrigation, transport…) but also for many aspects related to economy and society, such as outdoor activities, tourism, infrastructure or safety.

It is then easy to realize that being able to access different snow data is highly interesting for hydrologists and hydropower companies (also for many other sectors, as previously said). Moreover, being able to predict future short term forecasts or long term predictions is one of the main goals, and hydrological modelling is the way to achieve it.

The Statkraft Hydrological Forecasting Toolbox (SHyFT) is a newly developed hydrological toolbox that is used for forecasting inflow in the Statkraft system. This is a flexible system in which model setups (called stacks) can be developed and used for different simulation tasks. The model currently has three different methods for simulating snow accumulation and storage, and they are yet to be evaluated.

Available snow data is usually obtained by satellite digital data and/or field measurements. The typical pursued data is Snow Covered Area, Snow depth and, as it is directly related, Snow Water Equivalent.

1.2. Objective

The main objective of this report is to evaluate the three different snow routines implemented in SHyFT (Statkraft's Hydrological Forecasting Toolbox) against observed snow data such as satellite images or field measurements.

1.3. Structure of the report

The report is divided into 10 chapters:

1. Introduction: short background of the topic, main objective of the thesis and presentation of the organization of the report.
2. Literature review: information about the three different snow models in SHyFT.
3. Study area: description of the region of interest.
4. Data and work methodology: description of the working steps throughout the study.
5. SHyFT: introduction to the Statkraft's Hydrological Forecasting Toolbox.
6. Model calibration and validation: results of the process of calibration and validation of the three models.
7. Snow simulations: results of the snow data obtained from SHyFT.
8. Results and discussion: anaylisis of the results and further comments.
9. Conclusions.
10. References: bibliography.

2. LITERATURE REVIEW

This section describes the three snow routines implemented in SHyFT.

2.1. Gamma snow model

The Gamma –Snow routine is a snowmelt routine based on the gamma snow distribution. The gamma snow distribution simulates the evolution of snow storage within each cell of the grid in the model. It is an important aspect of the snow melt routines and describes the relationship between the Snow Covered Area (SCA) and the mass balance of the snowpack, represented by a Snow Depletion Curve (SDC).

In numerical hydrological snow models, the SDC is often used as a tool to explain the spatial variability in the snowpack within the elements of the model. It describes how SCA reduces gradually through the melt season and, moreover, relates the SCA to its respective Snow Water Equivalent (SWE), which is a model derived response variable.

The SDC can be understood as [1]:

- The cumulative probability distribution characterizing the subgrid heterogeneity of point snow storage X at the start of the melt season (left axis in Figure 2.1).
- The SCA as a function of accumulated snow melt depth λ (right axis in Figure 2.1).

In Figure 2.1, the parameters A_o, m and cv remain constant during the melt season. The λ state also divides the initial snowpack into already melted and still remaining snow. Of these four, only λ changes throughout the melt season. Gamma is a shape-scale distribution in which λ and m occur only as a ratio; thus, any change in m can be counterbalanced by a similar scaling of $\lambda(t)$for all t, without changing shape of the SDC.

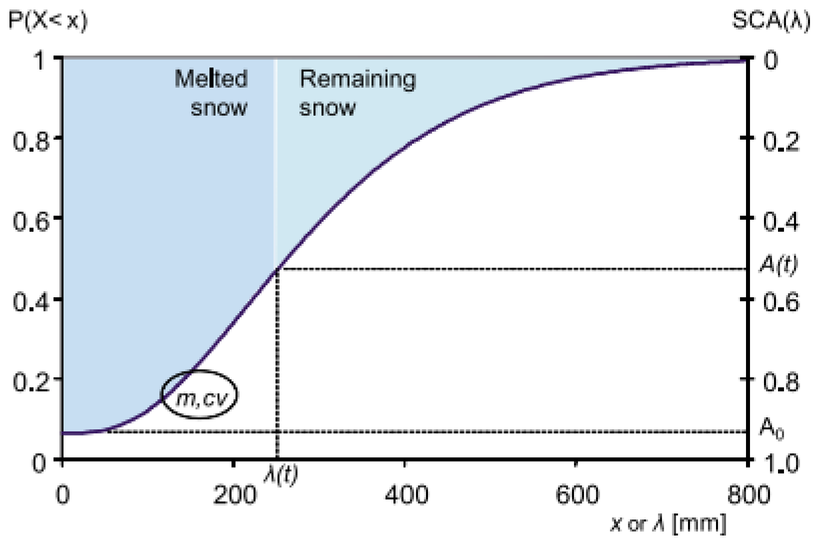


Figure 2.1. Snow Depletion Curve (SDC).

The principle of the Snow Depletion Curve as it is applied in each grid cell individually can be stated as follows:

At the start of the melt season, the SDC represents the spatial heterogeneity of the point snow storage x within the grid cell. It is parameterized by the average storage m, the coefficient of variation cv, and the initial bare ground fraction A_o. During the snow melt season, these parameters are kept constant, and the SDC gives the fractional bare ground $P(x)$as a function of the accumulated melt depth λ, which is assumed constant over the grid cell. During this period, the mass balance components Q and SWE are also functions of λ, and divide the initial snow pack m into accumulated snowmelt runoff and remaining snow storage, respectively.

The curve shown in Figure 2.1 is given a 3-parameter model: 2 of them refer quantitatively to the snowpack by a Gamma distribution and the third one quantifies the maximum SCA in the cell during the snowmelt process.

The model for a single cell is described by the following equations:

$$A(t) = A_0 \cdot \{1 - F[\lambda(t)]\}$$

$$F[\lambda(t)] = \int_0^{\lambda(t)} p(x; m, cv) dx = \gamma\left(\frac{1}{cv^2}, \frac{1}{cv^2} \cdot \frac{1}{m}\right)$$

Where,

p	probability density function (PDF).
$F(\,)$	cumulative probability distribution function. The value of F equals the Incomplete Gamma Function, γ, with shape and scale arguments.
$A(t)$	Snow Covered Area of the cell at a time t.
m	average Snow Water Equivalent.
cv	coefficient of variation at the beginning of the melt season.
A_0	Snow Covered Area at the beginning of the melt season.
$\lambda(t)$	accumulated melt depth since the beginning of the melt season.

2.2. HBV model

2.2.1. General overview of the model

HBV is an acronym formed from *Hydrologiske Byrån avdeling för Vattenbalans* at SMHI, Sweden.

The HBV-model is a hydrological model widely used for making runoff/inflow forecasts to hydropower systems and for making streamflow records for hydrological analyses in Norway, Sweden, Finland and some other many countries in and outside Europe.

Among the classifications of the different existing hydrological models, the HBV-model can be classified as:

- Linear model. Most of the mathematical expressions used by the model are linear. There are few of them nonlinear though.
- Lumped model. The catchment is handled as a homogeneous one unit, so the parameters of the model apply to the whole surface area and there is no consideration of spatial distribution. However, the first routine of the model (*snow routine*) runs as a distributed model, or better said, as a semi-distributed model.
- Conceptual model. The HBV-model is a deterministic, conceptual model. The principal physical elements of the catchment are represented in the model (typically, internal storages), as well as the main hydrological processes and their interrelations in a simplified form. Also, the internal states of the model are represented in the model (same input may yield different output depending on initial states). The conceptual models can be seen as grey boxes and are specially needed when amount, timing and variation pattern are important.
- Calibration is needed.

The HBV-model can be used for many different tasks. Some of them are listed here:

- Simulate catchment runoff, snow, soil moisture, etc.
- Extend or fill gaps in incomplete runoff series.
- Create “representative” runoff series in ungauged catchments.
- Make short-time (0-10 days) forecasts for future runoff/floods.
- Make long-time (1 week to 12 months) predictions for runoff, snow, soil moisture, etc.
- Produce statistics like Q_{normal}, Q_{max}, Q_{min}, $Q_{p75\%}$, $Q_{p25\%}$,… for flow, snow, soil moisture, etc. based on the current wetness situation of the catchment.

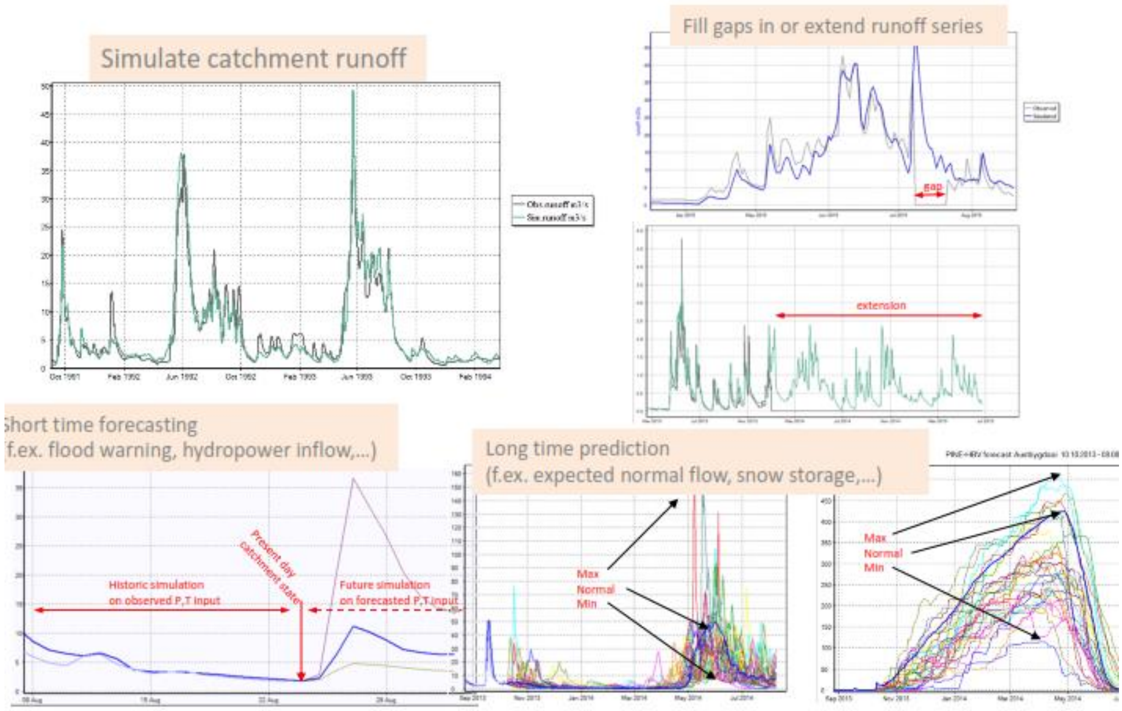


Figure 2.2. Examples of tasks in HBV.

2.2.2. Main components in the HBV model

The main structure of the HBV-model is a sequence of submodels [2]. Although this study is focused only on the three snow stacks in SHyFT, a brief introduction to all of the routines in the HBV-model is here presented:

- *Snow routine*: the model computes snow accumulation and melt in the catchment based on precipitation and air temperature input data. The main processes in this routine are Type of precipitation, Snow accumulation and Snowmelt.
- *Soil moisture* routine: the model computes storage of water in upper soil, evaporation from soil and vegetation (evapotranspiration) and runoff generation.
- *Upper zone*: the model computes the storage in surface water and in the active part of ground water. It transforms the runoff generation to a runoff hydrograph by accounting for transport delay and attenuation. Upper zone computes quick runoff.
- *Lower zone*: the model computes storage in deep groundwater and lakes, along with runoff delay and attenuation. Lower zone computes slow runoff ("Base flow") from groundwater reservoir and lakes. This flow will continue a long time after rainfall and/or snowmelt has stopped.

The next figure sketches the components of the HBV-model introduced above:

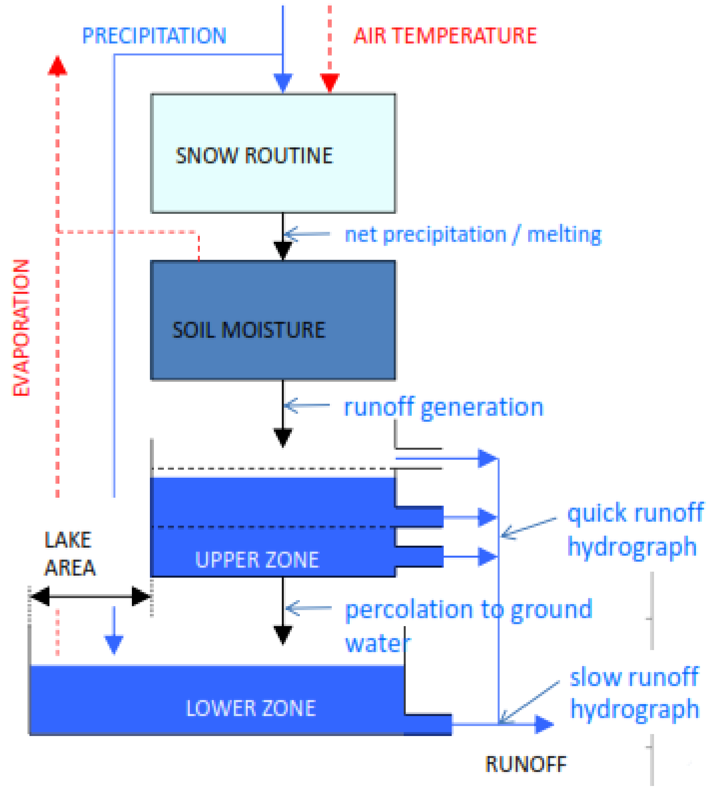


Figure 2.3. HBV model.

2.2.3. Meteorological input data in HBV

One well worth mentioning aspect of the HBV model is the correction of the meteorological input data.

The first input data that the model requires is Area precipitation and Area temperature, meaning the area of the catchment. However, in most gauged catchments, it is only available point observation data, both of precipitation and temperature. Therefore, these two input parameters must be adjusted and calculated from the point observed data. In order to do that, the following equations are used:

$$P_{area} = P_{obs} \times PCORR \times SCORR \times \left(1 + PGRAD \times \frac{H_{area} - H_{obs}}{100}\right)$$

$$T_{area} = T_{obs} \times TCGRAD \times \left(\frac{H_{area} - H_{obs}}{100}\right), \qquad days\ without\ precipitation$$

$$T_{area} = T_{obs} \times TPGRAD \times \left(\frac{H_{area} - H_{obs}}{100}\right), \qquad days\ with\ precipitation$$

Where,

$PCORR, SCORR$: precipitation correction factors for rain, snow.
$PGRAD$: precipitation increase coefficient with elevation [%/100 meter].
$TCGRAD$: temperature rate with elevation on clear days [ºC/100 meter].
$TPGRAD$: temperature lapse rate with elevation on cloudy days [ºC/100 meter].

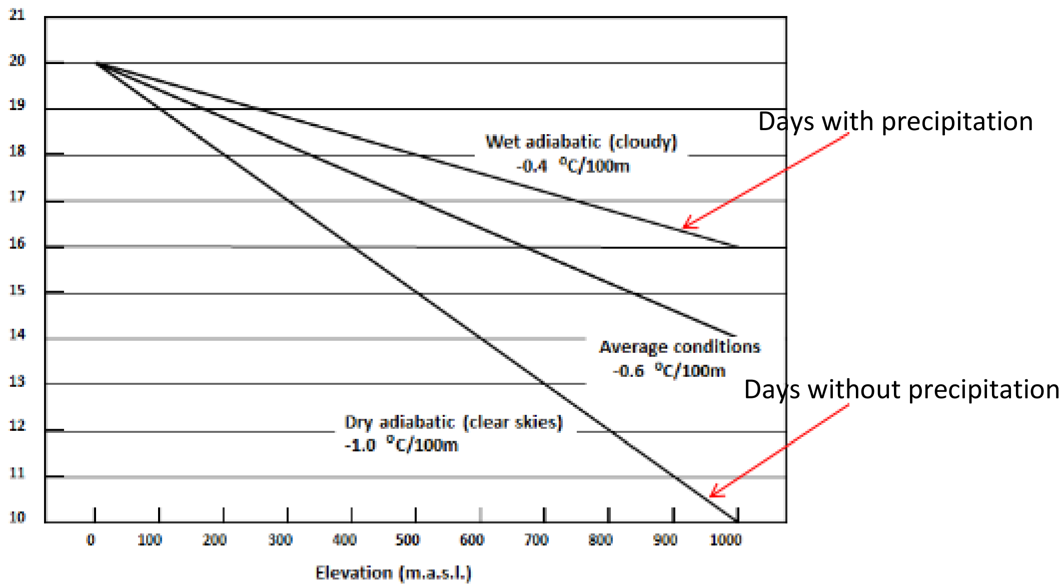


Figure 2.4. Air temperature lapse rate for three different meteorological conditions.

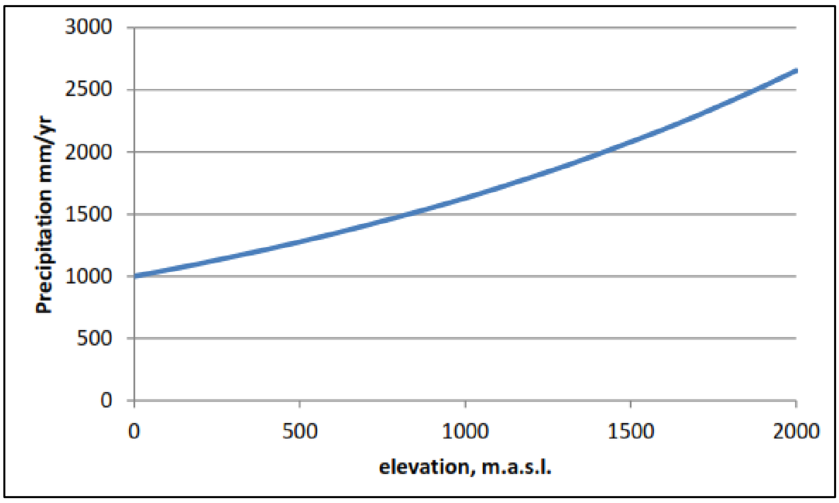


Figure 2.5. Precipitation gradient.

Experience says that precipitation increases some 3% - 8% every 100 meter, with an average of some 5% per 100 meter. However, there are variations between different climate regions. Compared to temperature, the precipitation gradient is more difficult to define by observation of different gauging stations. This is because precipitation has much greater temporal and spatial variation than temperature, and also because precipitation observations are more likely to present measurement errors than temperature, such as catch loss due to the wind, rain/snow evaporation loss, etc.

2.2.4. The snow routine of the HBV model

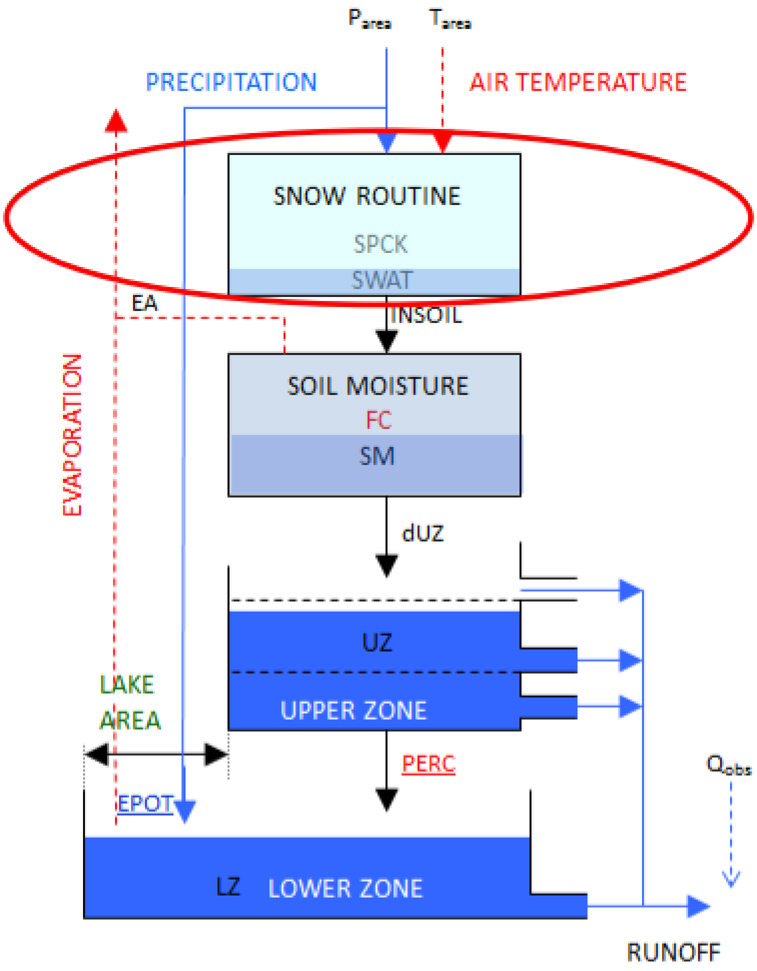


Figure 2.6. The snow routine in the HBV model.

The two main elements of the Snow routine are the dry snow and the free water in the snowpack. Regarding the processes in this routine, the main ones are:

- Snow accumulation
- Snow redistribution
- Snow melt
- Water accumulation in snow
- Water refreezing in snow
- Runoff from saturated snow.

The input data is the precipitation and the air temperature.

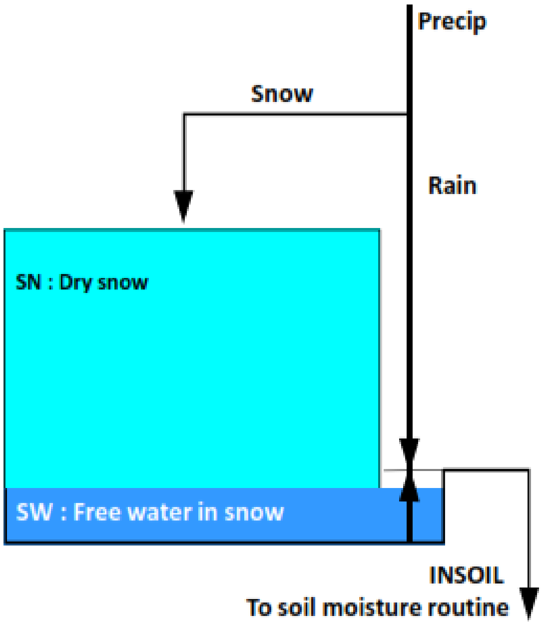


Figure 2.7. Main elements of the snow routine.

Approaches to compute snowmelt

One approach in order to calculate the snowmelt is by means of the snow energy balance. The general equation of the energy balance for snow surface can be written as:

$$F_e = K + L + H + LE + R + G$$

Where,

- **F_e** : **Net energy flux entering the snowpack** (change in storage).
- **K** : **Net short wave radiation**.
  Knet = Kin – Kout = Kin * (1 – albedo)
- **L : Net long wave radiation**.
  - Lnet = Lin – Lout
  - Lout = ε*σ*$Tsurf^4$, where ε is close to 1 and σ is the Stefan Boltzmann constant.
  - Lin = ε*σ*$Tair^4$, where ε is the atmospheric emissivity.
- **H : Sensible Heat**.
  - H = constant * Va* (Ta-Ts) , where Va is the wind velocity, Ta is the air temperature and Ts is the snow temperature.
- **LE : Latent Heat.**
  - LE = constant * Va * (ρa – ρs) , where ro is the water vapour pressure.
- **R : Heat from the rain.**
- **G : Heat from the ground.**

The next figure shows the fluxes of energy that take part in the energy balance:

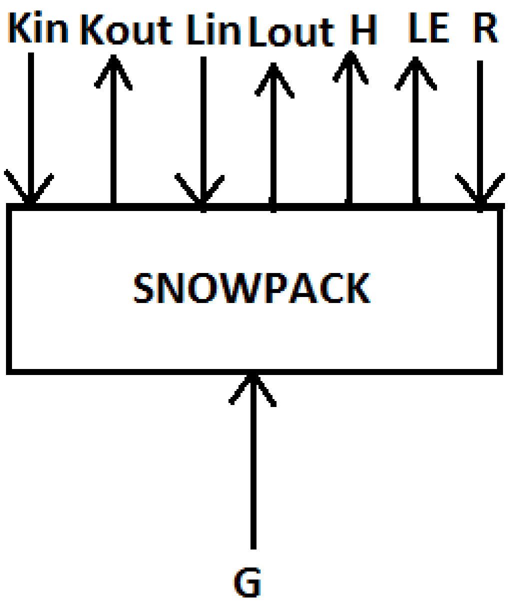


Figure 2.8. Sketch of the snow melt energy balance.

In the sketch, all fluxes are positive when directed into the snowpack. The sensible heat and the latent heat present typically negative values and that is why they are plotted exiting the snowpack. Also, the heat from the ground and the heat from the rain are often neglected from the balance.

Once the net energy flux entering the snowpack (F_e) is known, the daily rate of snowmelt in millimetres per day can be obtained as follows:

$$M\left[\frac{mm}{day}\right] = 1000 * \frac{F_e\left[{}^{kJ}\!/_{(m^2 * day)}\right]}{h_f\left[{}^{kJ}\!/_{kg}\right] * \rho_w\left[{}^{kg}\!/_{m^3}\right] * B[-]}$$

Where,

hf: latent heat of fusion.
ρw: density of water.
B: thermal quality of the snowpack i.e. fraction of ice in a unit mass of snow.

However, the practical difficulty of this physical approach relies on its large input data requirements: precipitation, air temperature, air humidity, short and long wave radiation, albedo, cloud cover, wind, thermal quality of the snowpack… This approach is therefore rejected. So what is the HBV-model approach in the snow routine?

Snowmelt has a strong correlation to air temperature:

$$SNWMLT = CX * (Ta - Ts)$$

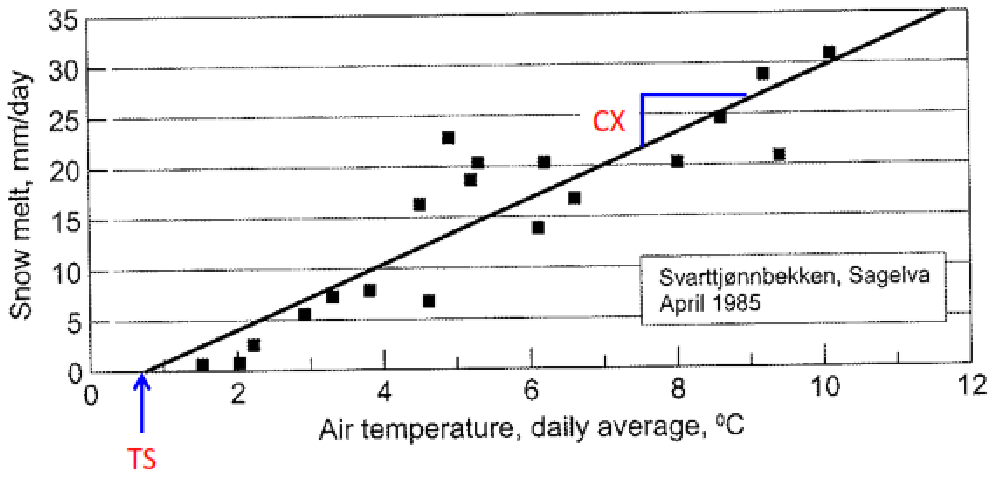


Figure 2.9. HBV model air temperature approach.

This is the approach used by the HBV-model.

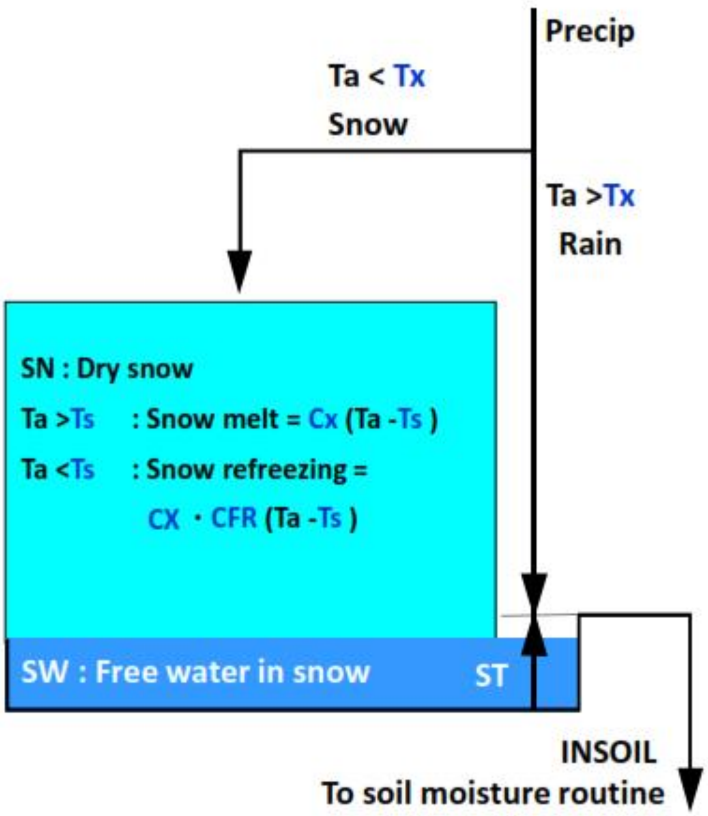


Figure 2.10. Physical processes in the HBV snow routine.

Where,

Ta: Air temperature (average for the period).
Tx: Threshold temperature between rain and snow.
Ts: Threshold temperature between snowmelt and snow refreezing.
CX: Degree-day factor, [mm/(ºC . day)].
CFR: Degree-day factor for refreezing.

ST: Maximum capacity of snow water = SN * CPRO
CPRO: Maximum ratio free water in the snowpack.

The amount of water that will flow to the next routine (the soil moisture routine), INSOIL, will be positive when SW exceeds ST. The water balance in the routine can be then expressed as:

$$\Delta(SN + SW) = Precip - INSOIL$$

However, there are still some shortcomings in this process: the correlation between snowmelt and air temperature is not identical in all catchments. Therefore, the Degree-day factor, CX, and the Threshold temperature, Ts, must be calibrated.

The snow routine as a semi-distributed hydrological model

There are some factors that make it convenient and necessary for the HBV-model snow routine to work as a distributed (semi-distributed) hydrological model. These are the 3 different distributions that are considered in the routine:

a. Elevation distribution:

It is easy to understand that many meteorological and hydrological parameters such as snow accumulation, precipitation or air temperature vary significantly with elevation. Indeed, more snow accumulates at higher elevation due to higher precipitation and lower air temperature. Since air temperature is influenced by elevation, precipitation type (rain/snow) is therefore also affected by elevation, as well as the snow melt rate.

Consequently, the catchment must be divided into a number of zones (typically 10 zones) from the lowest to highest levels, and compute snow accumulation and snowmelt separately in each zone.

In order to do that, the Area-Elevation or Hypsographic curve is constructed for the catchment and pairs of Precipitation-Temperature are assigned for each interval zone:

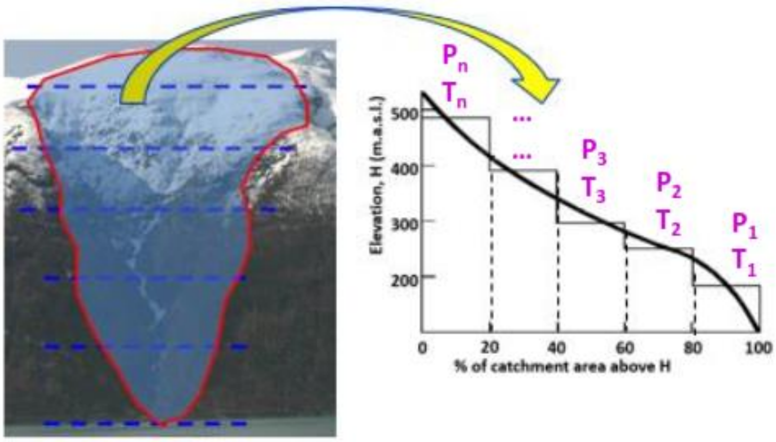


Figure 2.11. Distribution 1: elevation zones.

b. <u>Forested areas and open areas:</u>

Another issue to take into account is whether the catchment is a forested area or a bare area. The following factors are directly affected by this matter:

- Snow interception and snowpack redistribution is different for forested parts compared to bare areas.
- Net incoming radiation to snow surface is different for forested parts compared to bare areas.
- Wind over snow surface is different for forested parts compared to bare areas (turbulent energy transfer as well).

Hence, having separate melt factor, melt threshold and snow distributions in forested parts compared to forest free areas is important and necessary for a successful output in the model.

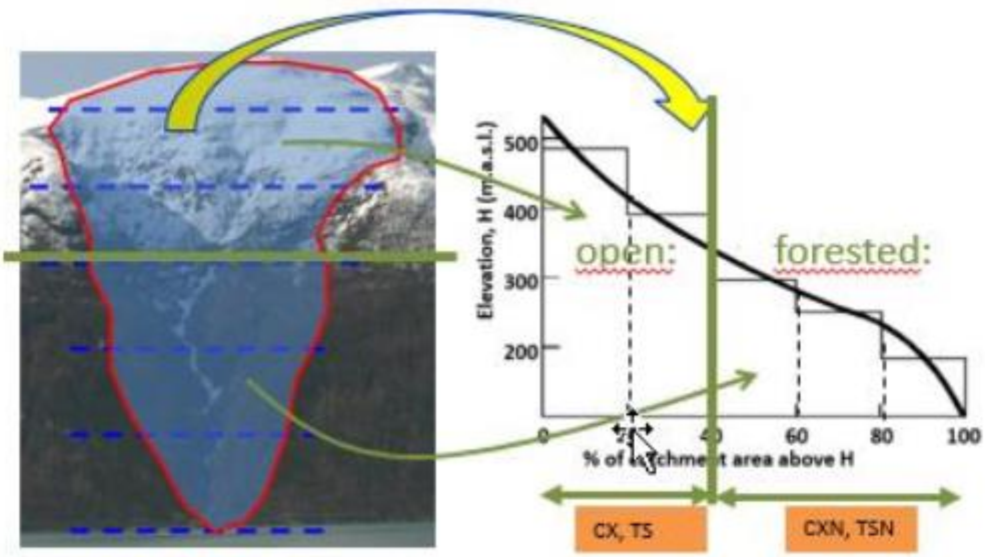


Figure 2.12. Distribution 2: forested and open areas.

- Forested zone: $SNWMLT_N = CX_N * (Ta - TS_N)$
- Forest free zones: $SNWMLT = CX * (Ta - TS)$

c. <u>Statistical distribution within each zone:</u>

This distribution is made because of the effect of wind on snow distribution across irregular terrain surface.

- The zone area is partitioned into 5 different snow depth blocks.
- Individual snow accumulation and snowmelt is calculated for each block.

2.3. Skaugen snow model

The Skaugen snow model, by Thomas Skaugen and Onof in 2014 and later Skaugen and Mengistu in 2015, is a rainfall-runoff model written in the programming language R and runs operationally at daily and 3-hourly time steps at the Norwegian flood forecasting service at the Norwegian Water Resources and Energy Directorate (NVE) [3].

The Skaugen snow model, also called Distance Distribution Dynamics model (DDD), was developed with the objective of reducing as much as possible the need of calibration against runoff. A reduced number of parameters to be calibrated, while maintaining the accuracy and detail required by modern hydrological models, will reduce parameter and model structure uncertainty and improve model diagnostics [4]. The model has a majority of its parameters estimated directly from observed data such as maps and runoff characteristics. Input to the DDD model is precipitation and temperature.

Regarding the name DDD, the model drives the dynamics of runoff from the distribution of distances from points in the catchments to the nearest stream. This distribution is unique for each catchment and is obtained from a geographical information system (GIS).

The model is semi-distributed:

- Rainfall, snowmelt and snow accumulation are performed for 10 different elevation zones of equal area.
- Catchment average precipitation and temperature are distributed to the elevation zones using calibrated lapse rates. Average precipitation is corrected by multiplying with a constant in order to get a correct long-term water balance.

Snowmelt is estimated using a degree-day model: the generated melted discharge is a linear function of the difference between air temperature and a calibrated threshold melting temperature.

The current routine in DDD for spatial Probability Density Function (PDF) of Snow Water Equivalent (SWE) is the Snow Distribution Log-Normal. This routine, SD_LN, distributes SWE log normally in space with a fixed and calibrated coefficient of variation (CV). In this routine, the PDF is modelled as the sum of uniform and log-normally distributed snowfall events [5]. The distribution is constant up to a certain specified threshold of accumulated SWE.

Every other snowfall event is log-normally distributed through a calibrated CV, θ_{CV}, and SWE is estimated for nine quantiles and added to previous quantile values. This approach gives every other snowfall event a spatial distribution of a fixed shape (through the calibrated CV, θ_{CV}), without taking into account the intensity of the event.

Furthermore, this method assures a perfect spatial correlation: a new snowfall event is distributed in such a way that the quantile with the highest SWE always gets the most SWE so that the coefficient of correlation of the sum of the events remains constant.

Let the next simple example show this:

Let Z be the accumulation of snow water equivalent of the sum of two different snowfall events, $Z = y_1 + y_2$, where y is log-normally distributed with mean μ_y and variance σ_y^2.

This way, the mean of Z is $E(Z) = 2\mu_y$ and the variance is $Var(Z) = \sigma_y^2 + \sigma_y^2 + COV(y_1, y_2)$. With perfect correlation the variance equals $Var(Z) = \sigma_y^2 + \sigma_y^2 + 2\sigma_y^2$ and it is easily observed that the coefficient of correlation, CV, for Z equals that of y:

$$CV_Z = \frac{\sigma_Z}{\mu_Z} = \frac{2\sigma_y}{2\mu_y} = CV_y$$

The spatial distribution of melt is constant and reduction in Snow Covered Area occurs when the Snow Water Equivalent associated with a quantile becomes 0 [3]. The fraction of snow-free areas is thus the sum of quantiles with zero SWE.

Among the relevant model parameters for snow accumulation and snowmelt, which are estimated by calibration against runoff, the following ones can be listed:

- θ_{CV} : describes the spatial distribution of SWE.
- θ_{CX} : degree-day factor.
- θ_{W_s} : maximum liquid water content in the snowpack.

Some of the parameters of the DDD model are given values obtained through experience in calibrating DDD for gauged catchments in Norway. Some others, however, are assigned standard values as suggested in different literature.

3. STUDY AREA

This section will focus on the location, climate conditions and hydro-meteorological characteristics of the catchment for this study.

3.1. Background

Norway is a Nordic country laying between 57° and 81° N in latitude and 4° and 32° E in longitude. It represents the western part of Scandinavia and it shares border line with Sweden, Finland and Russia, as it can be seen in Figure 3.1.

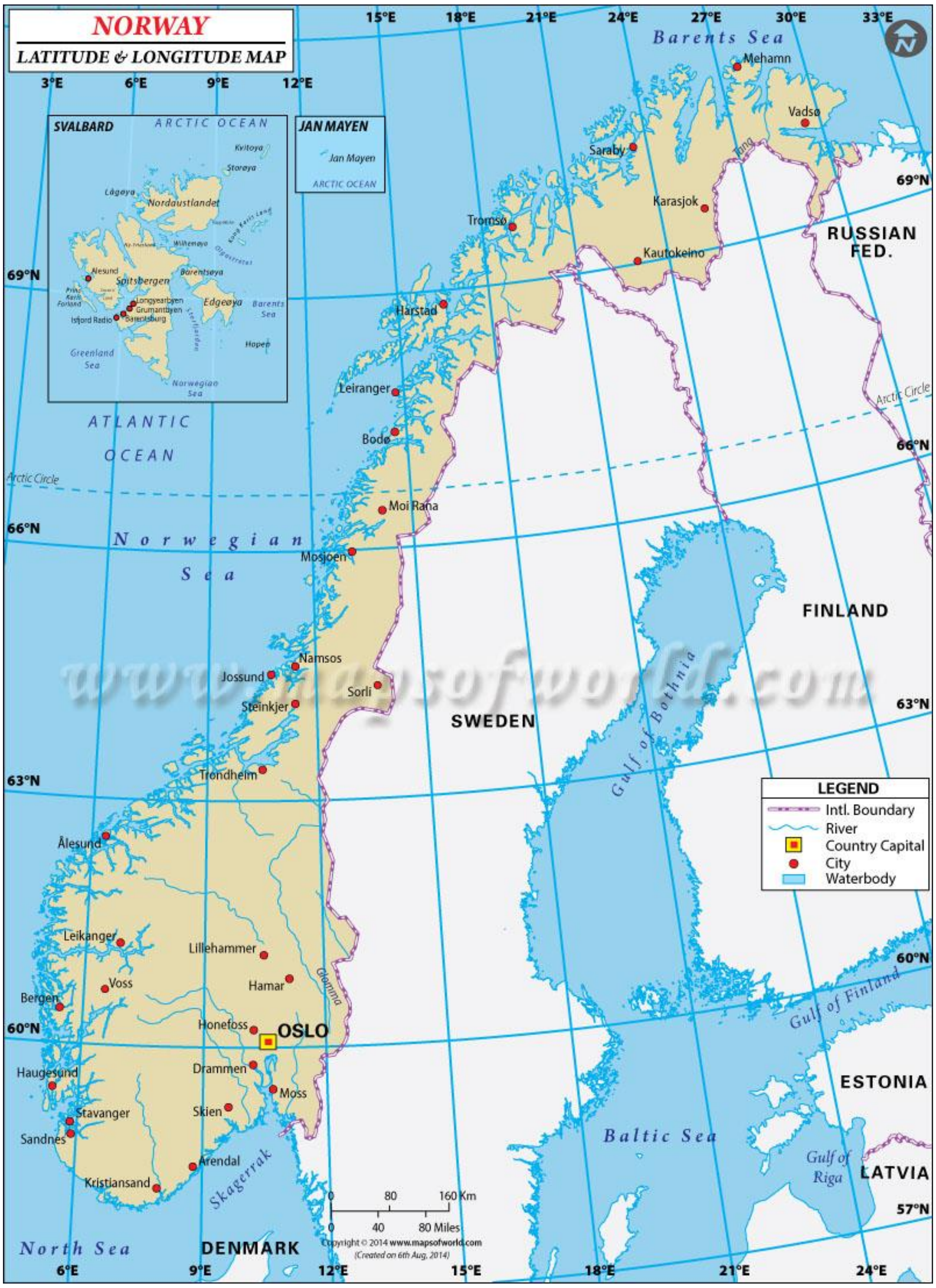


Figure 3.1. Map of Norway.

Norway has abundant rivers and lakes that have been organised into 16 river basin districts, out of which 10 are international sharing water courses with Sweden and Finland to the east, and the remaining 6 are solely Norwegian [6].

Figure 3.2. River basin districts in Norway.

Each river basin district is subdivided into smaller catchments as can be seen in Figure 3.3 [7]

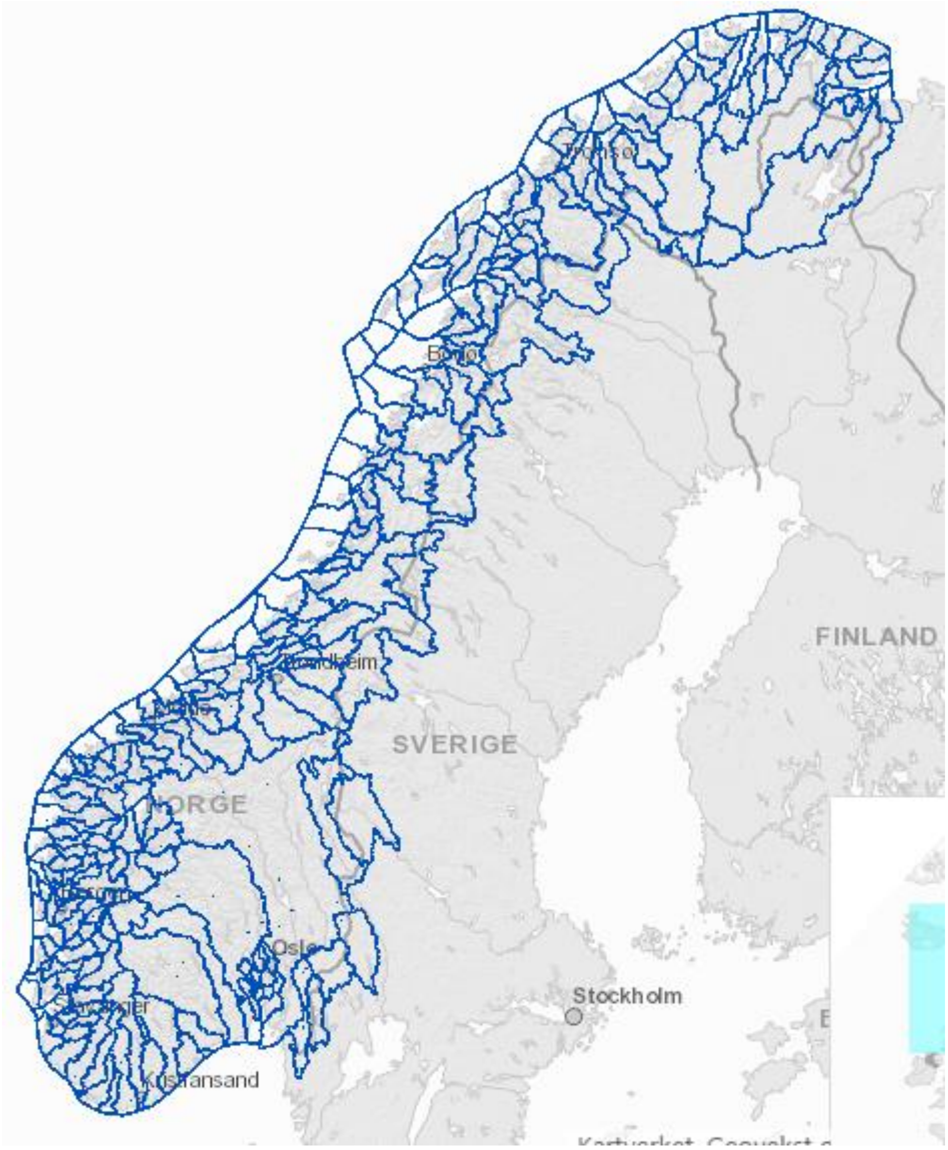


Figure 3.3. Catchments in Norway.

Among all the catchments in Norway, the area of study of this work is the Nea-Nidelva river basin.

3.2. Nea-Nidelva River Basin

3.2.1. Location and topography

Nea-Nidelva River Basin is defined in the central part of Norway, in Sør-Trøndelag, between latitude 63° and 64° N, and longitude 10° and 12° E.

The total area of the catchment is 3661 km2 and the lowest outlet lies in the fjord of Trondheim, end of the Nidelva river. Important rivers in the Nea-Nidelva river basin are the Nidelva, Nea, Rotla, Lødølja, and Tya. Amongst the biggest lakes are Sylsjön, Nesjøen, Stugusjøen, Finnkoisjøen, and Selbusjøen.

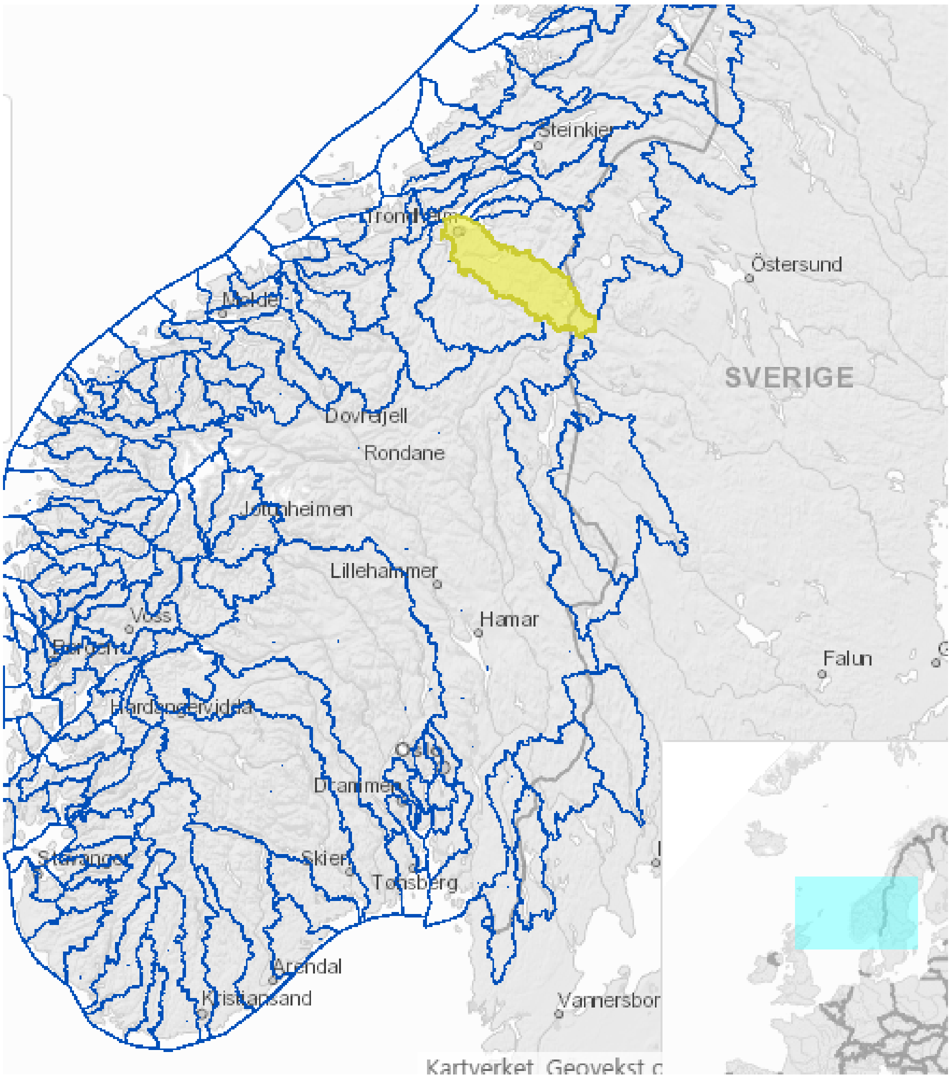


Figure 3.4. Nea-Nidelva River Basin in Norway [7].

The elevation difference of the catchment varies between 1 masl and 1789 masl. The next figures show the elevation distribution throughout the catchment and the corresponding hypsographic curve [8].

H_{min}	1 moh.
H_{10}	158 moh.
H_{20}	285 moh.
H_{30}	397 moh.
H_{40}	496 moh.
H_{50}	592 moh.
H_{60}	700 moh.
H_{70}	765 moh.
H_{80}	853 moh.
H_{90}	976 moh.
H_{max}	1789 moh.

Figure 3.5. Elevation distribution of Nea-Nidelva.

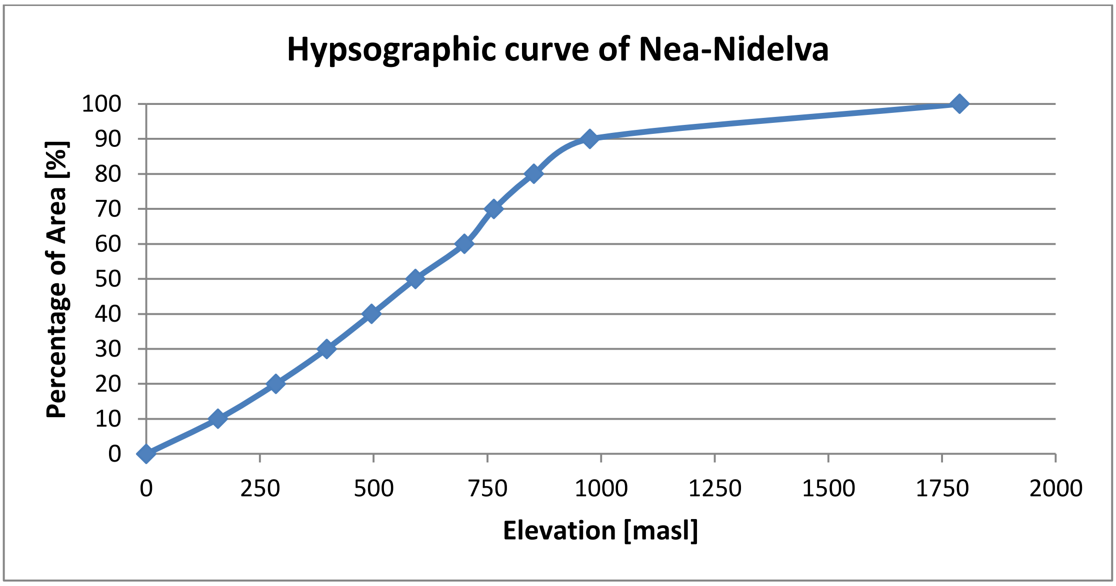


Figure 3.6. Hypsographic curve of Nea-Nidelva.

From the previous hypsographic curve, it can be said that about 90 percent of the total area of the catchment falls below the height of 976 masl. For the lower region, the elevation distribution is quiet uniform: the area under a certain height increases a 10 percent almost every 100 m.

Note: *since the snow data for the comparison later in this work was available for the south-east side of the catchment, the following subsections regarding meteorological description will present time series diagrams from gauging stations within that region.*

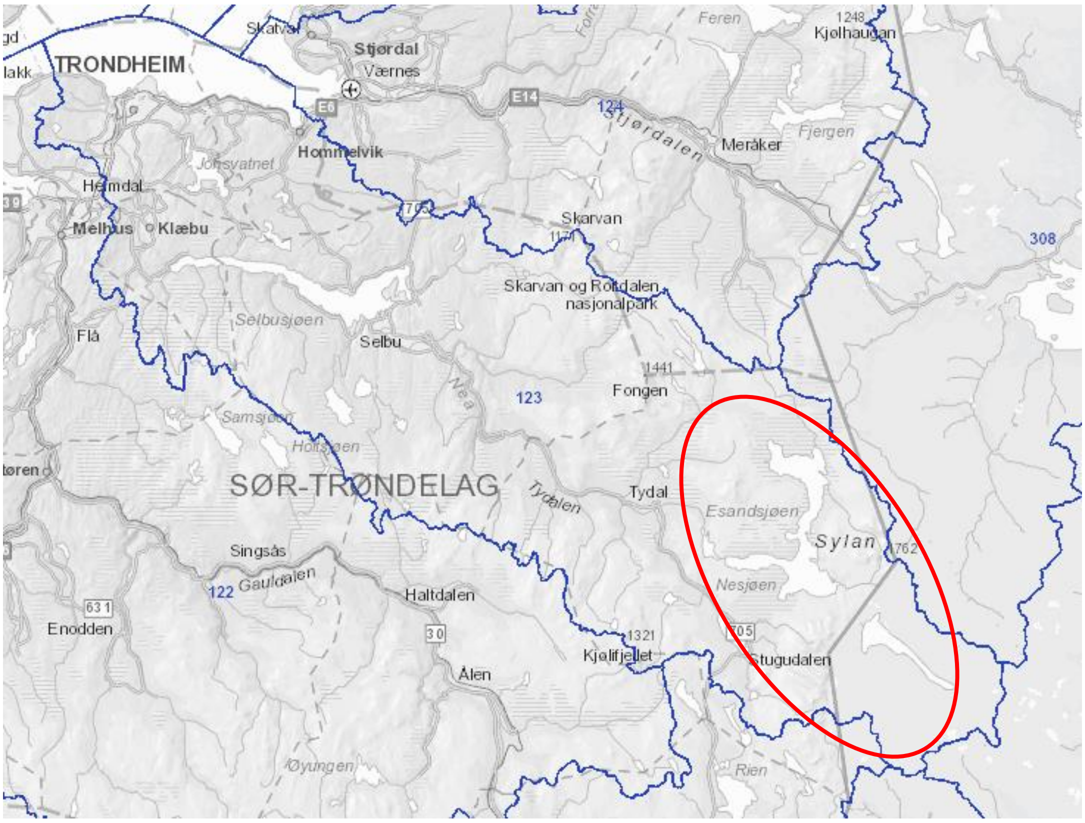


Figure 3.7. Region of interest in the catchment.

3.2.2. Precipitation

The normal annual precipitation of the catchment for the period between 1971 and 2000 can be illustrated as follows [9].

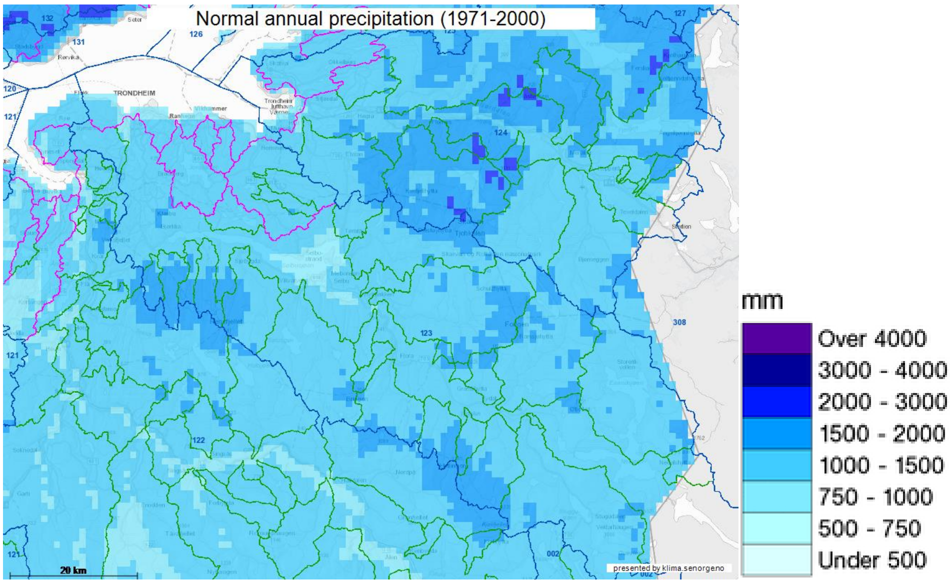


Figure 3.8. Normal annual precipitation (1971-2000).

It can be seen that the normal annual precipitation for the different points in the catchment takes values between 750 and 2000 mm.

Regarding the region of interest in the study, the normal annual precipitation for the same period takes a value of 909.5 mm and the variation in time can be plotted as [9]:

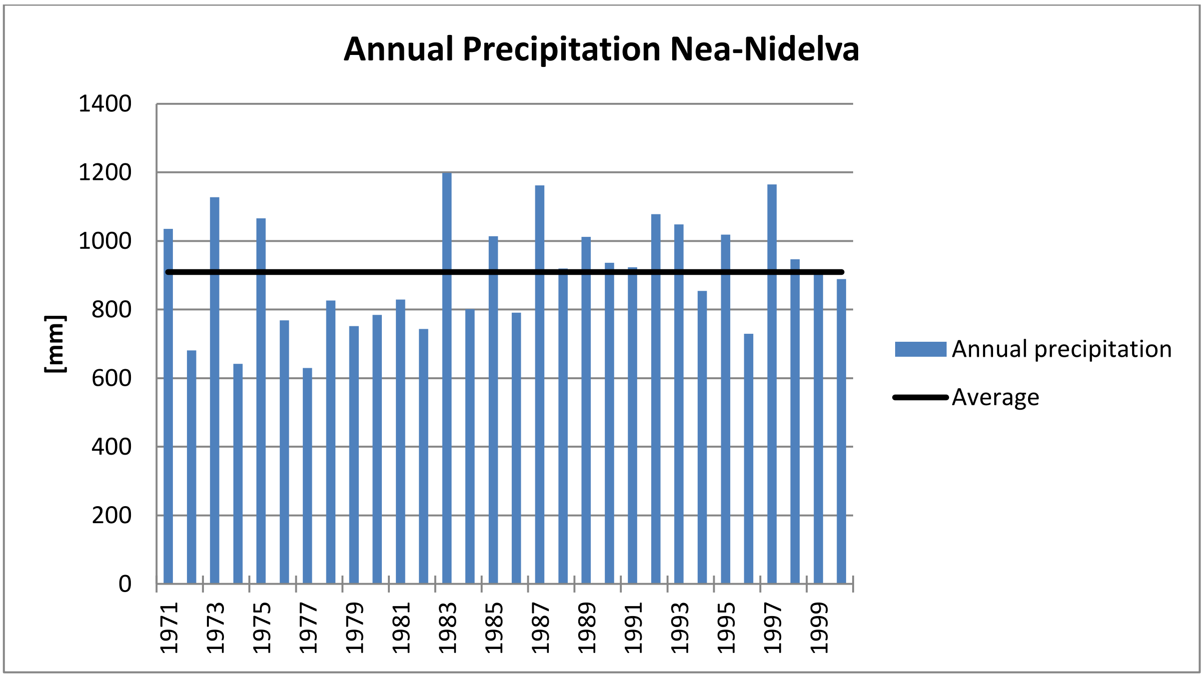


Figure 3.9. Annual Precipitation Nea-Nidelva.

3.2.3. Temperature

The normal annual temperature of the catchment for the period between 1971 and 2000 can be illustrated as follows [9].

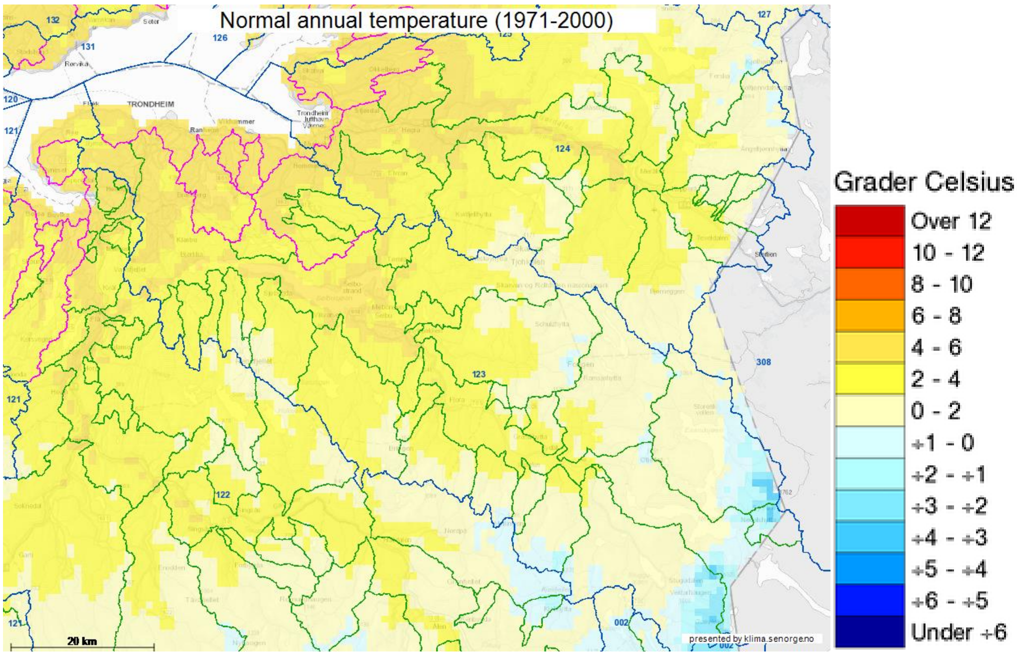


Figure 3.10. Normal annual temperature (1971-2000).

It can be seen that the normal annual temperature for the different points in the catchment in that period takes values approximately from -3ºC to 8ºC.

Regarding the region of interest in the study, the normal annual temperature for the same period takes a value of -1.4ºC and the variation in time can be plotted as [9]:

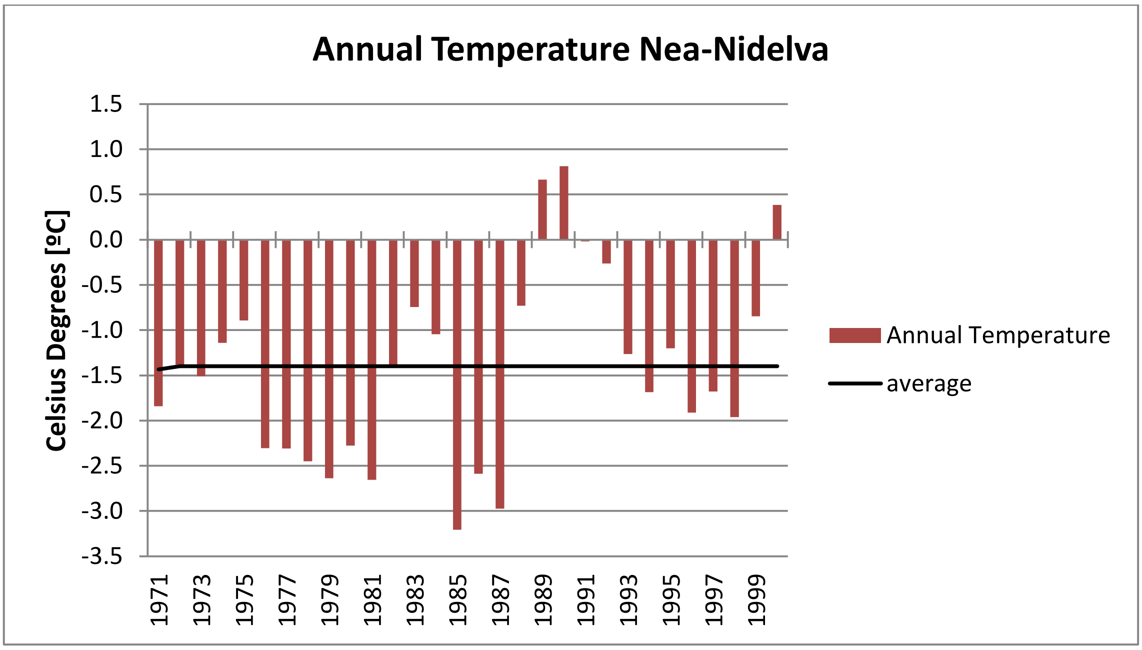


Figure 3.11. Annual Temperature Nea-Nidelva.

3.2.4. Snow

The Snow Water Equivalent (SWE) represents the amount of liquid water that is contained in a certain volume of snow. It depends on the density of the snow in each case, which depends on the age or the type of the snow in consideration. It can be understood as the depth of liquid water that would theoretically result if the whole snow pack instantaneously melted [10].

It can be simply calculated as:

$$SWE\ [mm] = snow\ depth\ [mm] * \frac{snow\ density\ [kg/m^3]}{water\ density\ [kg/m^3]}$$

However, one can distinguish between SWE and Snow Depth. In fact, many institutions such as the Norwegian Water Resources and Energy Directorate analyse and map both of them separately.

The normal annual maximum of snow amount in mm of water equivalent of the catchment for the period between 1971 and 2000 can be illustrated as follows [9]:

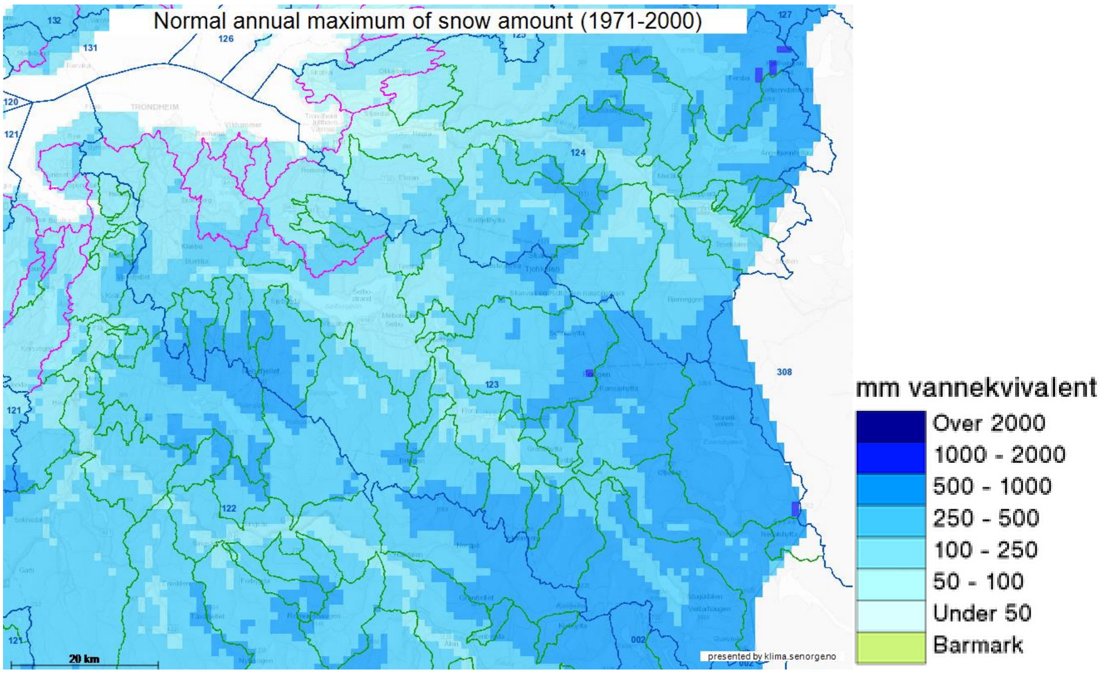


Figure 3.12. Normal annual maximum of snow amount (1971-2000).

It can be seen that the normal annual maximum of snow amount for the different points in the catchment in that period takes values approximately from 50 to 1000 mm of water equivalent.

Regarding the region of interest in the study, the normal annual maximum of snow amount for the same period takes a value of 459.81 mm of water equivalent and the variation in time can be plotted as [9]:

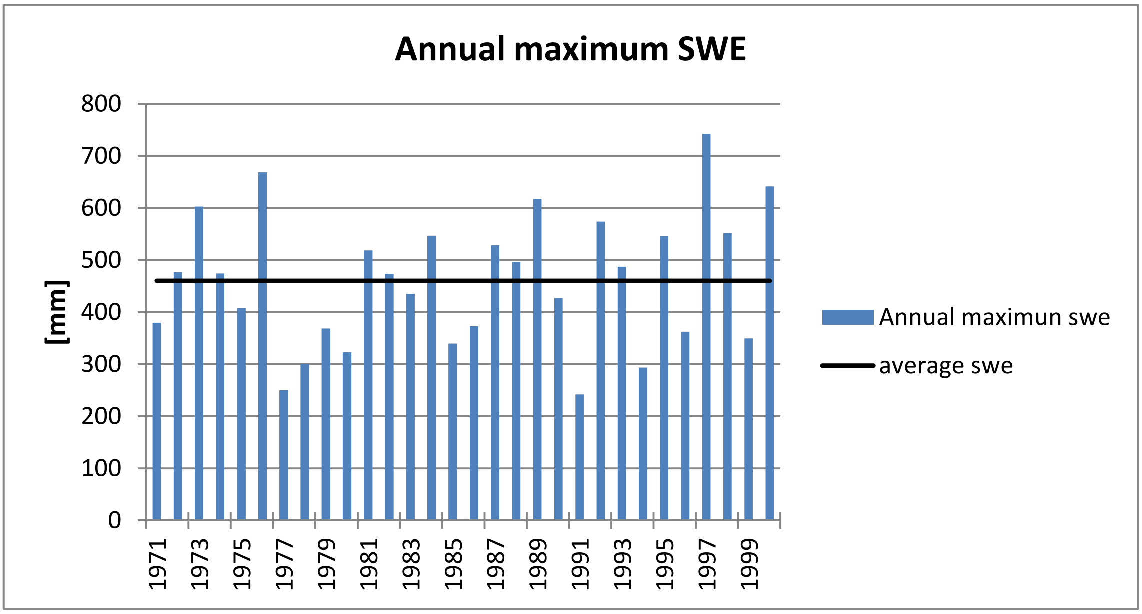


Figure 3.13. Annual maximum of snow amount in mm of water equivalent.

Regarding the snow depth in cm of snow cover or snowpack, the normal annual maximum of snow depth in cm of the catchment for the period between 1971 and 2000 can be illustrated as follows [9]:

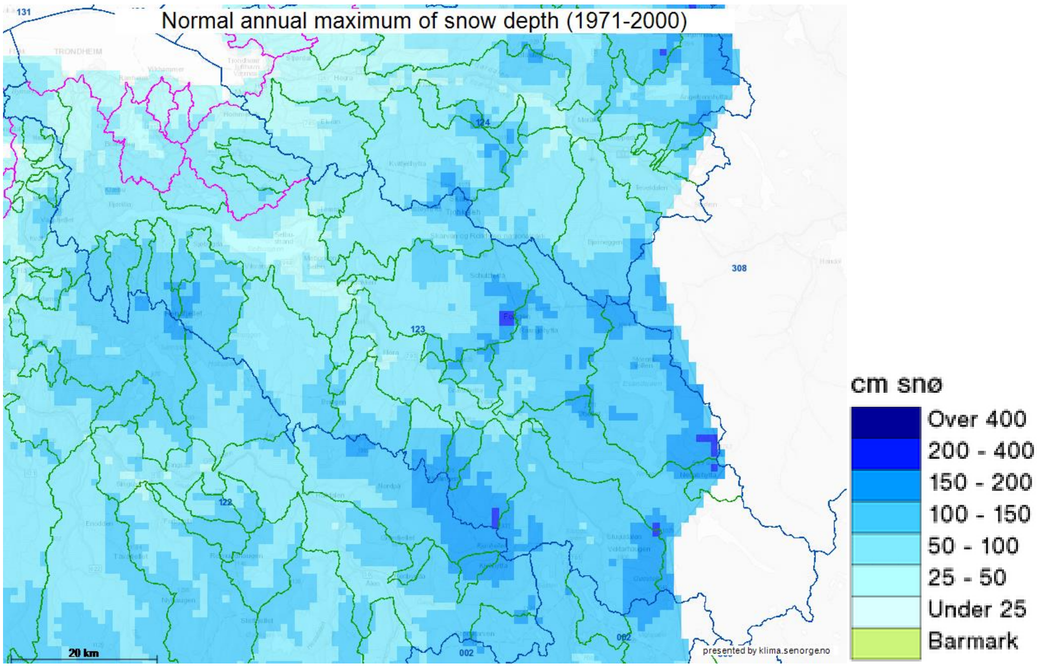


Figure 3.14. Normal annual maximum of snow depth (1971-2000).

It can be seen that the normal annual maximum of snow depth for the different points in the catchment in that period takes values approximately from under 25 to 400 cm of snowpack.

Regarding the region of interest in the study, the normal annual maximum of snow depth for the same period takes a value of 138.4 cm of snowpack and the variation in time can be plotted as [9]:

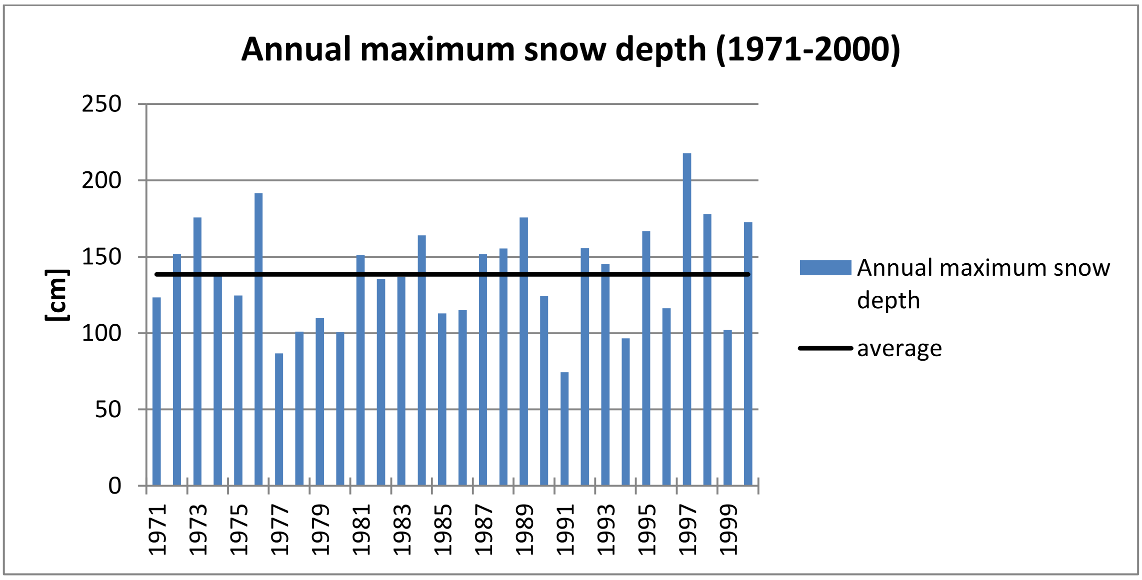


Figure 3.15. Annual maximum of snow depth in cm of snowpack.

Now that both snow water equivalent and snow depth have been presented, it is interesting to see the type of snow for the same period, which relates to the snow density. The Norwegian Water Resources and Energy Directorate also provides data about number of days with dry snow per year.

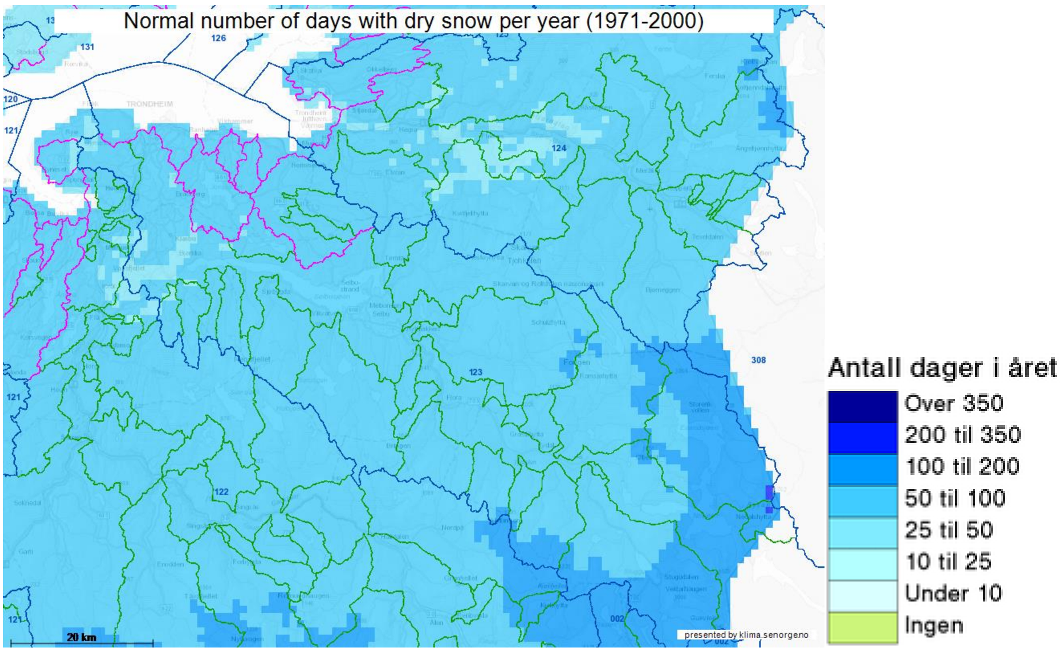


Figure 3.16. Normal number of days with dry snow per year (1971-2000).

It can be said that the values along the catchment vary from 10 to around 200 days of dry snow per year. For the specific point of the study, that value lies around 50-100 days per year.

3.2.5. Land cover

NEVINA is an interactive map tool which allows the user to calculate catchments, field parameters, climatic parameters and flow indexes for a freely chosen point along the Norwegian river system network [8].

Also, when defining a catchment, NEVINA calculates different parameters and allows the user to download a report as the outcome. The next figures show some sections of this report for the Nea-Nidelva catchment.

Figure 3.17. Nea-Nidelva catchment surface with NEVINA.

Amongst the different data that this map tool provides, it shows the different land cover as percentages of the total area:

Type of land cover	Percentage of total area
Cropland	2.6 %
Swamp	11.7 %
Sea	6.8 %
Forest	36.4 %
Bare mountain	29.8 %
Urban	1.2 %
Others	11.5 %

Table 3.1. Land cover distribution of Nea-Nidelva.

3.2.6. Gauging stations

The Nea-Nidelva River Basin has multiple hydro-meteorological gauging stations throughout its surface.

NVE Atlas is the main map tool of NVE on the Web. It contains most of the thematic geo-data from NVE. Examples: Catchments, lakes, rivers, annual runoff, embankments, areas prone to flood, hydro power plants, wind power plants, hydrological gauging stations and bathymetric maps for about 600 lakes [7].

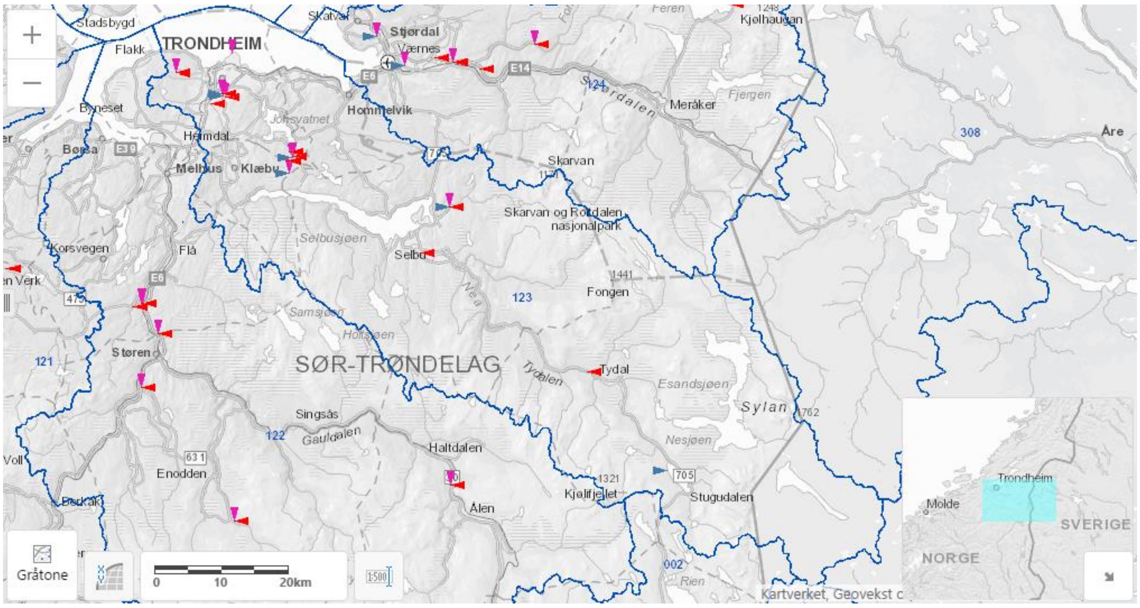


Figure 3.18. Hydro-meteorological stations in Nea-Nidelva.

Where,

Vannstand, vannføring sanntid	Water level, water supply
Snø sanntid	Snow
Meteorologiske data sanntid	Meteo

Table 3.2. Legend of types of gauging stations in Figure 1.18.

4. DATA AND WORK METHODOLOGY

This section will summarize the step-wise procedure carried out by the author of the thesis during the data preparation period.

Observed snow data

The purpose of this master thesis is to evaluate the SHyFT snow routines against observed snow data from satellite images and snow measurements in the field. This data was obtained by Statkraft and handed to the student in excel format.

The snow data was available from 2012 to 2016 and contained the measurements through the same 9 snow transects per year. The following information was given for each transect:

- ✓ Date of recording.
- ✓ Average depth [m].
- ✓ Signal velocity [m/ns].
- ✓ Average SWE [mm].
- ✓ Snow density Equation.
- ✓ UTM Zone, Easting and Northing coordinates of every measurement along the transect.
- ✓ Time, Altitude, Snow depth, SWE and Snow density of every measurement along the transect.
- ✓ Radar information.

Among all, one of the most important data was the geographical position of the measurements, i.e. the location of the snow transects, which in the case was given by Easting and Northing positioning in UTM Zone.

The reason why this information was very important is that being able to compare simulated results in SHyFT against real snow data at the same points makes the discussion much more reasonable. Moreover, comparing the available snow data against, for instance, the simulated results for the whole catchment would not give any trustable conclusions.

SHyFT is able to perform the simulations at different levels of action: one can run SHyFT and extract different variables such as Discharge, Snow Covered Area or Snow Water Equivalent for the whole catchment, for one particular subcatchment or even for one single cell of the grid if needed.

Therefore, once the location of the snow transects were known, one of the first steps was to find which cells represented the closest points to the real location of the available snow data in the catchment. In order to do that, the next steps were followed:

Note: *the following map with the location of the average point of each transect was made with Google Maps [11] and the coordinates were transformed from UTM to (lat, long) with the free online software Zonum Solutions [12].*

Note: *the maps with all the measurements per snow transect can be found in Appendix 1.*

i. First, the average Easting and Northing coordinates of the measurements per snow transect were calculated and converted into geographical latitude and longitude coordinates:

Transect	E (X)	N (Y)	UTM Zone	H [m]	Latitude, Longitude
1	339326.60	6989695,99	33V	743,53	63.01, 11.83
2	342280.30	6994911,28	32V	955,14	63.05, 11.88
3	345338.88	7005640,31	32V	874,17	63.15, 11.93
4	352837,21	6990820,77	33V	982,93	63.02, 12.09
5	362407,87	6979930,88	33V	1065,9	62.92, 12.29
6	365996,50	6970158,03	33V	987,62	62.84, 12.37
7	360123,68	6966759,96	33V	1215,08	62.80, 12.26
8	359276,13	6975010,07	33V	959,87	62.88, 12.23
9	348807,54	6981941,38	33V	999,69	62.94, 12.02

Table 4.1. Average coordinates of each snow transect

These coordinates represent the next points in the map:

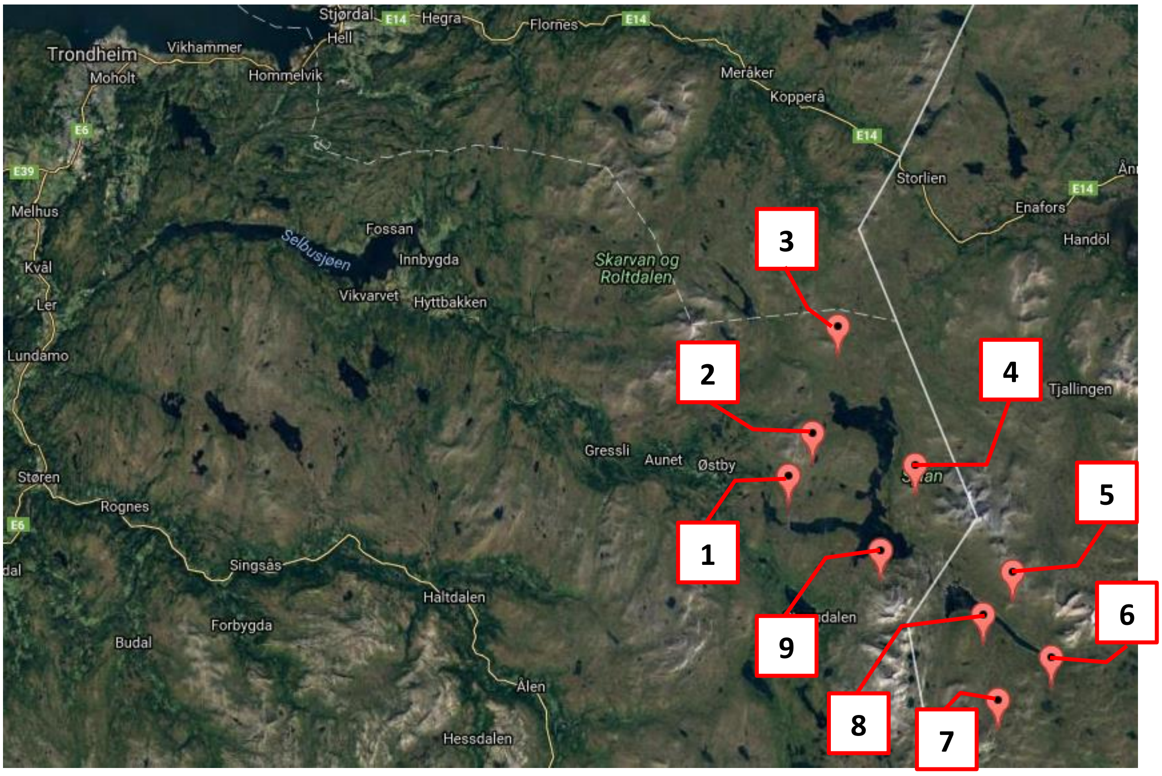


Figure 4.1. Map of the average points of each snow transect.

ii. In ShyFT, each cell of the grid has a geo-location (x, y, z), expressed in UTM Zone 33V coordinates. In order to find the closest cells to the average points of each snow-transect, the geo-location of every cell was printed as:

```
[…]
cells = simulator.region_model.get_cells()
x = np.array([cell.geo.mid_point().x for cell in cells])
y = np.array([cell.geo.mid_point().y for cell in cells])
   for i in range (0, 3661)
       print("X: ", x[i], "Y: ", y[i])
```

Figure 4.2. Code in SHyFT for printing the X and Y coordinates of the cells.

iii. The previous code showed the E (X) and N (Y) coordinate of every cell in the grid of the catchment. The result of plotting all those coordinates is the shape of the Nea-Nidelva catchment, as it should be.

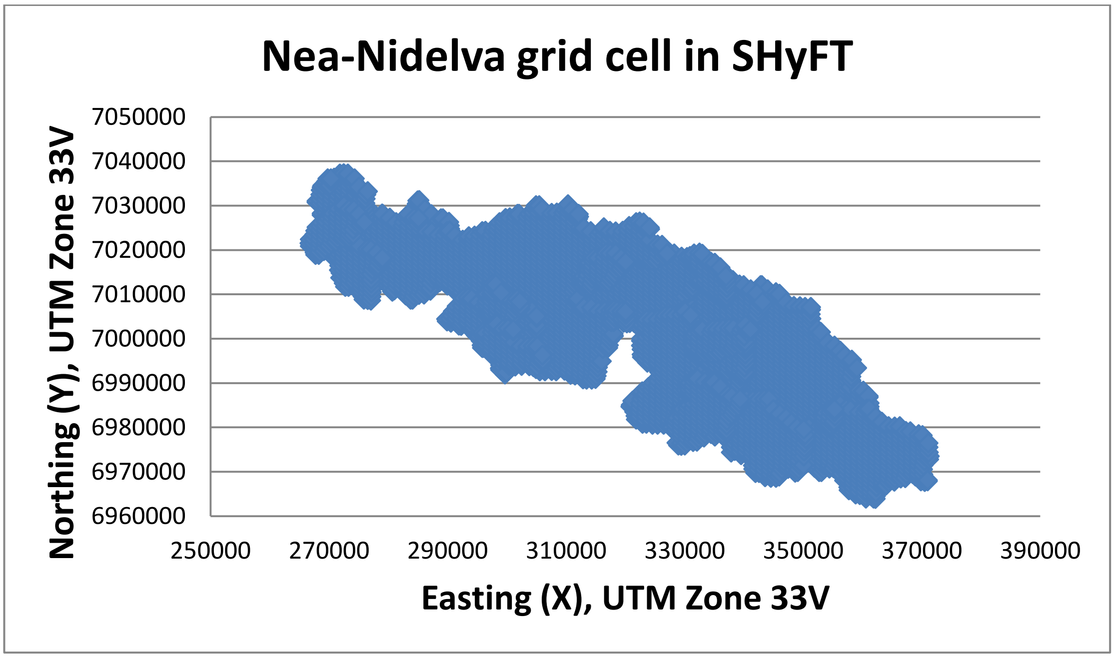


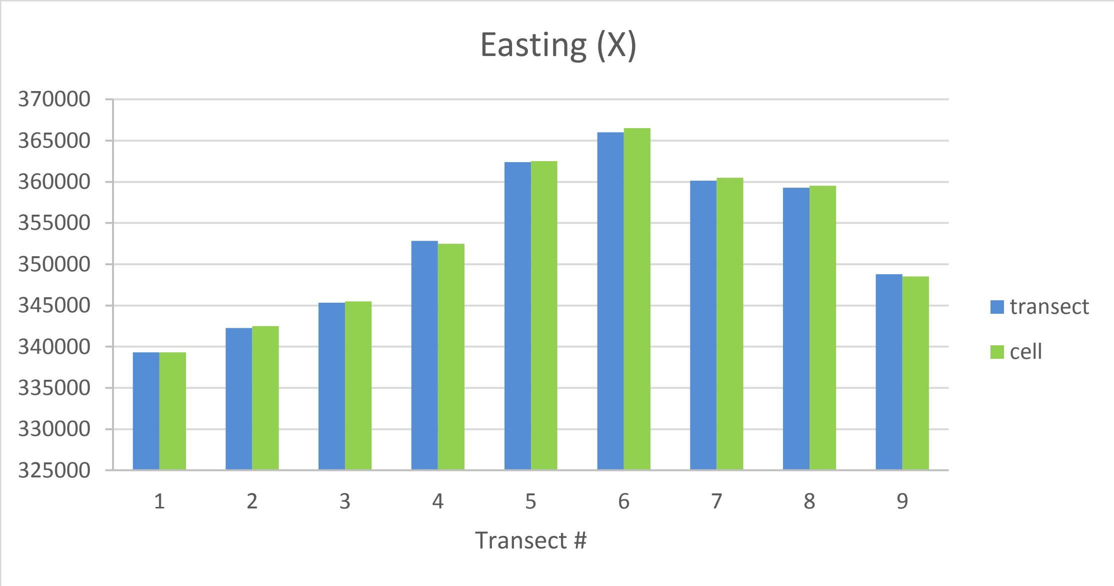
Figure 4.3. Plot of the 3662 cells contained in the grid of Nea-Nidelva catchment.

Once the coordinates of every cell were printed out, the most representative ones were chosen by simple comparison:

Transect	E (X)	N (Y)	E (X)	N (Y)	cell #
1	339326,6	6989695,99	339333	6990549	**1003**
2	342280,3	6994911,28	342500	6994500	**51**
3	345338,88	7005640,31	345500	6990500	**148**
4	352837,21	6990820,77	352500	6991500	**337**
5	362407,87	6979930,88	362500	6980500	**857**
6	365996,5	6970158,03	366500	6969500	**906**
7	360123,68	6966759,96	360500	6967500	**800**
8	359276,13	6975010,07	359500	6974500	**786**
9	348807,54	6981941,38	348502	6981502	**211**

Table 4.2. Selection of the cells in SHyFT.

The following graphs and tables show the accuracy of the selection process and the error in value committed in each case:



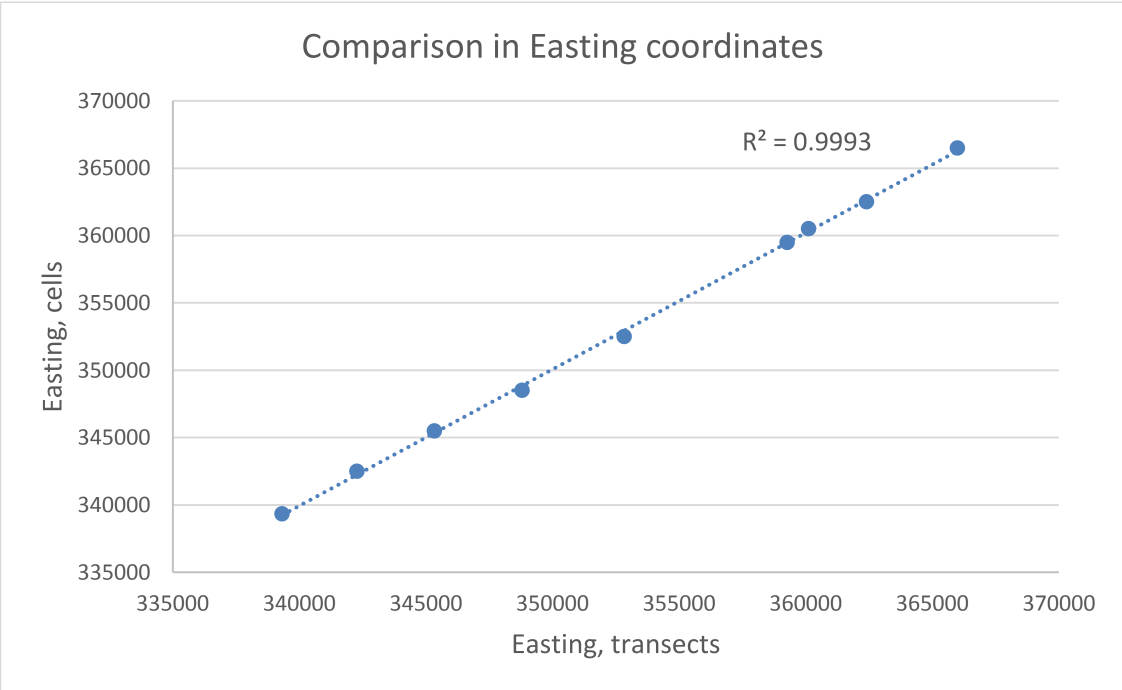
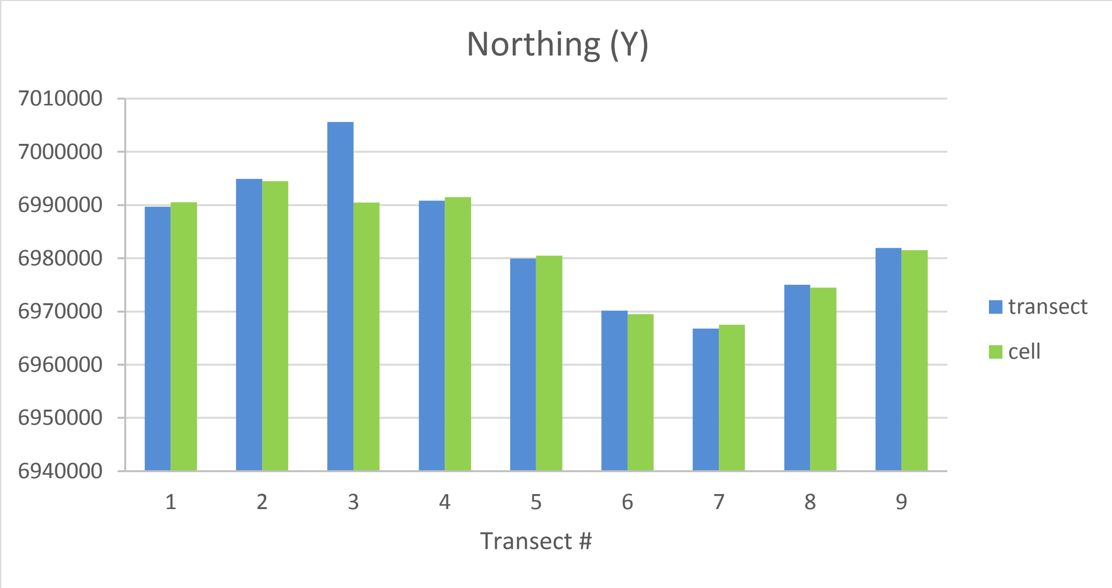


Figure 4.4. Comparison in Easting (X) coordinate between the snow transects and the respective chosen cells.



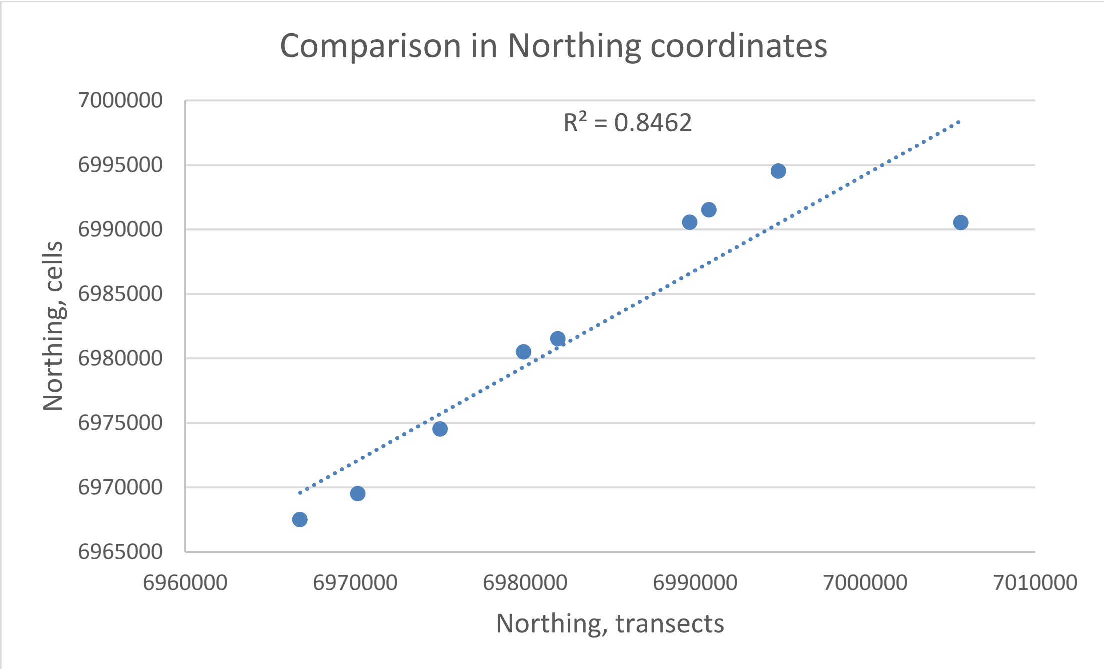


Figure 4.5. Comparison in Northing (Y) coordinate between the snow transects and the respective chosen cells.

Both comparison plots give high values of R^2, which means that the selection of the cells has been successful.

The next table shows numerically the difference of the Easting and Northing coordinates between the transects and their respective chosen cells:

ΔE(X)	ΔN(Y)
6,4	853,01
219,7	411,28
161,12	15140,31
337,21	679,23
92,13	569,12
503,5	658,03
376,32	740,04
223,87	510,07
305,54	439,38

Table 4.3. Difference in value of the snow transect coordinates with their respective chosen cells.

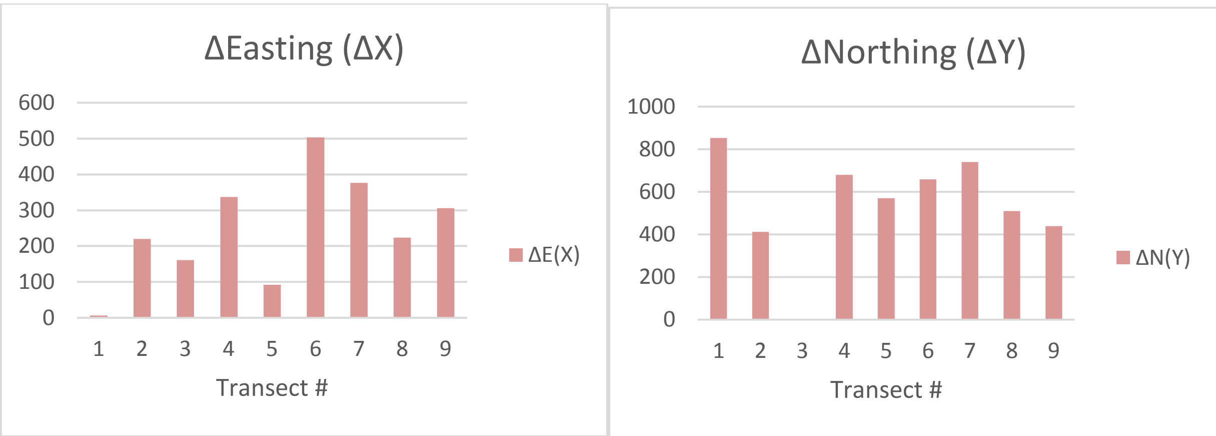


Figure 4.6. Difference between the snow transect coordinates with their respective chosen cells.

Note: *the value of "ΔNorthing (ΔY)" in the snow transect number 3 was not representative so it was decided to leave it out of the plot in order not to distort the graph.*

Looking at the previous results, it can be concluded that the error values are barely negligible compared to the values of the coordinates and that the selection of the cells was good enough to carry on with the study.

The following graph shows in columns the average Snow Water Equivalent per transect and year collected by Statkraft and analysed by the student:

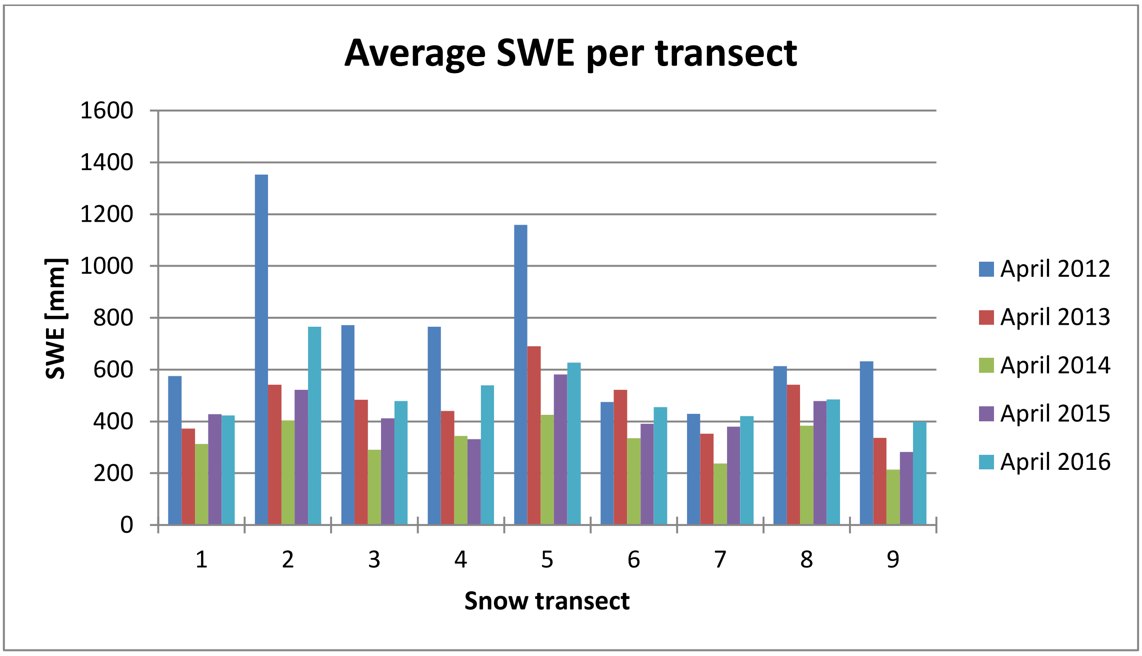


Figure 4.7. Average Snow Water Equivalent per transect and year (observed data by Statkraft).

Finally, the numerical values, expressed in mm of water equivalent, of the average Snow Water Equivalent per transect and year provided by Statkraft can be read in the following table:

	April 2012	April 2013	April 2014	April 2015	April 2016
1	574,59	372,23	312,52	427,82	422,48
2	1352,91	541,49	402,82	521,39	765,8
3	771,6	483,14	290,91	411,47	478,35
4	765,88	440,58	344,31	331,23	539,54
5	1159,25	689,97	425,27	581,41	627,1
6	475,22	522,06	335,07	390,43	455,46
7	429,57	352,3	237,18	379,36	419,99
8	613,33	541,55	383,12	478,08	484,48
9	632,21	336	213,44	281,95	397,78

Table 4.4. Average SWE per transect and year, expressed in mm of water equivalent.

All the field measurements of snow were performed within the first 10 days of April each year.

Hydro-meteorological data

SHyFT is a distributed hydrological model, thus, climate variables along with physiographic data are needed as input. In the case of this study, all this data for the Nea-Nidelva catchment was provided by Statkraft from AROME met data instead of using the already implemented gauging stations situated throughout the catchment. This AROME data package contained registered meteo data from 01/09/2012 to 03/10/2015.

5. SHyFT

This chapter will describe in detail the modelling tool that has been used during the work.

5.1. Introduction

The Statkraft Hydrologic Forecasting Toolbox (SHyFT) is an open source hydrological modelling toolbox developed by Statkraft. It has been optimized for highly efficient modelling of hydrological processes following the paradigm of distributed and lumped models, but recent developments have introduced more physically based and process-level methods. The code is based on an early initiative for distributed hydrological simulation, called ENKI funded by Statkraft and developed at Sintef [13].

SHyFT is used for forecasting inflow in the Statkraft system. This is a flexible system in which model can be custom designed for various purposes. The SHyFT toolbox currently has three different methods for simulating snow accumulation and storage, and these are yet to be evaluated with snow data.

The software provides a high level Python based interface to a modern C++ based underlying API [14].

5.2. Requirements and installation procedure

All necessary requirements for a successful installation and run of SHyFT in Windows can be found in Appendix 2 (sections A and B).

5.3. Model set up

SHyFT is developed to work in an operational environment. It is made out of a subsequence of routines and built in a way that it is user-friendly from the beginning.

The arrangement of the folders and subfolders of SHyFT is the one that can be seen in Figure 4.1. Here it is also shown the location of the input files, program files and configuration files. Please note that the files shown in Figure 1 are the ones that have been used in this study.

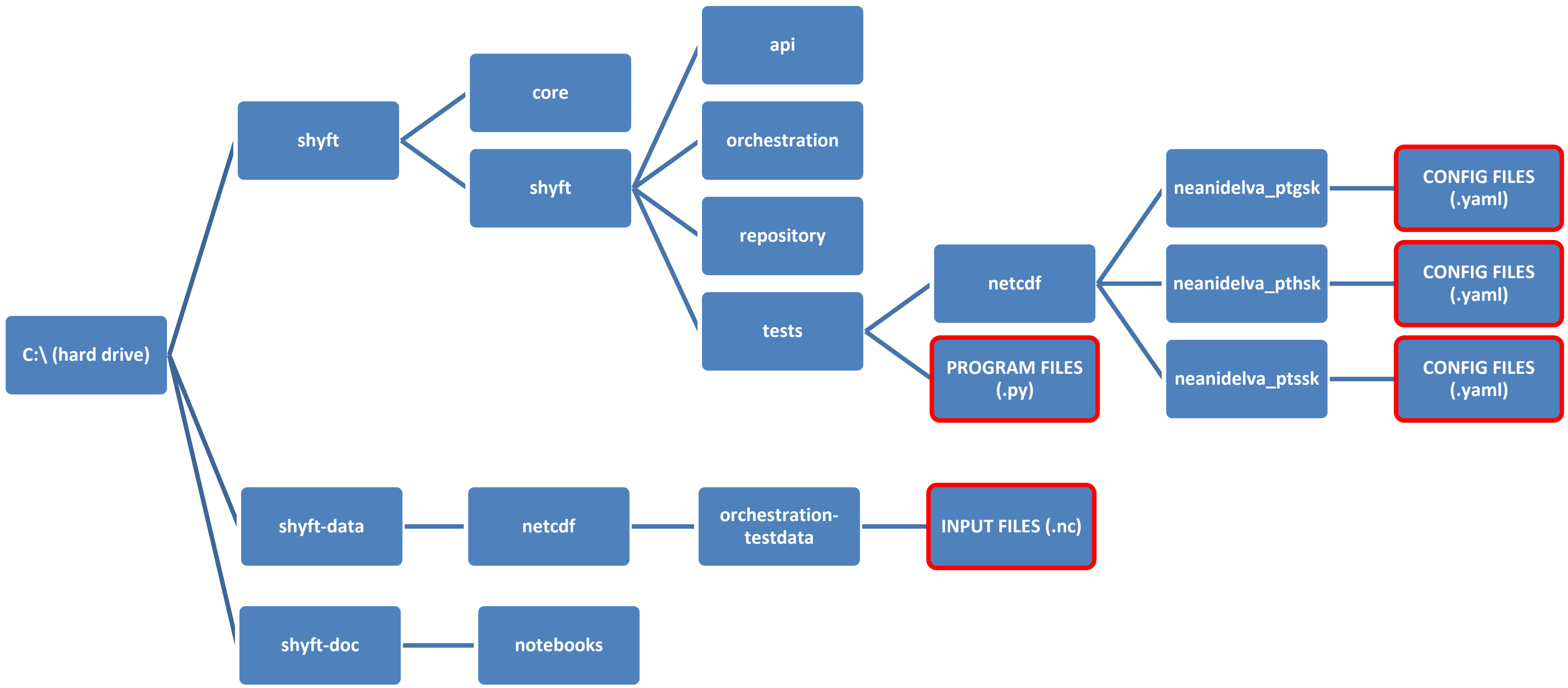


Figure 5.1. SHyFT set up and folders arrangement

The folder C:\shyft\shyft contains the subfolders api, orchestration, repository and tests. A brief explanation of these four is given as follows:

- **api**

  This subfolder contains the python wrappers for the shyft core, which contains basic data structure, cell-models, models and algorithms.

- **orchestration**

  The orchestration folder contains the infrastructure to read the orchestration code. This code uses YAML configuration files to define a simulation run or a calibration.

  In order to successfully run a simulation in SHyFT, incoming observed data such as meteorological or hydrological data must be ingested. This process of data ingestion is called orchestration. Prior to it, there is usually a round of calibration to fill up the internal data structure of SHyFT.

  The core of SHyFT is written in C++. However, all the orchestration code is written in Python and allows the user to add any other own code.

- **repository**

  The repository folder contains the python code that can read the data collected and feed it to the SHyFT core.

- **tests**

  This folder makes the integral part of operation of SHyFT. It contains the information of all routines involved with their respective equations and methods.

  The subfolder netcdf contains all the YAML configuration files which are crucial for SHyFT operation. A list of these can be found in Table 5.1.

The next table, Table 5.1, expands the list of Input Files, Program Files and Configuration Files that have been used in this study and were shown in Figure 4.1.

<table>
<tr><td>INPUT FILES</td><td colspan="2">arome_merged_Nea-Nidelv_Buffer5km_Chunck64x64x32_Complevel4_TimeUnlimited.nc</td></tr>
<tr><td>PROGRAM FILES</td><td colspan="2">- Run_Shyft_Q.py
- Run_Shyft_SCA.py
- Run_Shyft_SWE.py
- Calib_Shyft.py</td></tr>
<tr><td rowspan="4">CONFIGURATION FILES</td><td colspan="2">- neanidelva_region.yaml
- neanidelva_datasets.yaml
- neanidelva_interpolation.yaml
- neanidelva_simulation.yaml</td></tr>
<tr><td>Gamma Snow</td><td>- neanidelva_model.yaml
- neanidelva_calibration.yaml</td></tr>
<tr><td>HBV snow</td><td>- neanidelva_model.yaml
- neanidelva_calibration.yaml</td></tr>
<tr><td>Skaugen snow</td><td>- neanidelva_model.yaml
- neanidelva_calibration.yaml</td></tr>
</table>

Table 5.1. Input Files, Program Files and Configuration Files in SHyFT

Note: *The codes of all Program Files and Configuration Files are given in Appendix 2 (sections C and D).*

5.3.1. Configuration files

- **neanidelva_region.yaml** defines the area of study, in this case the Nea-Nidelva river basin, and connects it to the available physiographic data contained in shyft-data.

- **neanidelva_datasets.yaml** is the calling for the input climate data: precipitation, temperature, wind speed, relative humidity and radiation.

- **neanidelva_interpolation.yaml** returns the interpolation algorithm in the simulation YAML file during the simulation.

- **neanidelva_simulation.yaml** calls the region, datasets and model YAML files, asks the user to define some parameters such as start time, run time step and number of steps, and executes de simulation.

- **neanidelva_model.yaml** contains the parameter set obtained after each calibration.

- **neanidelva_calibration.yaml** executes the calibration of the model against the input discharge netcdf file available in shyft-data. It uses the simulation YAML file as simulator.

  This last two YAML files (model and calibration) vary from one snow routine to another one.

5.3.2. Program files

- **Run_Shyft_Q.py** executes de simulation by importing the simulation YAML file, plots the runoff for the indicated period and saves an excel sheet with the respective values.
- **Run_Shyft_SCA.py** executes de simulation by importing the simulation YAML file, plots the SCA for the indicated period and saves an excel sheet with the respective values.
- **Run_Shyft_SWE.py** executes de simulation by importing the simulation YAML file, plots the SWE for the indicated period and saves an excel sheet with the respective values.
- **Calib_Shyft.py** executes the calibration of the model in question by importing the calibration YAML file.

5.3.3. Input files

SHyFT model set up requires its input data in netcdf format (Network Common Data Frame). As a distributed hydrological model, climate variables along with physiographic data are needed as input. In the case of this study, all the data for the Nea-Nidelva catchment was provided by Statkraft from AROME met data instead of using the different gauging stations situated throughout the catchment.

5.4. Model structure

SHyFT is a distributed hydrological model. It works from regional level to cell level by distributing the input parameters (climate and hydrological data) into the individual cells. Figure 4.2 shows a sketch of the model structure in SHyFT. A description of its elements is given afterwards.

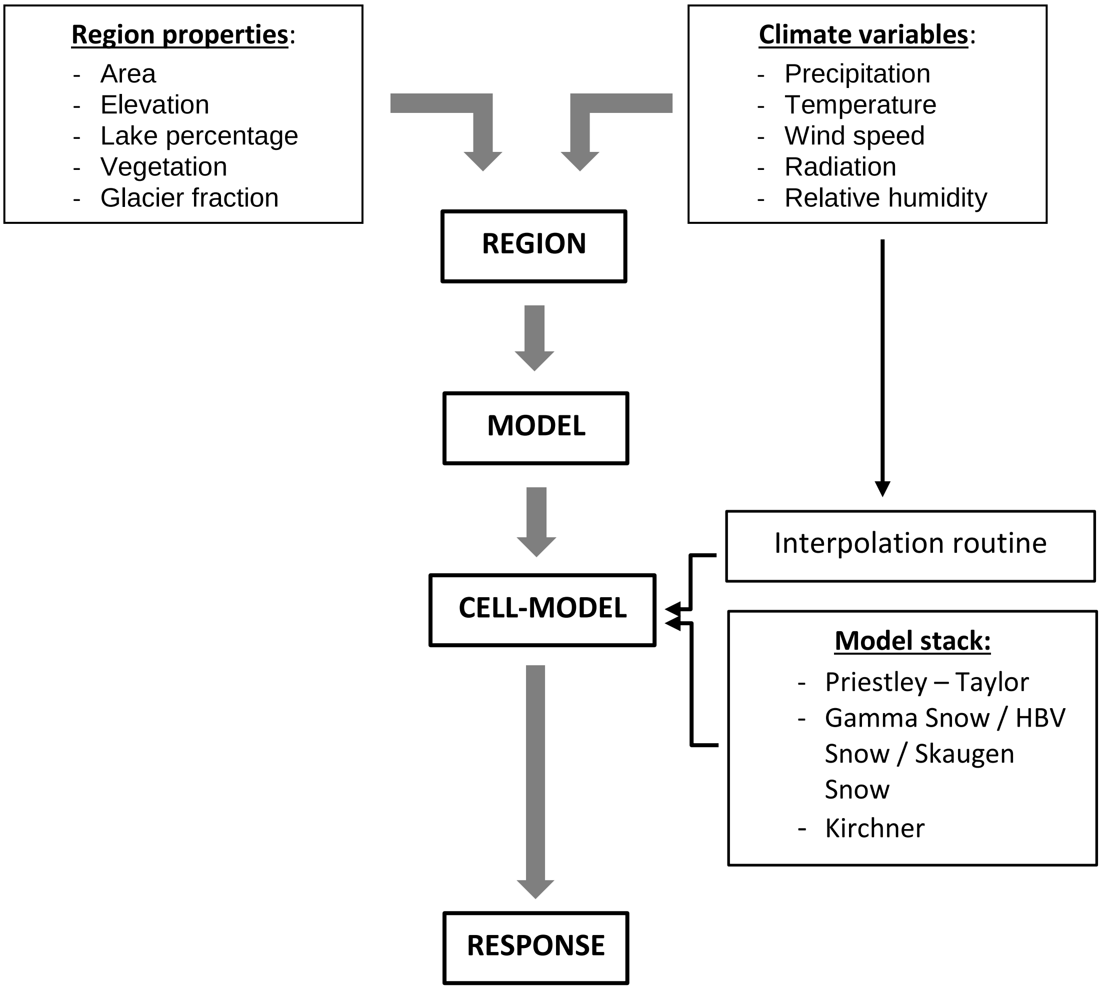


Figure 5.2. SHyFT model structure

- **Region**

It is a geographic region with some data associated that describes the properties at cell or grid level of the region. Physiographic data is treated as static data whereas climate parameters as time dependent variables.

- **Model**

It is a computational model that, given input from the region such as static physiographic properties, variable climate parameters and initial state data, can compute results, runoff forecasts, snow reservoir, new set of state, optimized model parameters, etc. according to the method composition and parameters selected.

- **Cell-Model**

The cell model takes initial cell-state, calibration parameters and cell environment inputs (precipitation, temperature, etc.) and computes the response and a new cell-state for each time step used.

6. MODEL CALIBRATION AND VALIDATION

6.1. Introduction

Calibration is the process of adjusting the model parameters to the values that make the simulation be as close as possible to the observed data. In hydrological modeling, this process is vital and it is used to estimate the values of the free parameters. The free parameters are those that cannot be measured manually and must be found therefore through the process of calibration.

Calibration can be approached either qualitatively or quantitatively. The former is based on visual comparison between observed and simulated plots, while the latter uses numerical criteria as a tool to determine the accuracy of the results. Calibration through visual analysis is highly subjective due to the lack of numerical foundation.

On the other hand, the quantitative method defines and uses an objective function. This function is an error function that calculates the difference between observed and simulated values during the calibration process and represents the goodness of the model performance.

One of the most widely used objective functions is the Nash-Shutcliffe efficiency criteria (R^2):

$$R^2 = 1 - \frac{\sum(Q_S - Q_o)^2}{\sum\left(Q_o - Q_{o_m}\right)^2}$$

Where,

- ✓ Qo Observed runoff
- ✓ Qs Simulated runoff
- ✓ Qom Mean of observed runoff

R^2 takes values from -∞ to +1; the closer to 1, the better model performance.

Since the input AROME data is only available for 3 years (from 01/09/2012 to 03/10/2015), first two years are used for calibration and the last year is utilized for validating the model.

6.2. Calibration of the parameters

For this study, the model was calibrated separately for each of the three snow routines (Gamma, HBV and Skaugen).

In order to maximize the R^2 coefficient, an automatic calibration was performed using SCE-UA and the default values as initial point. The calibration was run from 01/09/2012 to 01/09/2014

The following tables present the parameters to be calibrated in each model, as well as their description, default original values, values after calibration and R^2:

GAMMA SNOW			
Parameter	**Description**	**Default value**	**Calibrated value**
ae_scale_factor	Actual evapotranspiration scale factor	1.5	1.5
fast_albedo_decay_rate	Albedo decay rate during melt [days]	6.753	6.001
slow_albedo_decay_rate	Albedo decay rate in cold conditions [days]	37.173	30.000
glacier_albedo	Glacier ice fixed albedo	0.4	0.4
initial_bare_ground_fraction	Initial bare ground fraction	0.04	0.04
max_albedo	Maximum albedo value	0.9	0.9
min_albedo	Minimum albedo value	0.6	0.25
max_water	Maximum liquid water content	0.1	0.1
snow_cv	Spatial coefficient of variation of fresh snowfall	0.4	0.4
tx	Snow and rain threshold temperature [ºC]	-0.575	-0.500
snowfall_reset_depth	Snowfall required to reset albedo [mm]	5.0	5.0
surface_magnitude	Surface layer magnitude	30.0	30.0
wind_const	Intercept in turbulent wind function	1.0	3.500
wind_scale	Slope in turbulent wind function [m/s]	1.896	0.772
winter_end_day_of_year	End of the winter season	100	100
c1	First parameter in Kirchner model	-3.336	-3.942
c2	Second parameter in Kirchner model	0.334	0.549
c3	Third parameter in Kirchner model	-0.125	-1.181
scale_factor	Precipitation correction scale factor	1.0	0.7
albedo	Albedo in Priestley-Taylor stack	0.2	0.2
alpha	Alpha in Priestley-Taylor stack	1.26	1.26
alpha	routing.alpha	-	0.9
beta	routing.beta	-	3.0
velocity	routing.velocity	-	0.0
NASH (R^2)	**Nash-Shutcliffe efficiency**	**0.733**	

Table 6.1. Parameters to be calibrated in Gamma snow.

The Gamma snow based model was calibrated three times until the parameters stabilized between their lowest and highest limit values.

1st R^2 = 0.726

2nd R^2 = 0.731

3rd R^2 = 0.733

HBV SNOW			
Parameter	**Description**	**Default value**	**Calibrated value**
ae_scale_factor	Actual evapotranspiration scale factor	0.05	0.05
cfr	Degree-day factor for snow refreezing	0.005	0.0
cx	Degree-day factor for snow melt	0.5	0.431
lw	Liquid water	0.029	00.145
ts	Snow melt and snow refreezing threshold temperature [ºC]	0.144	-0.356
tx	Snow and rain threshold temperature [ºC]	0.197	-0.058
c1	First parameter in Kirchner model	-2.534	-4.092
c2	Second parameter in Kirchner model	0.565	0.578
c3	Third parameter in Kirchner model	-0.066	-0.011
scale_factor	Precipitation correction scale factor	1.0	1.0
dtf	Glacier melt	-	6.0
albedo	Albedo in Priestley-Taylor stack	0.2	0.2
alpha	Alpha in Priestley-Taylor stack	1.26	1.26
alpha	routing.alpha	-	0.9
beta	routing.beta	-	3.0
velocity	routing.velocity	-	0.0
NASH (R^2)	**Nash-Shutcliffe efficiency**	**0.755**	

Table 6.2. Parameters to be calibrated in HBV snow.

The HBV snow based model was calibrated five times until the parameters stabilized between their lowest and highest limit values.

1st R^2 = 0.692

2nd R^2 = 0.726

3rd R^2 = 0.746

4th R^2 = 0.746

5th R^2 = 0.746

6th R^2 = 0.755

SKAUGEN SNOW			
Parameter	**Description**	**Default value**	**Calibrated value**
alpha_0	skaugen_snow.alpha_0	40.77	40.558
cfr	Degree-day factor for snow refreezing	0.01	0.010
cx	Degree-day factor for snow melt	2.5	0.586
d_range	Skaugen_snow.d_range	113.0	110.718
max_water_fraction	Maximum water fraction value	0.1	0.345
ts	Snow melt and snow refreezing threshold temperature [ºC]	0.16	0.138
c1	First parameter in Kirchner model	-3.336	-3.919
c2	Second parameter in Kirchner model	0.334	0.529
c3	Third parameter in Kirchner model	-0.125	-0.020
ae_scale_factor	Actual evapotranspiration scale factor	1.5	0.765
dtf	Glacier melt	-	6.0
albedo	Albedo in Priestley-Taylor stack	0.2	0.2
alpha	Alpha in Priestley-Taylor stack	1.26	1.26
alpha	routing.alpha	-	0.9
beta	routing.beta	-	3.0
velocity	routing.velocity	-	0.0
NASH (R^2)	**Nash-Shutcliffe efficiency**	**0.784**	

Table 6.3. Parameters to be calibrated in Skaugen snow.

The Skaugen snow based model was calibrated five times until the parameters stabilized between their lowest and highest limit values.

1st R^2 = 0.737

2nd R^2 = 0.768

3rd R^2 = 0.780

4th R^2 = 0.782

5th R^2 = 0.784

6.3. Calibrated simulation

The three models were updated with the values of the calibrated parameters shown in the tables above, and new runoff simulations were run for each of the three snow models.

The observed runoff data is available at the gauging station called Aune (123.21.0), next to Tydal, and it is provided by SeNorge.

Note: *although the river branch is regulated by the reservoir upstream, the observed data for the study was analysed and it was accepted to be non-regulated. That way, one is able to use it for the comparison with the simulated results obtained from SHyFT.*

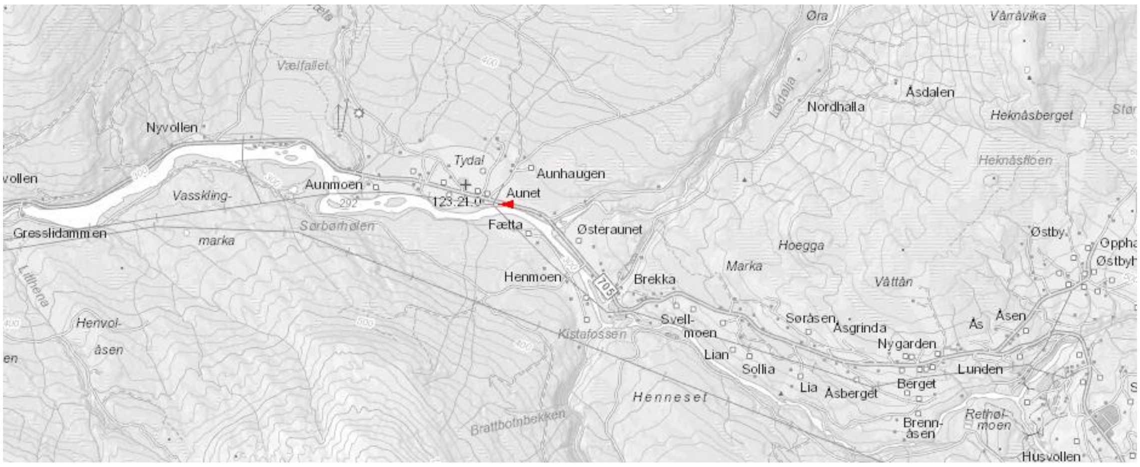


Figure 6.1. Aune runoff gauging station.

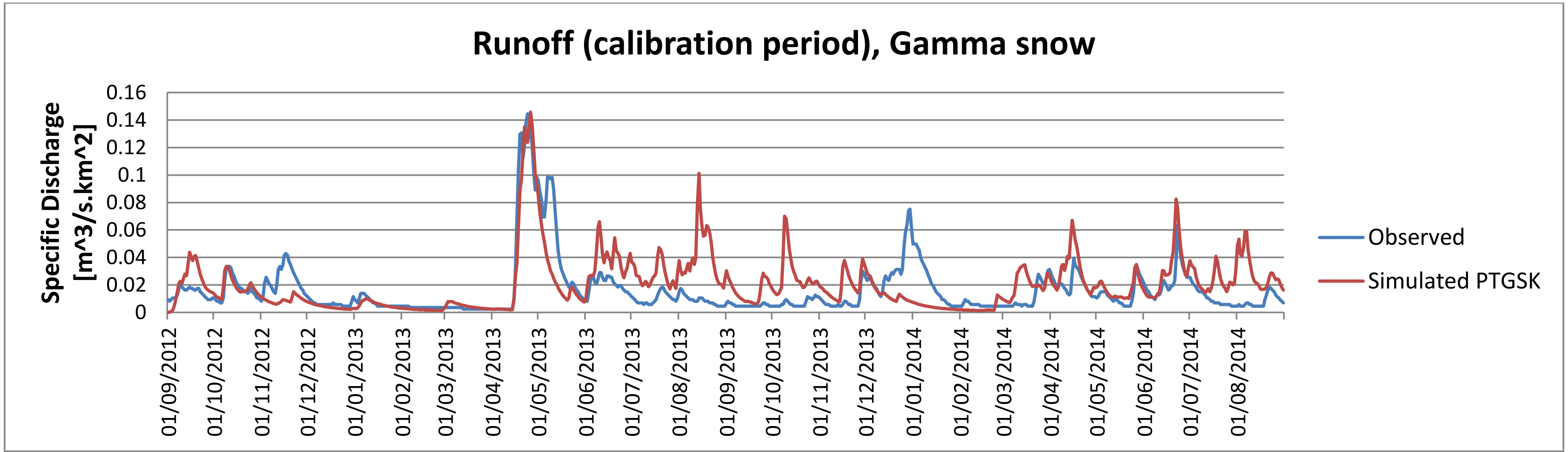
The same way as in *Section 3: Data and Work Methodology,* cell number 2745 was found to be the closest one to the gauging station coordinates.

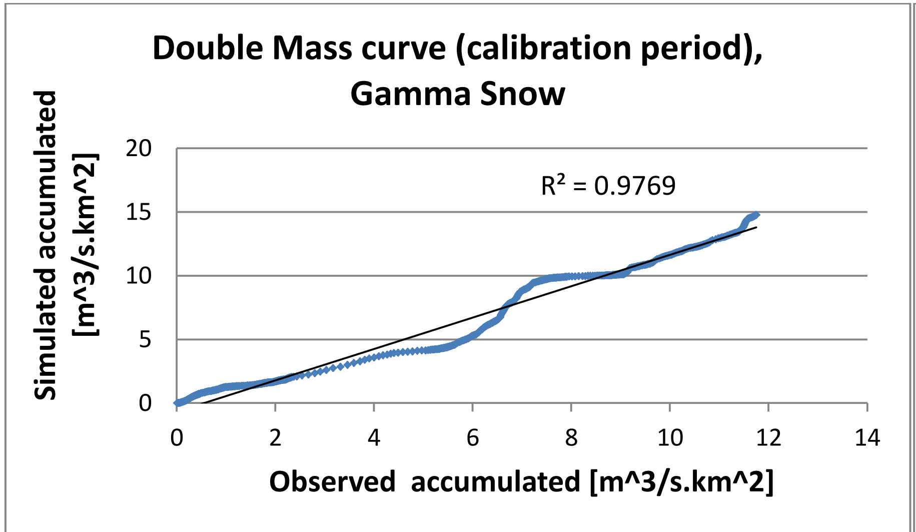
Results of the **calibration period** are shown in **Figure 6.2., 6.3. and 6.4.**

6.4. Validation of the model

Once the model was calibrated, a new simulation was run from 01/09/2014 to 01/09/2015 to validate the model.

Results of the **validation period** are shown in **Figure 6.5., 6.6. and 6.7.**





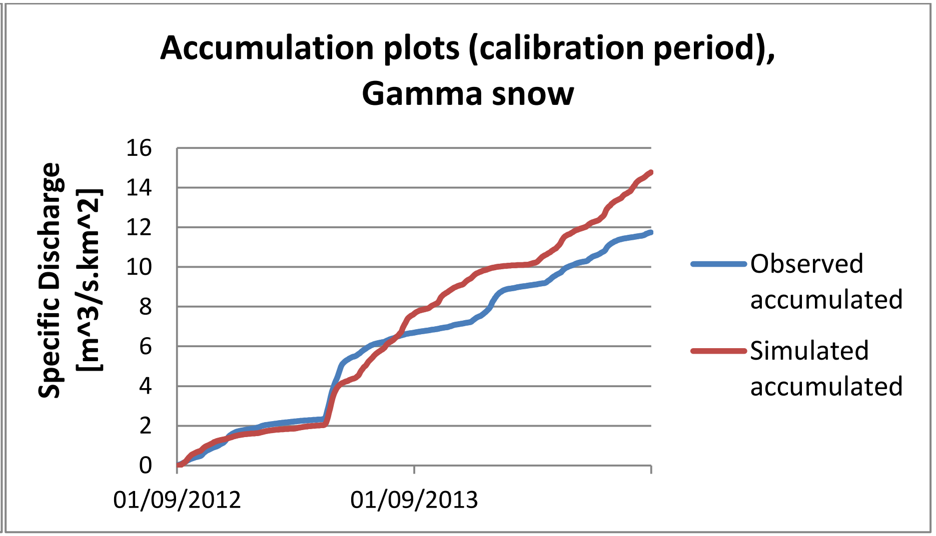


Figure 6.2. Calibration period for Gamma snow.

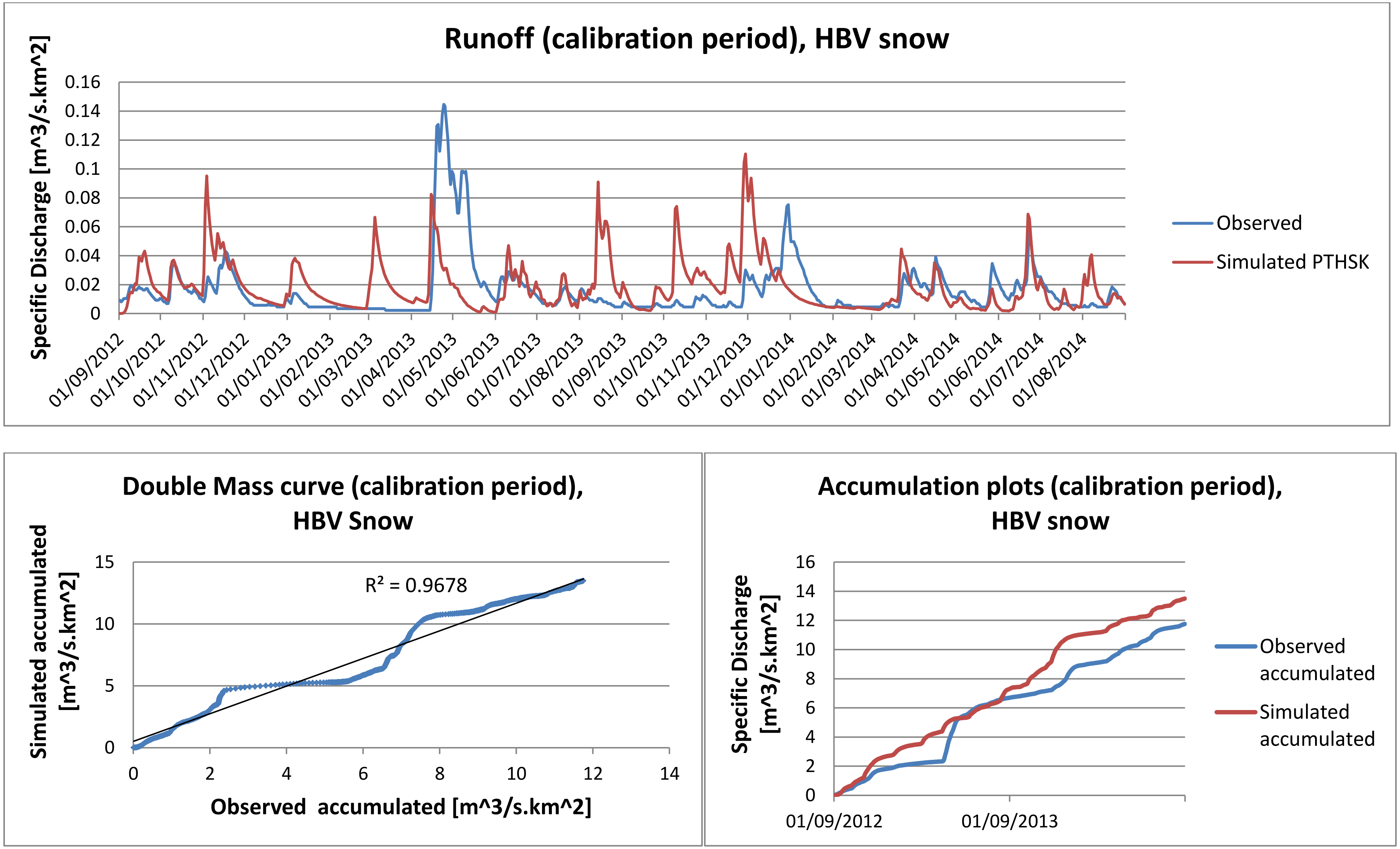
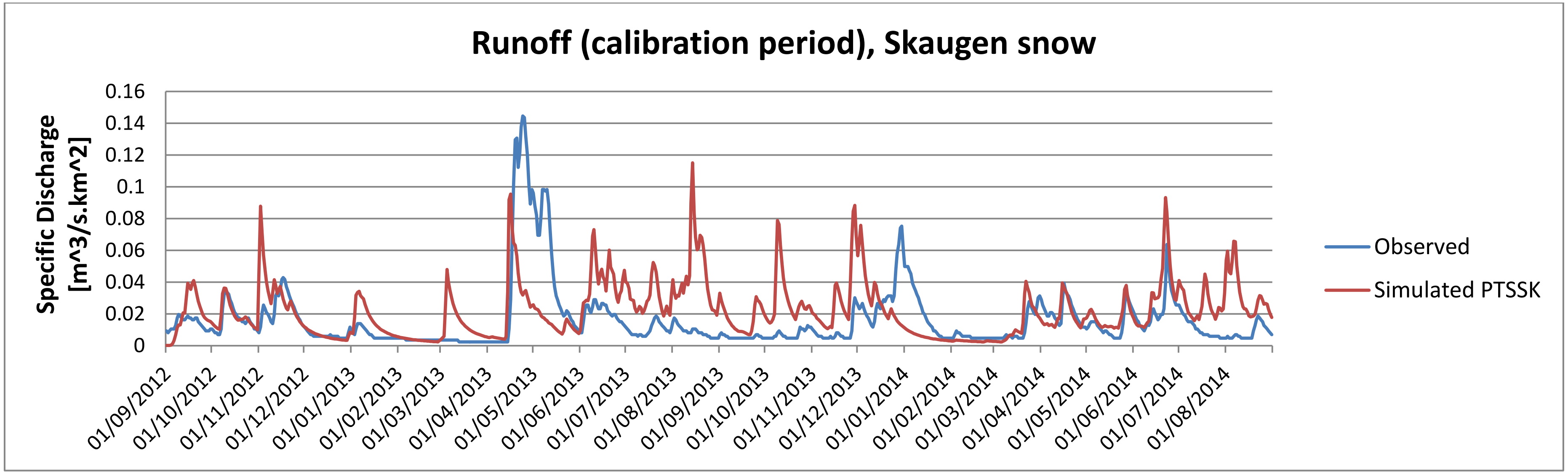
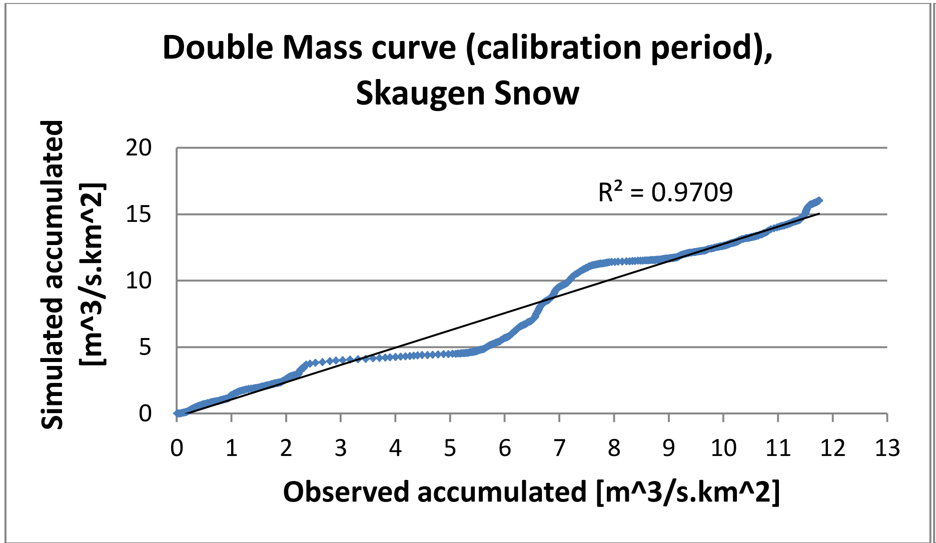


Figure 6.3. Calibration period for HBV snow.





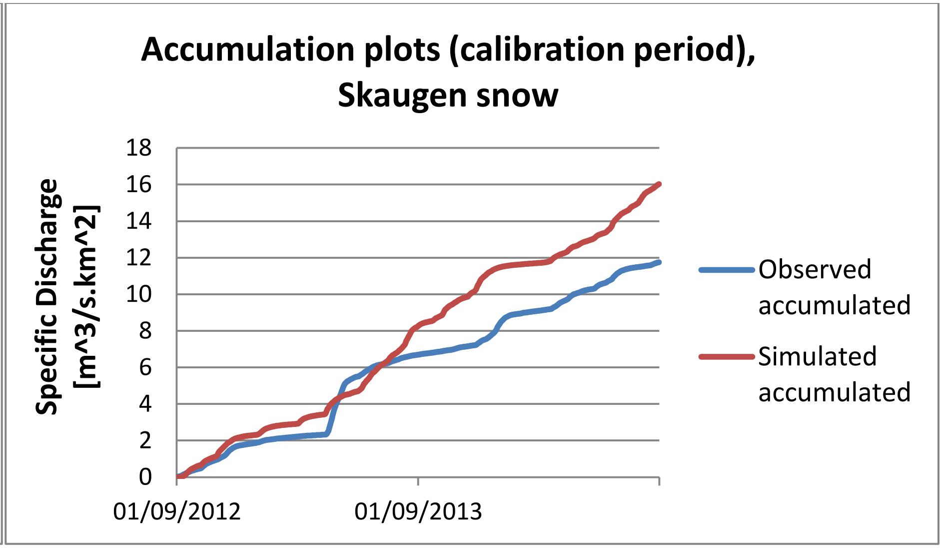
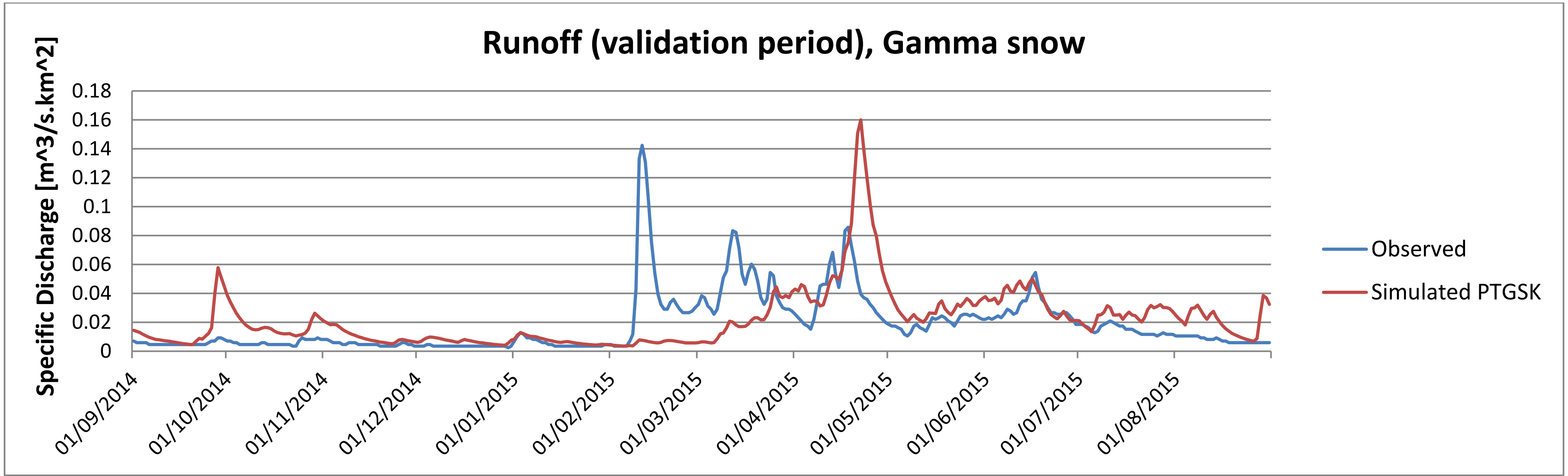
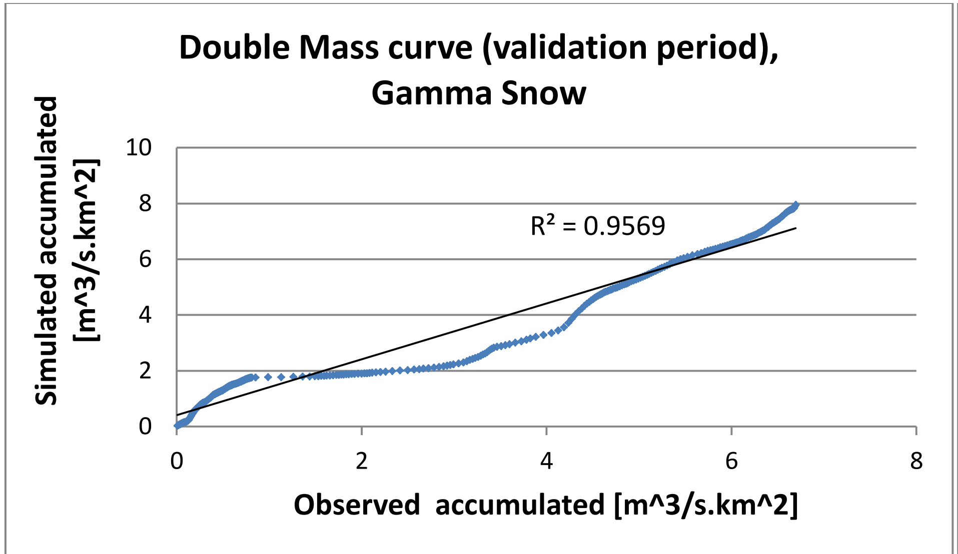


Figure 6.4. Calibration period for Skaugen snow.





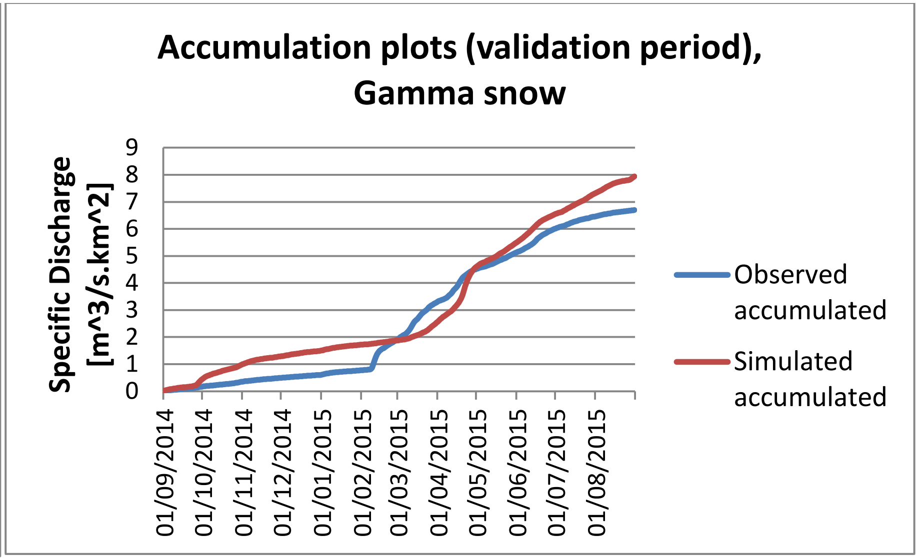


Figure 6.5. Validation period for Gamma snow.

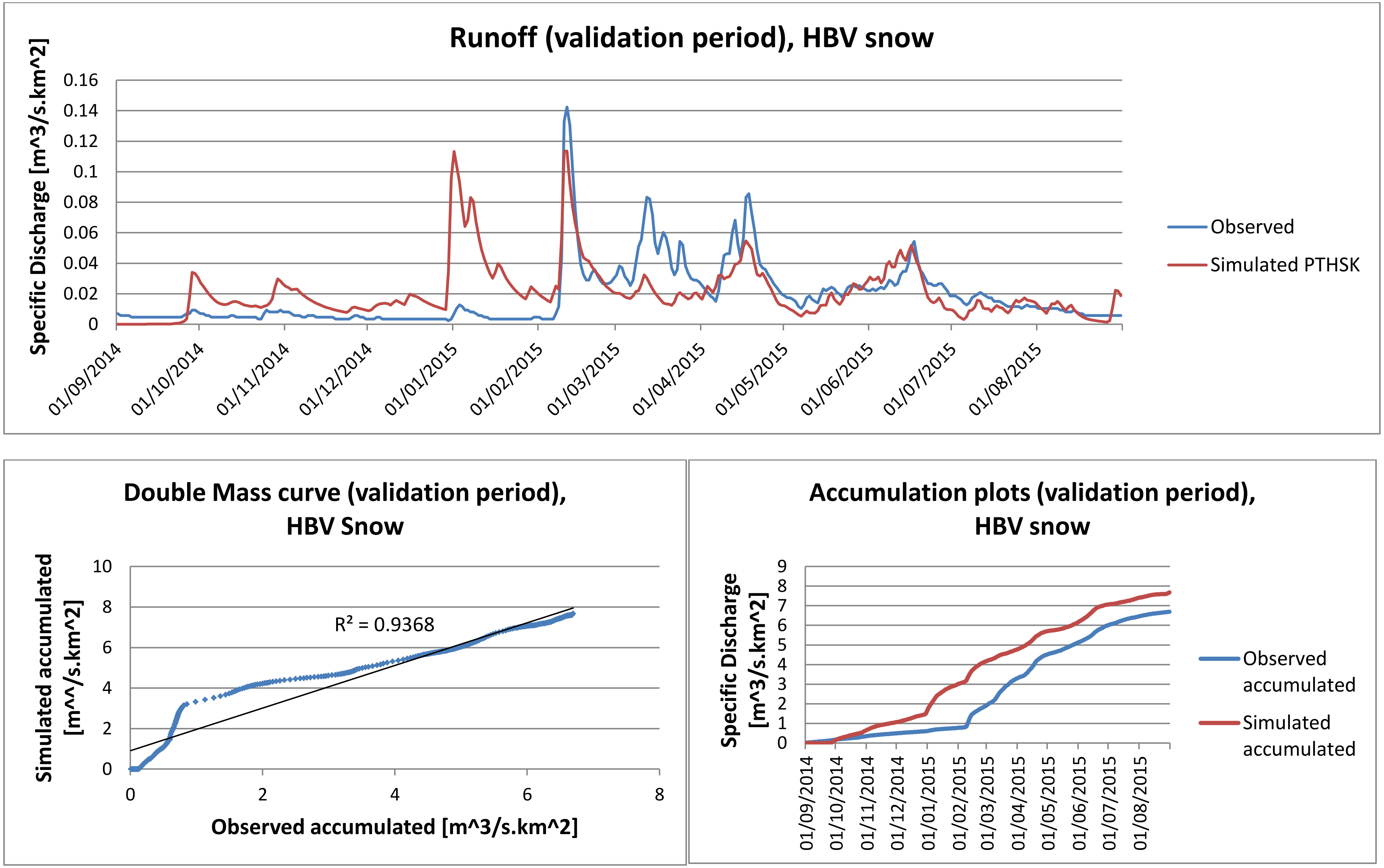
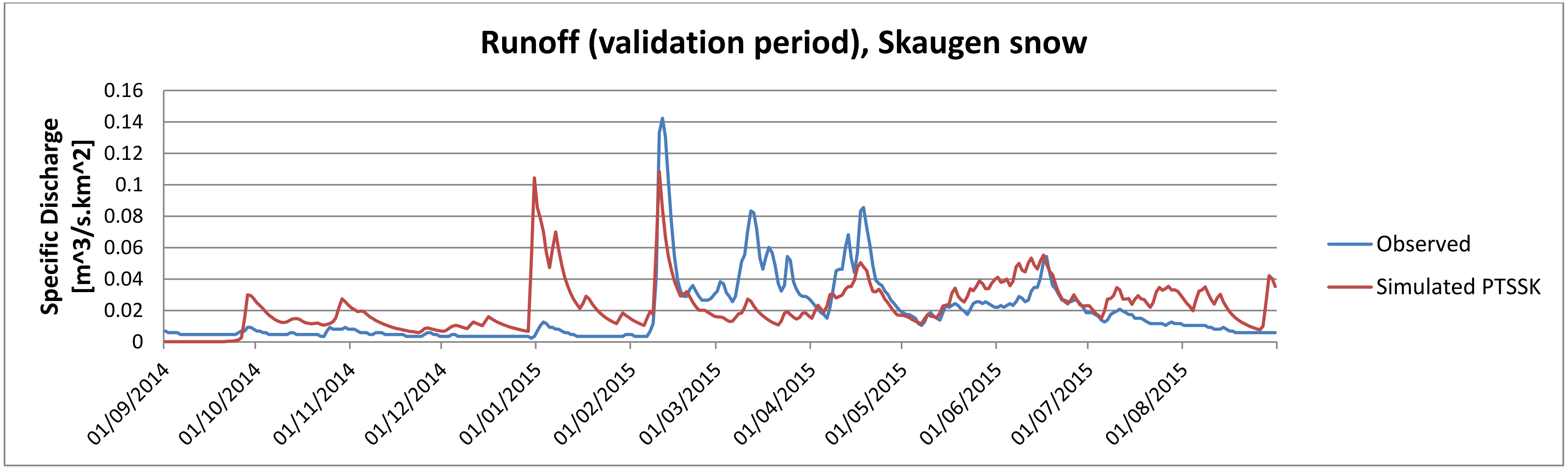
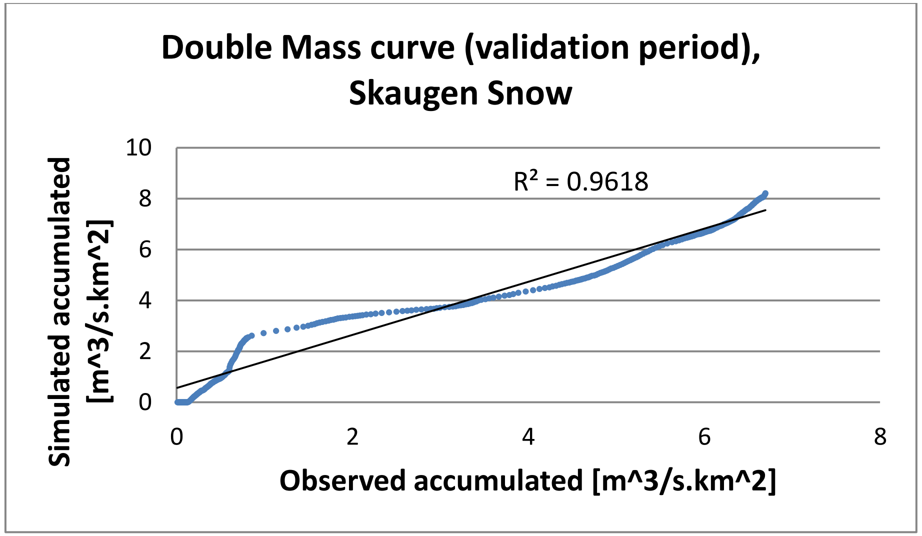


Figure 6.6. Validation period for HBV snow.





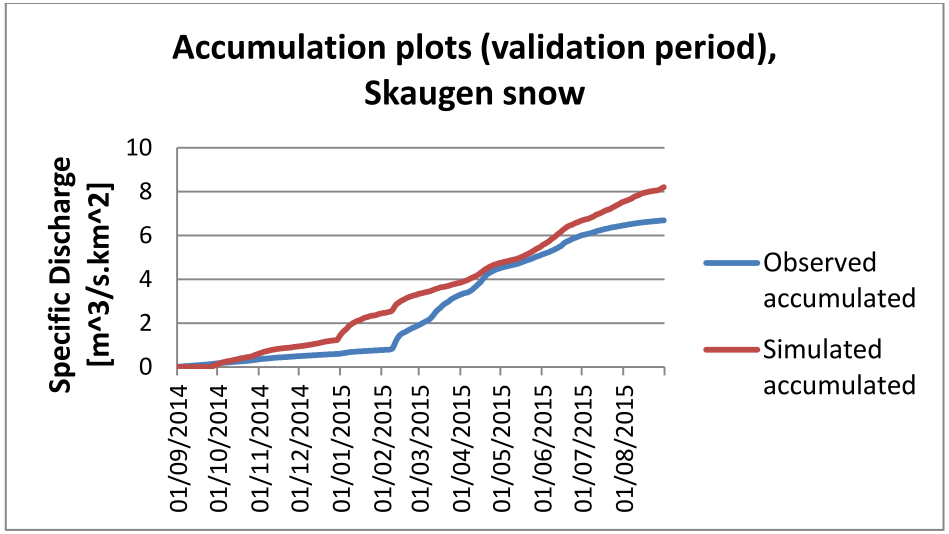


Figure 6.7. Validation period for Skaugen snow.

7. SNOW SIMULATIONS

In order to perform the comparison between observed snow data and simulated snow results from SHyFT, the three snow routines were run for a simulation period within 01/09/2012 and 01/09/2015.

The observed snow data was available as Snow Water Equivalent measurements along 9 different transects in the south-east region of the Nea-Nidelva catchment (see *Section 3: Data and Work Methodology*). Therefore, the SHyFT code was accordingly modified to extract SWE values of the most representative grid cells for the same period.

All the numerical results are presented in the following tables, where *obs* stands for *Observed data* and *sim*, for *Simulated data in SHyFT*:

	GAMMA SNOW SWE [mm]					
	April 2013		April 2014		April 2015	
transect #	**obs**	**sim**	**obs**	**sim**	**obs**	**sim**
1	372,23	314,300	312,52	286,6368	427,82	414,087
2	541,49	362,892	402,82	401,3548	521,39	502,175
3	483,14	324,072	290,91	290,7888	411,47	360,424
4	440,58	318,168	344,31	348,5348	331,23	357,306
5	689,97	306,291	425,27	421,0507	581,41	399,154
6	522,06	222,501	335,07	337,0466	390,43	300,308
7	352,3	331,456	237,18	346,9056	379,36	369,008
8	541,55	251,464	383,12	292,9262	478,08	297,018
9	336	338,334	213,44	371,9629	281,95	368,718

Table 7.1. Snow Water Equivalent, Gamma snow routine.

	HBV SNOW SWE [mm]					
	April 2013		April 2014		April 2015	
transect #	**obs**	**sim**	**obs**	**sim**	**obs**	**sim**
1	372,23	315,099	312,52	335,445	427,82	411,074
2	541,49	368,810	402,82	482,878	521,39	523,592
3	483,14	272,225	290,91	372,094	411,47	397,738
4	440,58	370,611	344,31	491,757	331,23	403,074
5	689,97	361,476	425,27	447,450	581,41	464,083
6	522,06	237,271	335,07	334,236	390,43	302,288
7	352,3	427,566	237,18	499,553	379,36	401,926
8	541,55	251,509	383,12	322,177	478,08	325,569
9	336	316,517	213,44	397,792	281,95	327,656

Table 7.2. Snow Water Equivalent, HBV snow routine.

	SKAUGEN SNOW SWE [mm]					
	April 2013		April 2014		April 2015	
transect #	**obs**	**sim**	**obs**	**sim**	**obs**	**sim**
1	372,23	302,224	312,52	315,099	427,82	408,146
2	541,49	367,359	402,82	482,878	521,39	523,592
3	483,14	230,462	290,91	297,371	411,47	312,239
4	440,58	304,805	344,31	364,279	331,23	334,542
5	689,97	339,188	425,27	432,117	581,41	441,409
6	522,06	232,321	335,07	339,188	390,43	297,371
7	352,3	304,805	237,18	367,996	379,36	323,390
8	541,55	250,906	383,12	311,310	478,08	325,249
9	336	334,542	213,44	382,865	281,95	353,128

Table 7.3. Snow Water Equivalent, Skaugen snow routine.

8. RESULTS AND DISCUSSION

8.1. Calibration

The three models are calibrated and good R^2 values are obtained: 0.733 for Gamma Snow, 0.755 for HBV Snow and 0.784 for Skaugen Snow.

Figure 6.2 shows the runoff comparison between observed and simulated discharge after the calibration from 01/09/2012 to 01/09/2014 of the Gamma snow model. It can be seen that both plots are very alike, except for the second half of year 2013. The double mass plot shows a high R2 value of 0.9769. The accumulative plots show a difference in discharge of almost 3 m^3/s.km^2 by the end of the period.

In the case of the HBV snow model, Figure 6.3, the simulated runoff for the calibration period does not match the observed data at all times. However, the double mass curve presents a good R2 value of 0.9678. The accumulative comparison shows a difference of 2 m^3/s.km^2 by the end of the calibration period.

Regarding the Skaugen snow model, Figure 6.4, the observed and simulated runoffs present both similarities and discrepancies. It can be said that the simulated discharge follows the trend of the observed data, but the volume of water is not accurate. The double mass curve present a value of R2 of 0.9709 and the accumulative plots, a difference in discharge of 4 m^3/s.km^2 by the end of the calibration period.

8.2. Validation

Figure 6.5 shows the runoff comparison between observed and simulated discharge for the validation period from 01/09/2014 to 01/09/2015 of the Gamma snow model. It can be seen that both plots are very alike, except for the observed discharge peak of February 2015, which is delayed by the simulation to May 2015. The double mass plot shows a R2 value of 0.9569 and the accumulative plots show a difference in discharge of approximately 1 m^3/s.km^2 by the end of the period.

In the case of the HBV snow model, Figure 6.6, the validation presents good results of observed and simulated discharges after February 2015. The double mass curve gives a R2 value of 0.9368 and the accumulative plots, a difference in discharge of around 1 m^3/s.km^2 by the end of the period.

Figure 6.7 shows the simulation of the Skaugen snow model, which is similar to the one of the HBV model commented previously. The R2 value of the double mass curve is 0.9618 and the difference in discharge of the accumulative plots is approximately 1.5 m^3/s.km^2 by the end of the validation period.

8.3. Snow simulations

First, it was decided to compare the SWE simulation results per year and see which of the three years showed the biggest difference between each of the three snow routines. This way, it would be more intuitive to evaluate those discrepancies between them. Once this particular year had been selected, the analysis would be carried out focusing on the results of that year.

The results per year are shown in the following figures, as percentage of difference in values between observed and simulated SWE over observed:

$$\% \, of \, diff = \frac{|obs - sim|}{obs} \times 100$$

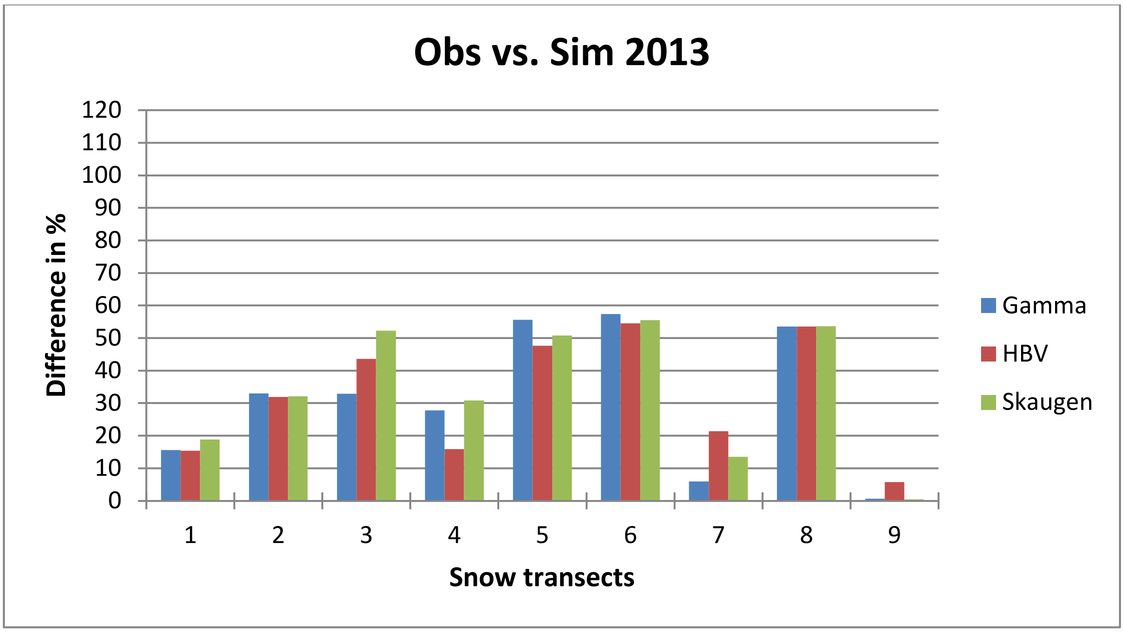


Figure 8.1. Difference in SWE, 2013.

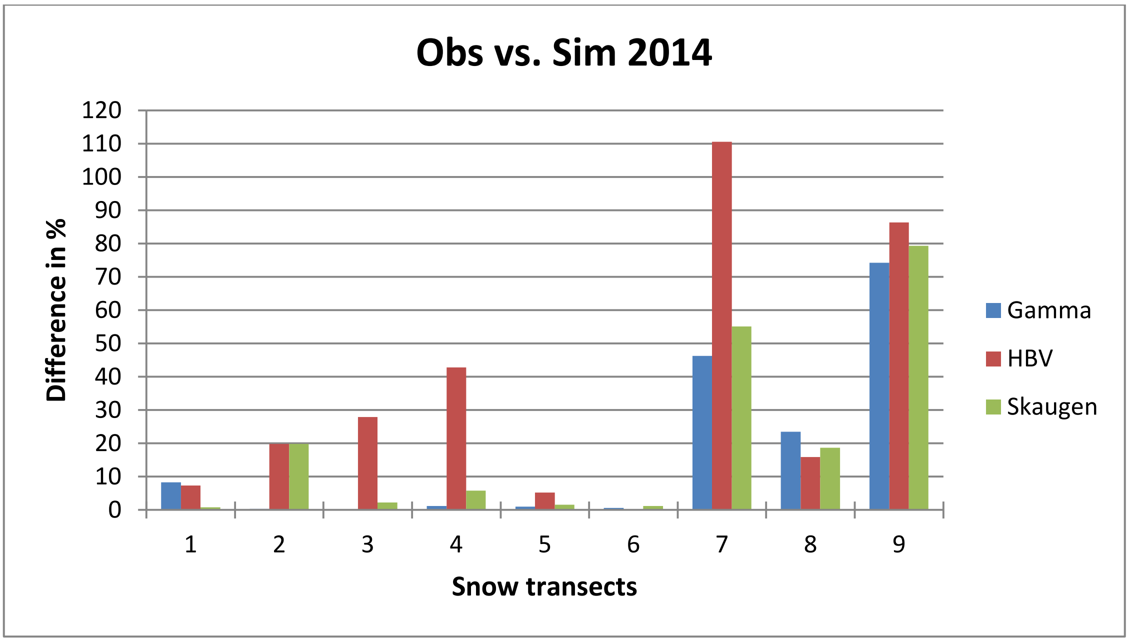


Figure 8.2. Difference in SWE, 2014.

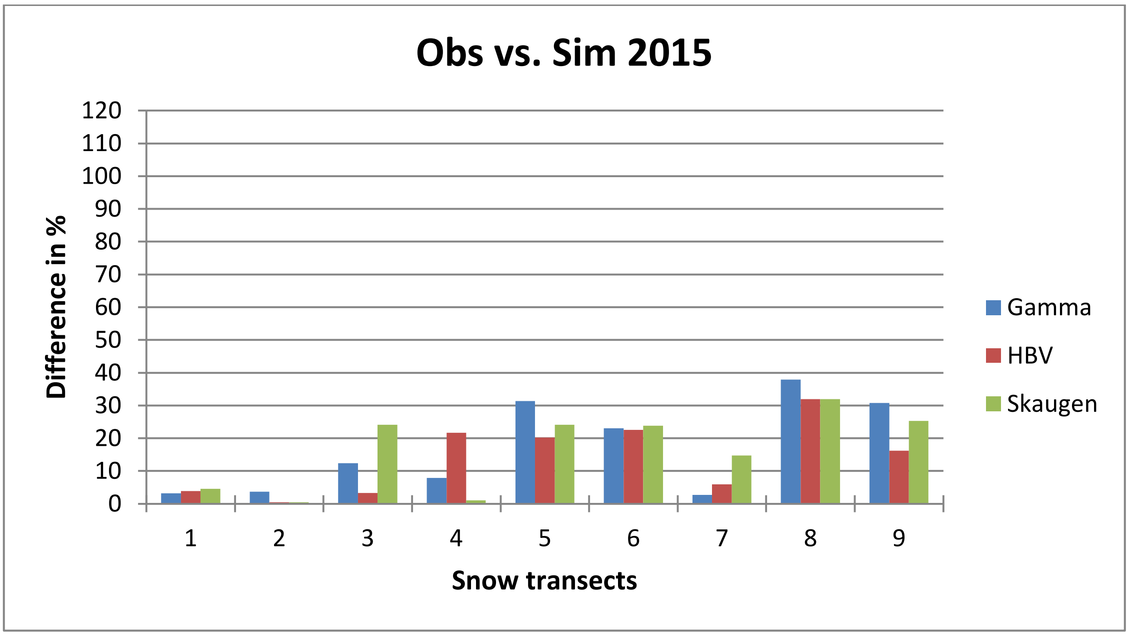


Figure 8.3. Difference in SWE, 2015.

The biggest differences between the models along the transects can be found in 2014 and it shows how HBV snow lacks accuracy in a greater way than the other two.

The following potential causes can be commented on:

8.3.1. Elevation (masl)

The quality of hydrological modelling depends on how well a model simulates the regional detail and topographic characteristics of the region, especially in mountainous regions. However, most mountainous regions exhibit higher errors in gridded datasets because they are usually based on an uneven number of stations that are mostly located at lower elevations [15].

That is why elevation was one of the first thoughts to comment on the inaccuracies and differences of the models. The snow transects are distributed in the following average elevations:

transect #	H [masl]
1	743,53
2	955,14
3	874,17
4	982,93
5	1065,9
6	987,62
7	1215,08
8	959,87
9	999,69

Table 8.1. Elevation distribution of the snow transects.

Reordering from lowest to highest elevation, the percentage of differences can be plotted as follows:

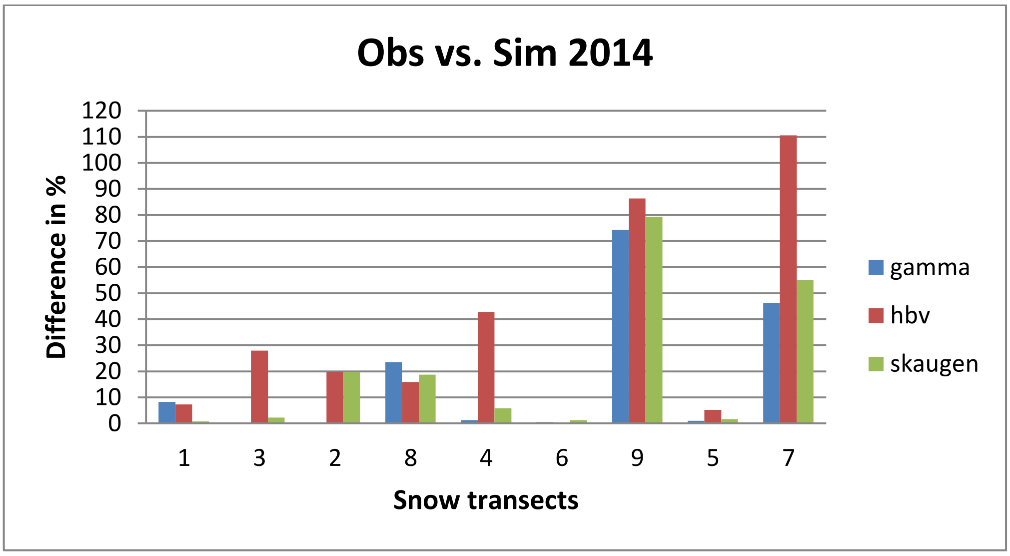


Figure 8.4. Difference in SWE, 2014, ordered from lowest to highest elevation.

This analysis, however, does not provide any too reasonable conclusions. It can be seen that none of the three snow routines follows a tendency of higher inaccuracy with elevation in masl.

8.3.2. Forested and bare areas

Another topic of interest is the location of the measurements and run simulations. In hydrological modelling, parameters that indicate and evaluate the fraction of bare ground or forested regions are not always included in the different models.

In the case of this study, Gamma snow and HBV snow routines take into consideration this factor. Gamma snow includes in the model set up the *initial_bare_ground_fraction* parameter and HBV snow distinguishes between forested and clear areas. However, the Skaugen snow routine does not contemplate this factor. In this case, the points are zones not too forested:

Figure 8.5. Average location points of the snow transects.

All the transects are located in zones with similar amount of low vegetation or bare ground. However, in the case of number 9, the transect goes from almost the lake up to the higher mountainous region next to it (see *Appendix 1, Transect 9*). This might have caused issues when simulating the snow for a single grid cell, approximating one of the cells to that transect, which combines different types of ground. It can be seen that transect number 9 presents high percentage of difference between observed and simulated snow data.

8.3.3. Others

- Redistribution of the snow

  The effect of wind in mountainous regions must not be neglected. Wind redistributes the snow cover, which is not contemplated by all hydrological models. The effect of wind is often more dependent on the elevation gradient than on the elevation itself; one surface might be more affected by the wind if it is part of a hill than another higher in elevation but flat one.

  In this study, transect number 9 presents high differences between observed and simulated snow data. This transect is located in a mountainous and open area which is probably hit by wind during most of the time. Same applies to transect number 7.

  Transects 5 and 6, even though they are located at high elevations, they are not that much part of inclined surfaces.

- Orientation of the surface

  Another important aspect is the incident radiation. More specifically, the heat provided by the sun is not the same during all hours in a day. Whether a surface is facing the sun during the daylight or during the twilight or evening does make a difference in the snow energy balance.

- Precission of time step

  The hydrological models used in this study have been run at a daily time step, but there is the possibility to reduce that step to hourly or even shorter, which would increase the accuracy of the simulations and results.

9. CONCLUSIONS

The study "Evaluation of snow simulations in SHyFT" calibrated and ran Statkraft's Hydrological Forecasting Toolbox in order to simulate snow data and compare and evaluate it against real observed snow measurements from the field.

This study was applied on the Nea-Nidelva catchment, in central Norway, and more specifically, on the south-east part where the observed snow data was available.

First of all, and becoming the most time consuming task of the work, SHyFT needed to be installed and prepared to start running simulations on the personal computer of the author of the thesis. After many failures and unsuccessful trials, finally the software was ready to run.

Due to the fact that the hydro-meteorological input data from AROME was only available from 2012 to 2015, it was decided to run the calibration period from 2012 to 2014 and the validation period of the model from 2014 to 2015.

The calibration results showed a Nash-Shutcliffe efficiency criteria (R^2) of 0.733 for Gamma Snow, 0.755 for HBV Snow and 0.784 for Skaugen Snow. Once the three versions of the model were calibrated, runoff simulations for the calibration period were performed and compared to the available observed discharge data for the same period. The unregulated observed runoff data was obtained from SeNorge and was provided from the gauging station called Aune, next to Tydal.

Even though the double mass analysis for the three models gave high values of R2, which means that the accuracy of the simulation is good, Gamma snow showed the best results. HBV and Skaugen had some discrepancies in both in volume and timing, compared to the observed values. However, during the validation period, HBV and Skaugen showed better-simulated runoffs than during their respective calibration periods.

Before starting to simulate snow, the observed snow data was analysed. Measurements of Snow Water Equivalent along 9 different snow transects for five years were provided by Statkraft. First, the average values of those measurements per transects were obtained. Then, in SHyFT, the closest and most representative grid cells to the average points of each transect were found.

In SHyFT, the SWE of those grid cells was extracted for April of the same years as the observed data. The comparison was carried out by observing the percentage of the difference between observed and simulated discharge per transect, and it was concluded that all of the three models present flaws and that the simulations were sometimes poor.

The models behaved in a way in 2013 and 2015 that they presented close difference values of SWE, whereas in 2014, the results of each model differ significantly from each other.

However, there are other sources of uncertainties not discussed here that may establish a range of possible impacts on hydrological simulations. For example, the hydrological model used in this study runs at a daily time step, which can be increased to hourly to refine the model performance.

10. REFERENCES

[1] S. Kolberg and L. Gottschalk: Stability of distributed snow depletion.

Water Resources Research, Vol. 46, W11555, 2010.

[2] Nordic HBV model – Karmen, release 3.14, March 20, 1995.

Nils Roar Sælthun, 1995

[3] A model for the spatial distribution of snow water equivalent parameterized from the spatial variability of precipitation. – The Cryosphere, September 2016.

T. Skaugen and I. H. Weltzien: Spatial distribution of SWE

[4] A rainfall-runoff model parameterized from GIS and runoff data.

Skaugen and Onof, 2014. Article in Hydrological Processes, July 2014

[5] Killingtveit and Sælthun, 1995; Sælthun, 1996.

[6] European Commission, Environment.

http://ec.europa.eu/environment/water/participation/map_mc/countries/norway_en.htm

[7] NVE Norges vassdrags- og energidirektorat

https://www.nve.no/
http://atlas.nve.no/html5Viewer/?viewer=nveatlas

[8] NEVINA Nedbørfelt-Vannføring-INdeks-Analyse

http://nevina.nve.no/

[9] SeNorge

http://www.senorge.no/

[10] NASA

https://disc.gsfc.nasa.gov/hydrology/data-holdings/parameters/snow_water_equivalent.shtml.

[11] Google Maps

https://www.darrinward.com/lat-long/?id=2951788

[12] Zonum Solutions

http://www.zonums.com/online/coords/cotrans.php?module=14

[13] Github - Statkraft

https://github.com/statkraft/shyft

[14] SHyFT Documentation. Release 0.1.

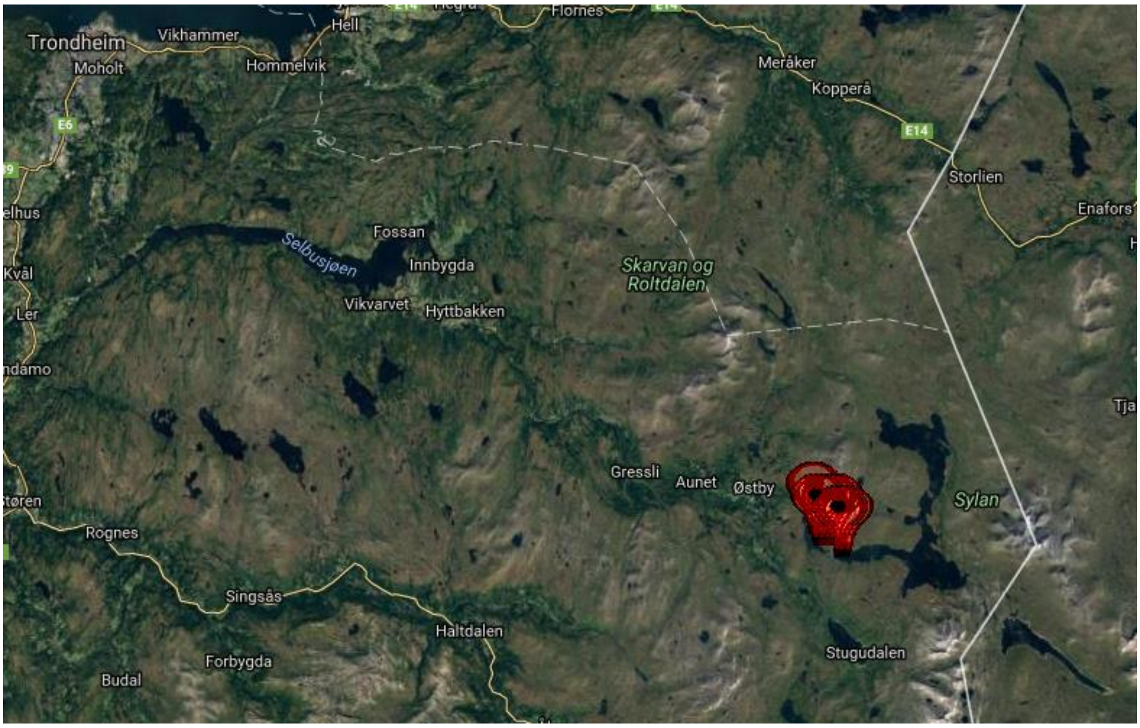
Sigbjørn Helset, Ola Skavhaug and others.

[15] S. U. Islam and S. J. Déry: Evaluating uncertainties in modelling snow hydrology. Hydrology and Earth System Sciences. 29 March 2017.

Appendix 1

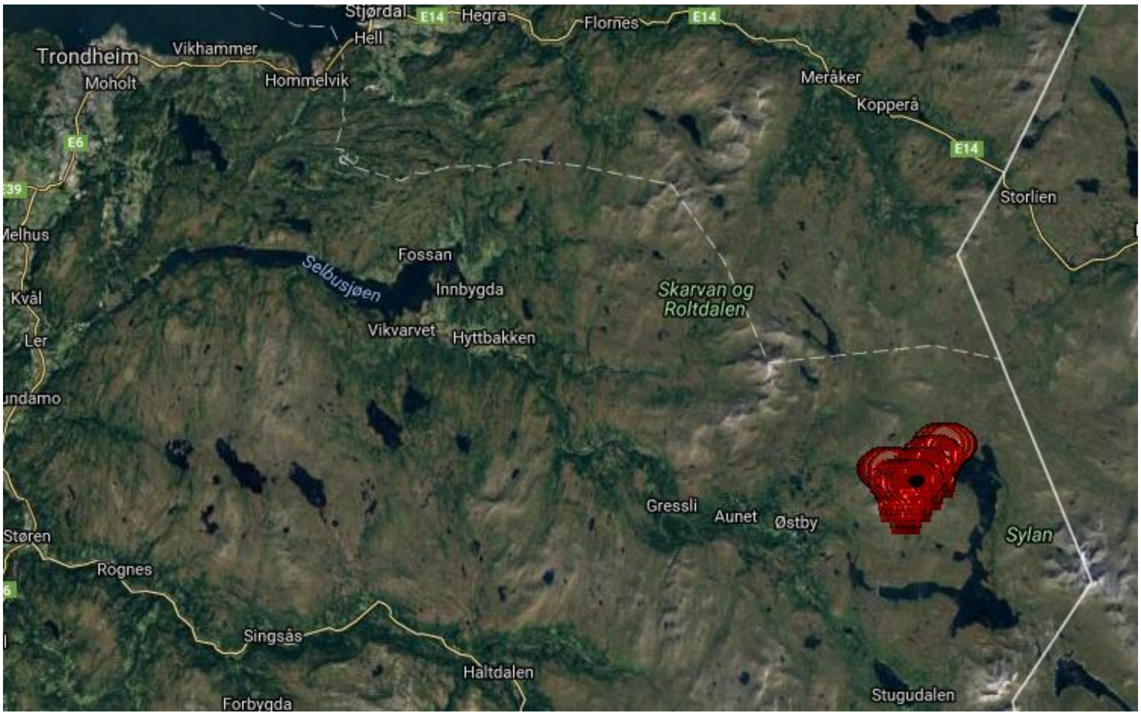
Snow transects

TRANSECT 1



TRANSECT 2


Nesjøen
Sverige
Norge
Østby
Sylan


Stjørdal
E14
Hegra
Flornes
E14
Hell
Trondheim
Vikhammer
Moholt
Hommelvik
Meråker
Kopperå
E6
E14
Storlien
Melhus
Fossan
Selbusjøen
Innbygda
Skarvan og
Roltdalen
Kvål
Vikvarvet
Hyttbakken
Ler
Gressli
Aunet
Østby
Sylan
Støren
Rognes
Singsås
Haltdalen
Forbygda
Stugudalen

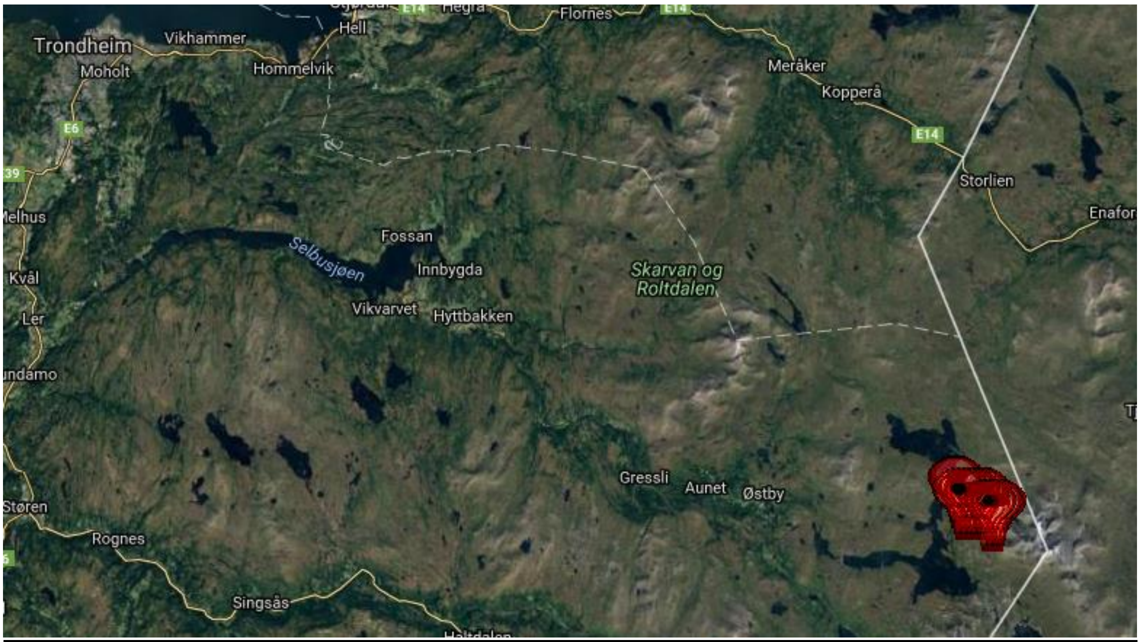
TRANSECT 3


TRONDELAG NORTE
SØR-TRØNDELAG
RONDELAG NORTE
SØR-TRØNDELAG
Sankkjølen
Storerikvollen

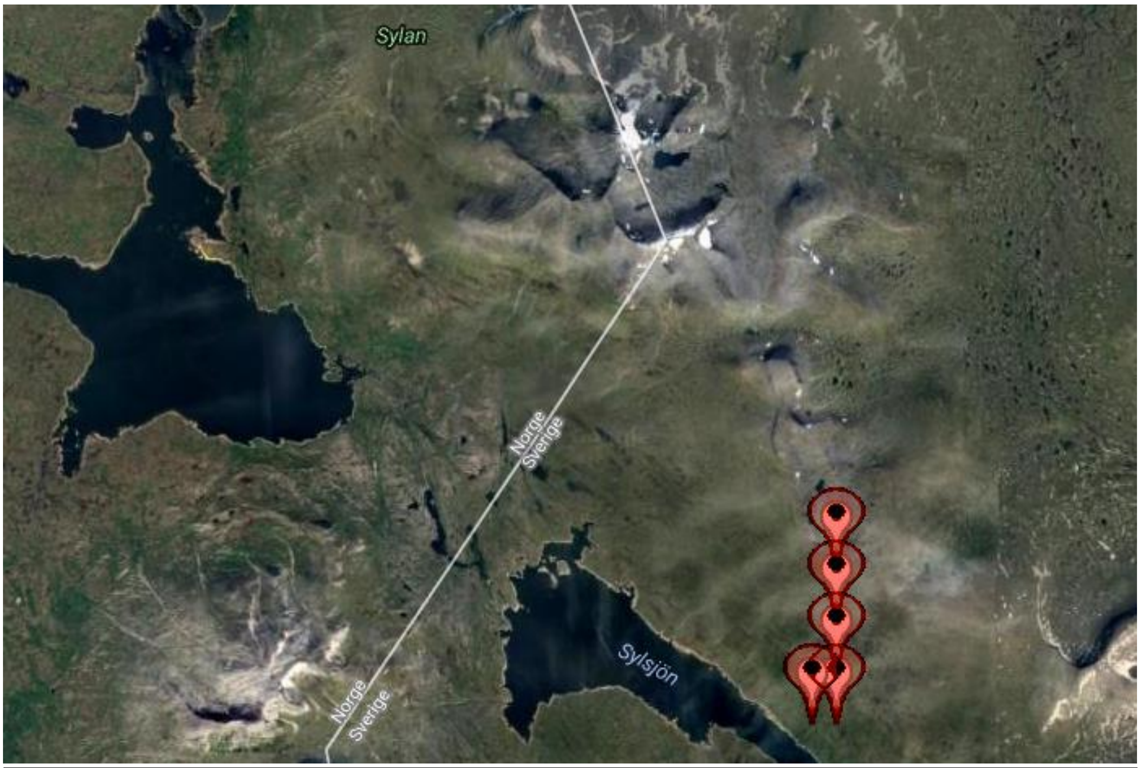

Hegra
Flornes
Hell
Trondheim
Vikhammer
Moholt
Hommelvik
Meråker
Kopperå
E6
E14
Storlien
Melhus
Fossan
Selbusjøen
Innbygda
Skarvan og Roltdalen
Kvål
Vikvarvet
Hyttbakken
Ler
Gressli
Aunet
Østby
Sylan
Støren
Rognes
Singsås

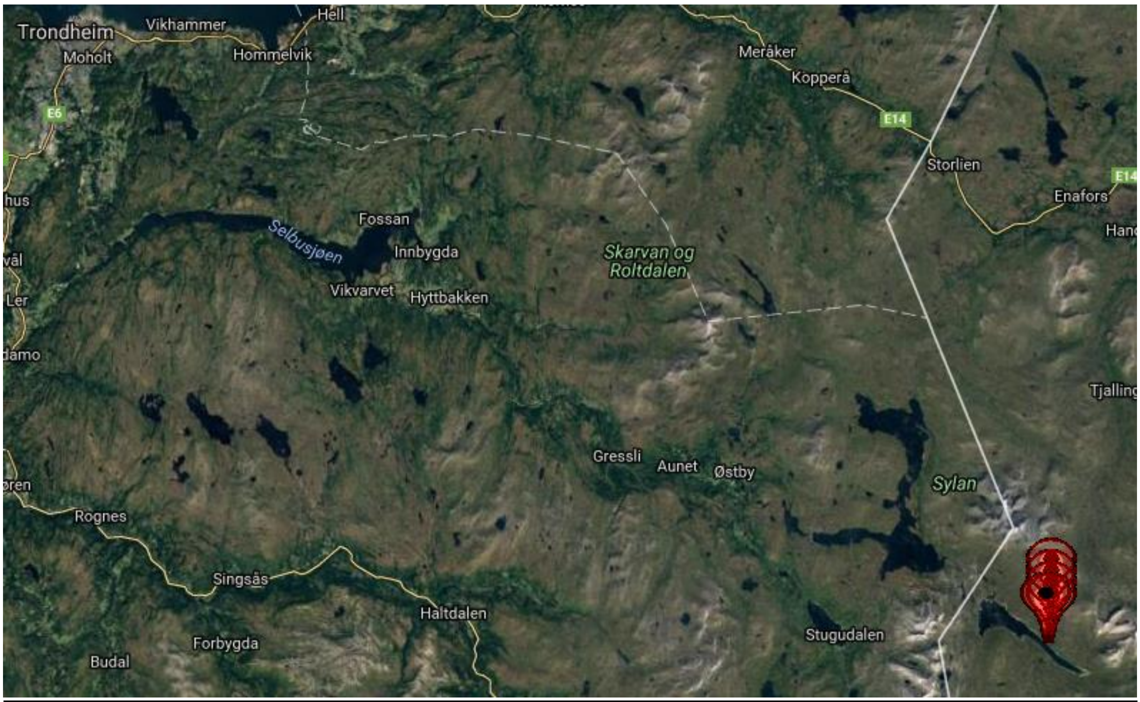
TRANSECT 4


Nesjøen
Sylan
Sverige
Norge
Norge
Sverige


Trondheim
Vikhammer
Moholt
Hommelvik
Hell
Hegra
E14
Flornes
E14
Meråker
Kopperå
E14
E6
Storlien
Enafors
Melhus
Fossan
Selbusjøen
Innbygda
Skarvan og Roltdalen
Kvål
Vikvarvet
Hyttbakken
Ler
Gressli
Aunet
Østby
Støren
Rognes
Singsås

TRANSECT 5


Sylan
Norge
Sverige
Norge
Sverige
Sylsjön


Trondheim
Vikhammer
Hell
Moholt
Hommelvik
E6
Meråker
Kopperå
E14
Storlien
E14
Enafors
Fossan
Selbusjøen
Innbygda
Vikvarvet
Hyttbakken
Skarvan og Roltdalen
Gressli
Aunet
Østby
Sylan
Rognes
Singsås
Haltdalen
Forbygda
Budal
Stugudalen

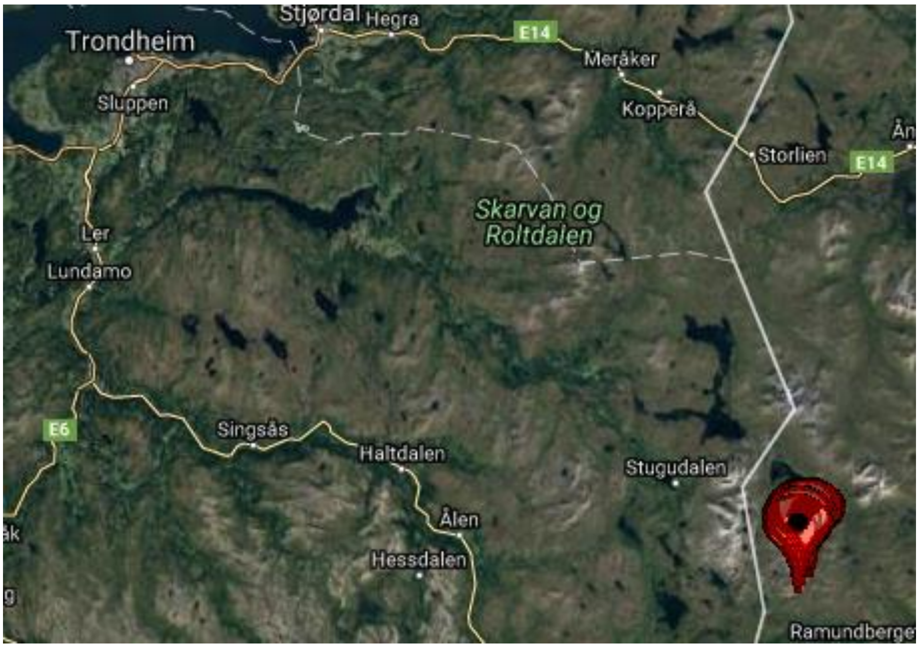
TRANSECT 6


Sylsjön
Norge
Sverige
Sverige
Norge


Trondheim
Hegra
E14
Meråker
Sluppen
Kopperå
Storlien
Ånn
E14
Skarvan og
Roltdalen
Ler
Lundamo
E6
Singsås
Haltdalen
Stugudalen
Ålen
Hessdalen

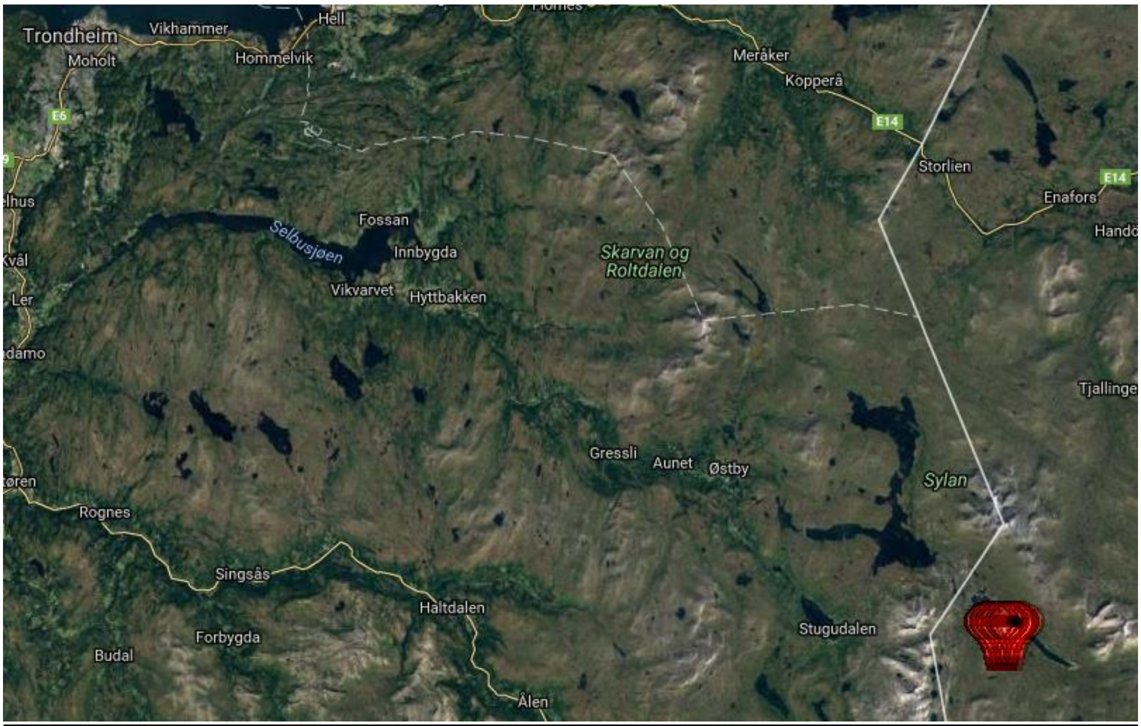
TRANSECT 7


Sverige
Norge

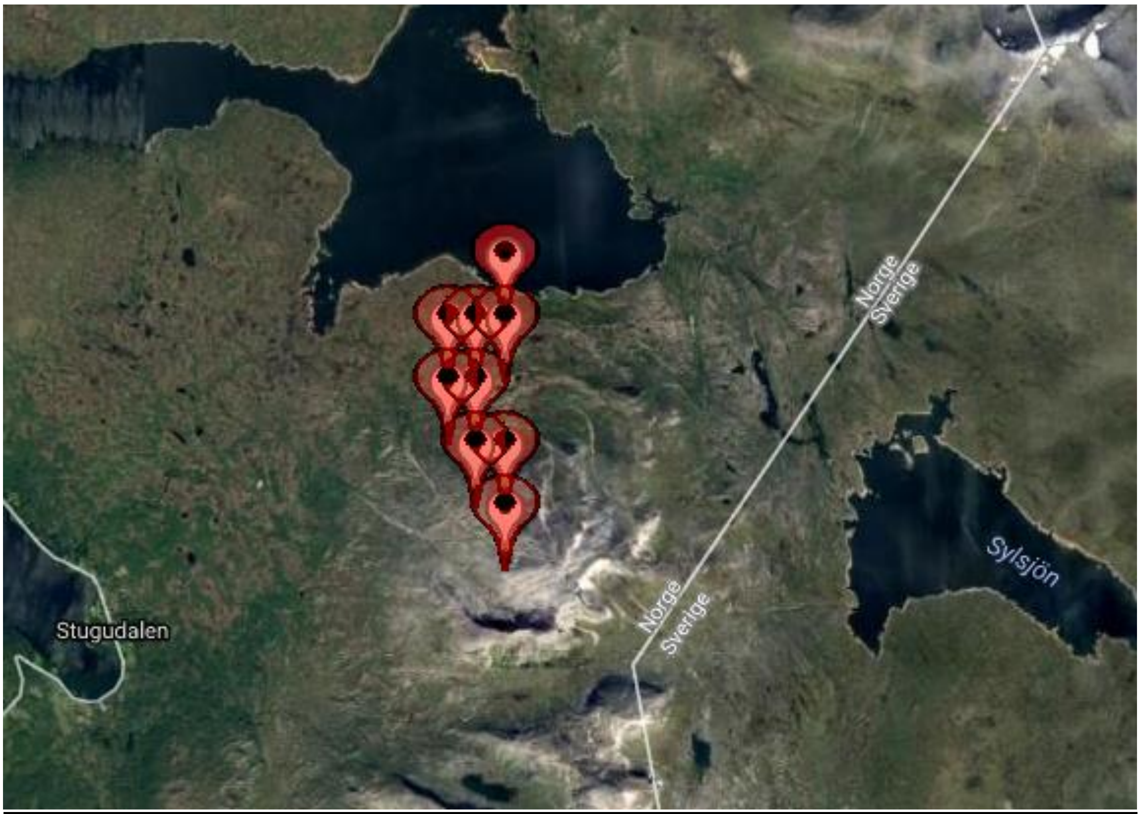

Stjørdal
Hegra
Trondheim
E14
Meråker
Sluppen
Kopperå
Storlien
E14
Skarvan og Roltdalen
Ler
Lundamo
E6
Singsås
Haltdalen
Stugudalen
Ålen
Hessdalen
Ramundberge

TRANSECT 8


Sylsjön
Norge
Sverige


Trondheim
Vikhammer
Hell
Moholt
Hommelvik
Meråker
Kopperå
E6
E14
Storlien
Enafors
Fossan
Selbusjøen
Innbygda
Skarvan og Roltdalen
Vikvarvet
Hyttbakken
Gressli
Aunet
Østby
Sylan
Rognes
Singsås
Haltdalen
Stugudalen
Forbygda
Budal
Ålen

TRANSECT 9


Norge
Sverige
Sylsjön
Stugudalen
Norge
Sverige


Trondheim
Vikhammer
Hell
Hegra
Flornes
E14
Moholt
Hommelvik
Meråker
Kopperå
E6
E14
Storlien
elhus
Fossan
Selbusjøen
Innbygda
Skarvan og
Roltdalen
Kvål
Vikvarvet
Hyttbakken
Ler
ndamo
Gressli
Aunet
Østby
Sylan
Støren
Rognes
Singsås
Haltdalen
Stugudalen
Forbygda
Budal

Appendix 2

SHyFT

A. Requirements

B. Installation procedure

C. Configuration files

a. neanidelva_region.yaml
b. neanidelva_datasets.yaml
c. neanidelva_interpolation.yaml
d. neanidelva_simulation.yaml
e. neanidelva_ptgsk
    e.i neanidelva_model.yaml
    e.ii neanidelva_calibration.yaml
f. neanidelva_pthsk
    f.i neanidelva_model.yaml
    f.ii neanidelva_calibration.yaml
g. neanidelva_ptssk
    g.i neanidelva_model.yaml
    g.ii neanidelva_calibration.yaml

D. Program files

a. Run_Shyft_Q.py
b. Run_Shyft_SCA
c. Run_Shyft_SWE
d. Calib_Shyft

A. Requirements

The following list of programs and packages were to be installed in order to use SHyFT in Windows:

a. GIT Bash and GIT cmd.
b. A Python3 (3.4 or higher) interpreter.
c. The SWIG wrapping tool (3.0.5 or higher).
d. NumPy
e. netcdf4
f. gdal
g. matplotlib
h. nose
i. shapely
j. pyproj

B. Installation procedure

The step-wise procedure to get SHyFT running successfully is the one that follows:

a. Find and download the ShyFT package at: https://github.com/statkraft/shyft
b. Install GIT from: https://git-scm.com
c. In the GIT window, using the comand "cd", change the directory to where ShyFT will be unpacked.
d. In order to copy the program files from ShyFT to the hard disk, type "git clone https://github.com/statkraft/shyft.git" in the GIT window.
e. In order to copy the data files from ShyFT to the hard disk, type "git clone https://github.com/statkraft/shyft-data.git" in the GIT window. Keep the same directory as for ShyFT.
f. Go to https://github.com/statkraft/shyft/releases and search for the latest release, which for this thesis was SK_2017_02_18. Download the zip files win_x64_common_bin and win_x64_shyft_shyft_api_np1104_py34.
g. Install Anaconda 3.4, SWIG and PyCharm.
h. Install the required packages such as NumPy, netcdf4, gdal, matplotlib, nose, shapely and pyproj. That can be done by typing in the commad line: conda install «name of the package».

Additionally, in order to avoid some common errors that may make SHyFT not work properly or not work at all, the following is highly recommended:

a. Open the command prompt.
b. Update the conda package: type "conda update conda".
c. Downgrade Python to version 3.4: type "conda install python=3.4".
d. Update/Downgrade the following packages: type "conda install netcdf4=1.2.4 nose=1.3.7 matplotlib=2.0.0 pyproj=1.9.5.1"
e. Install shapely 1.5.13: type "conda install –c scitools shapely=1.5.13".

C. Configuration files

a. neanidelva_region.yaml


```yaml
---
repository:
  class:
!!python/name:shyft.repository.netcdf.cf_region_model_repository
.CFRegionModelRepository
  params:
    data_file: netcdf/orchestration-testdata/cell_data.nc

domain:
  EPSG: 32633
  nx: 109
  ny: 80
  step_x: 1000
  step_y: 1000
  lower_left_x: 266000
  lower_left_y: 6960000

catchment_indices:
  - 1228
  - 1308
  - 1330
  - 1394
  - 1443
  - 1726
  - 1867
  - 1966
  - 1996
  - 2041
  - 2129
  - 2195
  - 2198
  - 2277
  - 2402
  - 2446
  - 2465
  - 2545
  - 2640
  - 2718
  - 2728
  - 3002
  - 3178
  - 3536
  - 3630
  - 1000010
  - 1000011
```

b. neanidelva_datasets.yaml


```yaml
---
sources:
  - repository:
!!python/name:shyft.repository.netcdf.arome_concat_data_reposito
ry.AromeConcatDataRepository
    types:
      - precipitation
      - wind_speed
      - temperature
      - relative_humidity
      - radiation
    params:
      filename: netcdf/orchestration-testdata/arome_merged_Nea-
Nidelv_Buffer5km_Chunck64x64x32_Complevel4_TimeUnlimited.nc
      nb_fc_to_drop: 0
      nb_fc_interval_to_concat: 1
...
```

c. neanidelva_interpolation


```yaml
interpolation_parameters:
  temperature:
    # method: btk
    # params:
      # temperature_gradient: -0.6
      # temperature_gradient_sd: 0.25
      # nug: 0.5
      # range: 200000.0
      # sill: 25.0
      # zscale: 20.0
    method: idw
    params:
      max_distance: 3000.0
      max_members: 5
      distance_measure_factor: 1.0
      default_temp_gradient: -0.005 # degC/m, so -0.5 degC/100m
      gradient_by_equation: false
  precipitation:
    method: idw
    params:
      max_distance: 3000.0
      max_members: 5
      distance_measure_factor: 1
      scale_factor: 1.02
  radiation:
    method: idw
    params:
      max_distance: 3000.0
      max_members: 5
      distance_measure_factor: 1.0
  wind_speed:
    method: idw
    params:
      max_distance: 3000.0
      max_members: 5
      distance_measure_factor: 1.0
  relative_humidity:
    method: idw
    params:
      max_distance: 3000.0
      max_members: 5
      distance_measure_factor: 1.0
```

d. neanidelva_simulation.yaml


```yaml
---
neanidelva:
  region_config_file: neanidelva_region.yaml
  model_config_file: neanidelva_model.yaml
  datasets_config_file: neanidelva_datasets.yaml
  interpolation_config_file: neanidelva_interpolation.yaml
  start_datetime: 2012-09-01T00:00:00
  run_time_step: 86400 # 1 day time step in seconds
  number_of_steps: 1095   # 3 years in days
  region_model_id: 'neanidelva-ptgsk' # Gamma/hbv/Skaugen
  #interpolation_id: 2   # this is optional (default 0)
  initial_state:
    repository:
      class:
!!python/name:shyft.repository.generated_state_repository.Genera
tedStateRepository
      params:
        model: !!python/name:shyft.api.pt_gs_k.PTGSKModel #
Gamma/hbv/Skaugen
    tags: []
  references:
  - repository:
!!python/name:shyft.repository.netcdf.cf_ts_repository.CFTsRepos
itory
    params:
      file: netcdf/orchestration-testdata/discharge.nc
      var_type: discharge
    1D_timeseries:
    - catch_id: [1308,1394,1867,2198,2402,2545]
      type: discharge
      uid: /TEV.-Tya...........-D9100A3B1060R123.999
      run_time_step: 86400 # 3600
    - catch_id:
[1228,1443,1726,2041,2129,2195,2277,2465,2718,3002,3630]
      type: discharge
      uid: /TEV.-Selbu-lok.....-D9100A3B1070R123.020
      run_time_step: 86400 # 3600
    - catch_id: [1996,2446,2640,3536]
      type: discharge
      uid: /TEV.-Nea...........-D9100A3B1050R123.998
      run_time_step: 86400 # 3600
...
```

e. neanidelva_ptgsk

f.i neanidelva_model.yaml


```yaml
# This file has been automatically generated after a calibration
run
model_parameters:
  gamma_snow:
    calculate_iso_pot_energy: false
    fast_albedo_decay_rate: 6.0
    glacier_albedo: 0.4
    initial_bare_ground_fraction: 0.04
    max_albedo: 0.9
    max_water: 0.1
    min_albedo: 0.25
    slow_albedo_decay_rate: 30.0
    snow_cv: 0.4
    snow_cv_altitude_factor: 0.0
    snow_cv_forest_factor: 0.0
    snowfall_reset_depth: 5.0
    surface_magnitude: 30.0
    tx: -0.5
    wind_const: 3.5
    wind_scale: 0.7720234294504736
    winter_end_day_of_year: 100
  glacier_melt:
    dtf: 6.0
  hbv_actual_evapotranspiration:
    ae_scale_factor: 1.5
  kirchner:
    c1: -3.9416530465560666
    c2: 0.5492115556225241
    c3: -1.181
  precipitation_correction:
    scale_factor: 0.7000039387570476
  priestley_taylor:
    albedo: 0.2
    alpha: 1.26
  routing:
    alpha: 0.9
    beta: 3.0
    velocity: 0.0
model_t: !!python/name:shyft.api.pt_gs_k._pt_gs_k.PTGSKModel
```

e. neanidelva_ptgsk

f.ii neanidelva_calibration.yaml


```yaml
neanidelva:
  model_config_file: neanidelva_simulation.yaml
  calibrated_model_file: calibrated_model.yaml  # file where the
calibrated params will go
  optimization_method:
    name: min_bobyqa # can be 'min_bobyqa', 'dream' or 'sceua'
    params:
      max_n_evaluations: 1500
      tr_start: 0.1
      tr_stop: 1.0e-5
  target:
  - repository:
!!python/name:shyft.repository.netcdf.cf_ts_repository.CFTsRepos
itory
    params:
      file: netcdf/orchestration-testdata/discharge.nc
      var_type: discharge
    1D_timeseries:
    - catch_id: [1308,1394,1867,2198,2402,2545]
      uid: /TEV.-Tya...........-D9100A3B1060R123.999
      start_datetime: 2013-09-01T00:00:00
      run_time_step: 86400 # 3600
      number_of_steps: 365 # 26280
      weight: 1.0
      obj_func:
        name: NSE # Nash-Sutcliffe efficiency (NSE) or Kling-
Gupta efficiency (KGE)
        scaling_factors:
          s_corr: 1.0
          s_var: 1.0
          s_bias: 1.0
    - catch_id:
[1228,1443,1726,2041,2129,2195,2277,2465,2718,3002,3630]
      uid: /TEV.-Selbu-lok.....-D9100A3B1070R123.020
      start_datetime: 2013-09-01T00:00:00
      run_time_step: 86400 # 3600
      number_of_steps: 365 # 26280
      weight: 1.0
      obj_func:
        name: NSE # Nash-Sutcliffe efficiency (NSE) or Kling-
Gupta efficiency (KGE)
        scaling_factors:
          s_corr: 1.0
          s_var: 1.0
          s_bias: 1.0
```

```
  - catch_id: [1996,2446,2640,3536]
    uid: /TEV.-Nea..........-D9100A3B1050R123.998
    start_datetime: 2013-09-01T00:00:00
    run_time_step: 86400 # 3600
    number_of_steps: 365 # 26280
    weight: 1.0
    obj_func:
      name: NSE # Nash-Sutcliffe efficiency (NSE) or Kling-Gupta efficiency (KGE)
      scaling_factors:
        s_corr: 1.0
        s_var: 1.0
        s_bias: 1.0
overrides:
  model:
    model_t: !!python/name:shyft.api.pt_gs_k.PTGSKOptModel
calibration_parameters:
  kirchner.c1:
    min: -6.0 # -3.0
    max: -2.0 # 2.0
  kirchner.c2:
    min: 0.1 # 0.8
    max: 1.0 # 1.2
  kirchner.c3:
    min: -0.15
    max: 0.15
  ae.ae_scale_factor:
    min: 1.5
    max: 1.5
  gs.tx:
    min: -3.0
    max: 2.0
  gs.wind_scale:
    min: 0.0
    max: 2.0
  gs.max_water:
    min: 0.1
    max: 0.1
  gs.wind_const:
    min: 1.0
    max: 6.0
  gs.fast_albedo_decay_rate:
    min: 2.0 # 5.0
    max: 10.0 # 15.0
  gs.slow_albedo_decay_rate:
    min: 20.0 # 20.0
    max: 40.0 # 40.0
  gs.surface_magnitude:
```

```
    min: 30.0
    max: 30.0
  gs.max_albedo:
    min: 0.9
    max: 0.9
  gs.min_albedo:
    min: 0.1
    max: 0.4
  gs.snowfall_reset_depth:
    min: 5.0
    max: 5.0
  gs.snow_cv:
    min: 0.4
    max: 0.4
  gs.snow_cv_forest_factor:
    min: 0.0
    max: 0.0
  gs.snow_cv_altitude_factor:
    min: 0.0
    max: 0.0
  gs.glacier_albedo:
    min: 0.4
    max: 0.4
  p_corr.scale_factor:
    min: 0.4
    max: 1.0
  pt.albedo:
    min: 0.2
    max: 0.2
  pt.alpha:
    min: 1.26
    max: 1.26
  gs.initial_bare_ground_fraction:
    min: 0.04
    max: 0.04
  gs.winter_end_day_of_year:
    min: 100
    max: 100
  gs.calculate_iso_pot_energy:
    min: 0
    max: 0
  gm.dtf:
    min: 6.0
    max: 6.0
  routing.velocity:
    min: 0.0
    max: 0.0
  routing.alpha:
```

```yaml
  min: 0.9
  max: 0.9
routing.beta:
  min: 3.0
  max: 3.0
```

f. neanidelva_pthsk

f.i neanidelva_model.yaml


```yaml
# This file has been automatically generated after a calibration
run
model_parameters:
  glacier_melt:
    dtf: 6.0
  hbv_actual_evapotranspiration:
    ae_scale_factor: 0.05
  hbv_snow:
    cfr: 0.0004277970782749167
    cx: 0.4312917287934755
    lw: 0. 14517552601229675
    ts: -0.3561834730342437
    tx: 0.05778703071180713
  kirchner:
    c1: -4.092099354524807
    c2: 0.5784737327123776
    c3: -0.011268379976200199
  precipitation_correction:
    scale_factor: 1.0
  priestley_taylor:
    albedo: 0.2
    alpha: 1.26
  routing:
    alpha: 0.9
    beta: 3.0
    velocity: 0.0
model_t: !!python/name:shyft.api.pt_hs_k._pt_hs_k.PTHSKModel
```

f. neanidelva_pthsk

f.ii neanidelva_calibration.yaml


```yaml
neanidelva:
  model_config_file: neanidelva_simulation.yaml
  calibrated_model_file: calibrated_model.yaml  # file where the calibrated params will go
  optimization_method:
    name: min_bobyqa # can be 'min_bobyqa', 'dream' or 'sceua'
    params:
      max_n_evaluations: 1500
      tr_start: 0.1
      tr_stop: 1.0e-5
  target:
  - repository: !!python/name:shyft.repository.netcdf.cf_ts_repository.CFTsRepository
    params:
      file: netcdf/orchestration-testdata/discharge.nc
      var_type: discharge
    1D_timeseries:
    - catch_id: [1308,1394,1867,2198,2402,2545]
      uid: /TEV.-Tya...........-D9100A3B1060R123.999
      start_datetime: 2013-09-01T00:00:00
      run_time_step: 86400 # 3600
      number_of_steps: 730 # 26280
      weight: 1.0
      obj_func:
        name: NSE # Nash-Sutcliffe efficiency (NSE) or Kling-Gupta efficiency (KGE)
        scaling_factors:
          s_corr: 1.0
          s_var: 1.0
          s_bias: 1.0
    - catch_id: [1228,1443,1726,2041,2129,2195,2277,2465,2718,3002,3630]
      uid: /TEV.-Selbu-lok.....-D9100A3B1070R123.020
      start_datetime: 2013-09-01T00:00:00
      run_time_step: 86400 # 3600
      number_of_steps: 730 # 26280
      weight: 1.0
      obj_func:
        name: NSE # Nash-Sutcliffe efficiency (NSE) or Kling-Gupta efficiency (KGE)
        scaling_factors:
          s_corr: 1.0
          s_var: 1.0
          s_bias: 1.0
```

```yaml
    - catch_id: [1996,2446,2640,3536]
      uid: /TEV.-Nea..........-D9100A3B1050R123.998
      start_datetime: 2013-09-01T00:00:00
      run_time_step: 86400 # 3600
      number_of_steps: 730 # 26280
      weight: 1.0
      obj_func:
        name: NSE # Nash-Sutcliffe efficiency (NSE) or Kling-
Gupta efficiency (KGE)
        scaling_factors:
          s_corr: 1.0
          s_var: 1.0
          s_bias: 1.0
  overrides:
    model:
      model_t: !!python/name:shyft.api.pt_hs_k.PTHSKOptModel
  calibration_parameters:
    kirchner.c1:
      min: -6.0
      max: -2.0
    kirchner.c2:
      min: 0.3
      max: 1.2
    kirchner.c3:
      min: -0.1
      max: 0.0
    ae.ae_scale_factor:
      min: 0.05
      max: 0.05
    hs.lw:
      min: 0.10
      max: 0.20
    hs.tx:
      min: -0.5
      max: 0.8
    hs.cx:
      min: 0.0
      max: 0.6
    hs.ts:
      min: -1.0
      max: 0.5
    hs.cfr:
      min: 0.00005
      max: 0.001
    p_corr.scale_factor:
      min: 1.0
      max: 1.0
    pt.albedo:
```

```yaml
    min: 0.2
    max: 0.2
  pt.alpha:
    min: 1.26
    max: 1.26
  gm.dtf:
    min: 6.0
    max: 6.0
  routing.velocity:
    min: 0.0
    max: 0.0
  routing.alpha:
    min: 0.9
    max: 0.9
  routing.beta:
    min: 3.0
    max: 3.0
```

g. neanidelva_ptssk

g.i. neanidelva_model.yaml


```yaml
# This file has been automatically generated after a calibration
run
model_parameters:
  glacier_melt:
    dtf: 6.0
  hbv_actual_evapotranspiration:
    ae_scale_factor: 1.5
  kirchner:
    c1: -3.9193472117560217
    c2: 0.5287373657677946
    c3: -0.019789614977839445
  precipitation_correction:
    scale_factor: 0.7652132602052532
  priestley_taylor:
    albedo: 0.2
    alpha: 1.26
  routing:
    alpha: 0.9
    beta: 3.0
    velocity: 0.0
  skaugen_snow:
    alpha_0: 40.558490601603964
    cfr: 0.009805303599122295
    cx: 0.5857483219073707
    d_range: 110.71827615528363
    max_water_fraction: 0.34530499856862423
    ts: 0.13775258235272983
    tx: -0.004269554524006347
    unit_size: 0.18585660728780456
model_t: !!python/name:shyft.api.pt_ss_k._pt_ss_k.PTSSKModel
```

g. neanidelva_ptssk

g.ii. neanidelva_calibration.yaml


```yaml
neanidelva:
  model_config_file: neanidelva_simulation.yaml
  calibrated_model_file: calibrated_model.yaml  # file where the calibrated params will go
  optimization_method:
    name: min_bobyqa # can be 'min_bobyqa', 'dream' or 'sceua'
    params:
      max_n_evaluations: 1500
      tr_start: 0.1
      tr_stop: 1.0e-5
  target:
  - repository:
!!python/name:shyft.repository.netcdf.cf_ts_repository.CFTsRepository
    params:
      file: netcdf/orchestration-testdata/discharge.nc
      var_type: discharge
    1D_timeseries:
    - catch_id: [1308,1394,1867,2198,2402,2545]
      uid: /TEV.-Tya...........-D9100A3B1060R123.999
      start_datetime: 2013-09-01T00:00:00
      run_time_step: 86400 # 3600
      number_of_steps: 730 # 26280
      weight: 1.0
      obj_func:
        name: NSE # Nash-Sutcliffe efficiency (NSE) or Kling-Gupta efficiency (KGE)
        scaling_factors:
          s_corr: 1.0
          s_var: 1.0
          s_bias: 1.0
    - catch_id:
[1228,1443,1726,2041,2129,2195,2277,2465,2718,3002,3630]
      uid: /TEV.-Selbu-lok.....-D9100A3B1070R123.020
      start_datetime: 2013-09-01T00:00:00
      run_time_step: 86400 # 3600
      number_of_steps: 730 # 26280
      weight: 1.0
      obj_func:
        name: NSE # Nash-Sutcliffe efficiency (NSE) or Kling-Gupta efficiency (KGE)
        scaling_factors:
          s_corr: 1.0
          s_var: 1.0
          s_bias: 1.0
```

```yaml
    - catch_id: [1996,2446,2640,3536]
      uid: /TEV.-Nea..........-D9100A3B1050R123.998
      start_datetime: 2013-09-01T00:00:00
      run_time_step: 86400 # 86400
      number_of_steps: 730 # 365
      weight: 1.0
      obj_func:
        name: NSE # Nash-Sutcliffe efficiency (NSE) or Kling-Gupta efficiency (KGE)
        scaling_factors:
          s_corr: 1.0
          s_var: 1.0
          s_bias: 1.0
  overrides:
    model:
      model_t: !!python/name:shyft.api.pt_ss_k.PTSSKOptModel
  calibration_parameters:
    kirchner.c1:
      min: -6.0 # -3.0
      max: -2.0 # 2.0
    kirchner.c2:
      min: 0.1 # 0.8
      max: 1.0 # 1.2
    kirchner.c3:
      min: -0.05
      max: 0.0
    ae.ae_scale_factor:
      min: 1.5
      max: 1.5
    ss.alpha_0:
      min: 25
      max: 55
    ss.d_range:
      min: 95
      max: 125
    ss.unit_size:
      min: 0.0
      max: 1.0
    ss.max_water_fraction:
      min: 0.0
      max: 1.0
    ss.tx:
      min: -0.5
      max: 0.5
    ss.cx:
      min: -0.5
      max: 2.0
    ss.ts:
```

```yaml
    min: -0.1
    max: 0.6
  ss.cfr:
    min: 0.0
    max: 0.3
  p_corr.scale_factor:
    min: 0.5
    max: 1.0
  pt.albedo:
    min: 0.2
    max: 0.2
  pt.alpha:
    min: 1.26
    max: 1.26
  gm.dtf:
    min: 6.0
    max: 6.0
  routing.velocity:
    min: 0.0
    max: 0.0
  routing.alpha:
    min: 0.9
    max: 0.9
  routing.beta:
    min: 3.0
    max: 3.0
```

D. Program files

a. Run_Shyft_Q.py

```
# from os import path

# from shyft import shyftdata_dir
from shyft import api
from shyft.repository.default_state_repository import
DefaultStateRepository
from shyft.orchestration.configuration import yaml_configs
from shyft.orchestration.simulators.config_simulator import
ConfigSimulator

config_dir =
'C:\\thesis\\shyft\\shyft\\tests\\netcdf\\neanidelva_ptssk\\nean
idelva_simulation.yaml'  # Type the config file
cfg = yaml_configs.YAMLSimConfig(config_dir, "neanidelva")

# get a simulator
simulator = ConfigSimulator(cfg)

n_cells = simulator.region_model.size()
state_repos =
DefaultStateRepository(simulator.region_model.__class__,
n_cells)
simulator.run(cfg.time_axis, state_repos.get_state(0))

# access a particular cell
cells = simulator.region_model.cells
cell = cells[330]

# access the discharge of that cell as a time series
ts_q = cell.rc.avg_discharge
# get the values
q = ts_q.v.to_numpy()

# plotting
from datetime import datetime
times = [datetime.utcfromtimestamp(ts_q.time(i)) for i in
range(ts_q.size())]
import matplotlib.pyplot as plt
fig,ax = plt.subplots()
plt.ylabel("Simulated discharge [m$^3$/s]", fontsize=12)
plt.suptitle("Discharge, cell#330")
ax.plot(times,q)
plt.show()

#exporting swe values to excel
```

```
#import libraries
import openpyxl
# create workbook object
workbook = openpyxl.Workbook()
sheet = workbook.get_active_sheet()
sheet.tittle = 'Sheet #'
# loop to set the value of each cell
for i in range(0, len(q)):
    sheet.cell(row = i+1, column = 2).value = q[i]
# save the file and give it a name # specify directory and name
workbook.save('C:\\Users\\USUARIO\\Dropbox\\NTNU\\Project + MSc
Thesis\\THESIS\\shyft
prints\\ptssk\\runoff\\Q_values_cell#2745_1year.xlsx')
```

b. Run_Shyft_SCA.py

```
# from os import path

# from shyft import shyftdata_dir
from shyft import api
from shyft.repository.default_state_repository import
DefaultStateRepository
from shyft.orchestration.configuration import yaml_configs
from shyft.orchestration.simulators.config_simulator import
ConfigSimulator

config_dir =
'C:\\thesis\\shyft\\shyft\\tests\\netcdf\\neanidelva_ptgsk\\nean
idelva_simulation.yaml'  # Type the config file
cfg = yaml_configs.YAMLSimConfig(config_dir, "neanidelva")

# get a simulator
simulator = ConfigSimulator(cfg)

n_cells = simulator.region_model.size()
state_repos =
DefaultStateRepository(simulator.region_model.__class__,
n_cells)
simulator.run(cfg.time_axis, state_repos.get_state(0))

from matplotlib import pylab as plt

# extract SCA
import numpy as np
# first, set a date: year, month, day, (hour of day if hourly
time step)
oslo = api.Calendar('Europe/Oslo')  # specifying input calendar
in Oslo tz-id
time_x = oslo.time(2014, 2, 1)  # the oslo calendar(incl dst)
converts calendar coordinates Y,M,D.. to its utc-time

# we need to get the index of the time_axis for the time
try:
    idx = simulator.region_model.time_axis.index_of(time_x)  #
index of time x on time-axis
except:
    print("Date out of range, setting index to 0")
    idx = 0

# fetching SCA (the response variable is named "snow_sca")
# You can use tab-completion to explore the `rc`, short
# for "response collector"
```

```
# object of the cell, to see further response variables
available.
# specifying empty list [] indicates all catchments, otherwise
pass catchment_id

sca = simulator.region_model.gamma_snow_response.sca([], idx)  #
specify snow routine and catch_ID

# for attr in dir(simulator.region_model):
#     if attr[0] is not '_': #ignore privates
#         print(attr)
# # and don't forget:
# help(simulator.region_model.gamma_snow_state)

cells = simulator.region_model.get_cells()

# Once we have the cells, we can get their coordinate
information
# and fetch the x- and y-location of the cells
x = np.array([cell.geo.mid_point().x for cell in cells])
y = np.array([cell.geo.mid_point().y for cell in cells])

# We can make a simple scatter plot again for quick
visualization
fig, ax = plt.subplots(figsize=(15, 5))
cm = plt.cm.get_cmap('winter')
plot = ax.scatter(x, y, c=sca,
                  vmin=0, vmax=1,
                  marker='s', s=40, lw=0,
                  cmap=cm)
print("print SCA")
plt.colorbar(plot)
plt.title('Snow Covered Area of {0} on
{1}'.format(cfg.region_model_id, oslo.to_string(time_x)))
plt.show()

# look at the catchment-wide average:
nea_avg_sca = np.average(sca)
print("Average SCA for Nea Nidelva: {0}".format(nea_avg_sca))

# And let's compute histogram of the snow covered area as well
fig, ax = plt.subplots()
ax.hist(sca, bins=20, range=(0, 1), color='y', alpha=0.5)
ax.set_xlabel("SCA of grid cell")
ax.set_ylabel("frequency")
print("print histogram")
plt.show()
print("ok6")
```

c. Run_Shyft_SWE.py

```
# from os import path

# from shyft import shyftdata_dir
from shyft import api
from shyft.repository.default_state_repository import
DefaultStateRepository
from shyft.orchestration.configuration import yaml_configs
from shyft.orchestration.simulators.config_simulator import
ConfigSimulator

config_dir =
'C:\\thesis\\shyft\\shyft\\tests\\netcdf\\neanidelva_ptgsk\\neanidel
va_simulation.yaml'  # Type the config file
cfg = yaml_configs.YAMLSimConfig(config_dir, "neanidelva")

# ------------------------- cell level --------------------------

# get a simulator
simulator = ConfigSimulator(cfg)

# activate collection (saving) of ALL states since it is deactivated
by default
simulator.region_model.set_state_collection(-1, True) # -1 means for
all catchments

n_cells = simulator.region_model.size()
state_repos =
DefaultStateRepository(simulator.region_model.__class__, n_cells)
simulator.run(cfg.time_axis, state_repos.get_state(0))

# access a particular cell
cells = simulator.region_model.cells
cell = cells[330]

# access the swe of that cell as a time series
ts_swe = cell.rc.snow_swe
# get the values
swe = ts_swe.v.to_numpy()

# plotting
from datetime import datetime
times = [datetime.utcfromtimestamp(ts_swe.time(i)) for i in
range(ts_swe.size())]
import matplotlib.pyplot as plt
fig,ax = plt.subplots()
plt.ylabel("Snow Water Equivalent [mm]", fontsize=12)
plt.suptitle("Snow Water Equivalent, cell#330")
```

```
ax.plot(times,swe)
plt.show()

#exporting swe values to excel

#import libraries
import openpyxl
# create workbook object
workbook = openpyxl.Workbook()
sheet = workbook.get_active_sheet()
sheet.tittle = 'Sheet #'
# loop to set the value of each cell
for i in range(0, len(swe)):
    sheet.cell(row = i+1, column = 2).value = swe[i]
# save the file and give it a name # specify directory and name
workbook.save('C:\\Users\\USUARIO\\Dropbox\\NTNU\\Project + MSc
Thesis\\THESIS\\shyft prints\\ptssk\\transect
9\\SWE_values_cell#211.xlsx')

# -------------------- subcatchment level ------------------

# # get a simulator
# simulator = ConfigSimulator(cfg)
#
# # activate collection (saving) of swe and sca since it is
deactivated by default
# # This is usually used to activate calibration using snow data
# # simulator.region_model.set_snow_sca_swe_collection(-1, True) # -
1 means for all catchments
# # activate collection (saving) of ALL states since it is
deactivated by default
# simulator.region_model.set_state_collection(-1, True) # -1 means
for all catchments
# n_cells = simulator.region_model.size()
# state_repos =
DefaultStateRepository(simulator.region_model.__class__, n_cells)
# simulator.run(cfg.time_axis, state_repos.get_state(0))
#
# # SWE and SCA are responses in ptgsk
# # to extract swe from ptgsk as a numpy array
# # swe =
simulator.region_model.gamma_snow_response.swe([]).v.to_numpy()
#
# # SWE and SCA are states in ptssk and pthsk
# # to extract swe from ptssk as a numpy array
# # swe =
simulator.region_model.skaugen_snow_state.swe([]).v.to_numpy()
# # to extract swe from pthsk as a numpy array
# # swe = simulator.region_model.hbv_snow_state.swe([]).v.to_numpy()
#
```

```
# ts_swe = simulator.region_model.skaugen_snow_state.swe([3536]) #
specify snow routine and state/response !! and subcatchment
# swe = ts_swe.v.to_numpy()
#
# # plotting
# from datetime import datetime
# times = [datetime.utcfromtimestamp(ts_swe.time(i)) for i in
range(ts_swe.size())]
# import matplotlib.pyplot as plt
# fig,ax = plt.subplots()
# plt.ylabel("Snow Water Equivalent [mm]", fontsize=12)
# plt.suptitle("Snow Water Equivalent, catch_ID#1996")
# ax.plot(times,swe)
```

d. Calib_Shyft.py

```
# importing thrid-party python modules
import sys
import os
# setting the path to the shyft build
shyft_path = os.path.abspath("C:\\thesis\\shyft")
sys.path.insert(0,shyft_path)

# importing the shyft modules needed for running a calibration
from shyft.repository.default_state_repository import
DefaultStateRepository
from shyft.orchestration.configuration.yaml_configs import
YAMLCalibConfig
from shyft.orchestration.simulators.config_simulator import
ConfigCalibrator

# set up configuration using *.yaml configuration files
config_file_path =
os.path.abspath("C:\\thesis\\shyft\\shyft\\tests\\netcdf\\neanid
elva_ptgsk\\neanidelva_calibration.yaml") # here is the *.yaml
file
cfg = YAMLCalibConfig(config_file_path, "neanidelva")
# initialize an instance of the orchestration's
ConfigCalcalibrator class, which has all the functionality
needed
# to run a calibration using the above initiated configuration
calib = ConfigCalibrator(cfg)
n_cells = calib.region_model.size()
state_repos =
DefaultStateRepository(calib.region_model.__class__, n_cells)
#calib.init()
# once the calibrator is set up, all you need to do is running
the calibration...
# the calibrated parameters are stored in a model.yaml.
results = calib.calibrate(cfg.sim_config.time_axis,
state_repos.get_state(0),
                          cfg.optimization_method['name'],
                          cfg.optimization_method['params'])
# Get NSE of calibrated run:
result_params = []
for i in range(results.size()):
    result_params.append(results.get(i))
print("Final NSE =", 1-
calib.optimizer.calculate_goal_function(result_params))
# Check out the calibrated parameters.

diff = 1.0E-3
print("{0:30s} {1:10s}".format("PARAM-NAME", "CALIB-VALUE"))
```

```
for i in range(results.size()):
    print("{0:30s} {1:10f}".format(results.get_name(i),
results.get(i)))
```