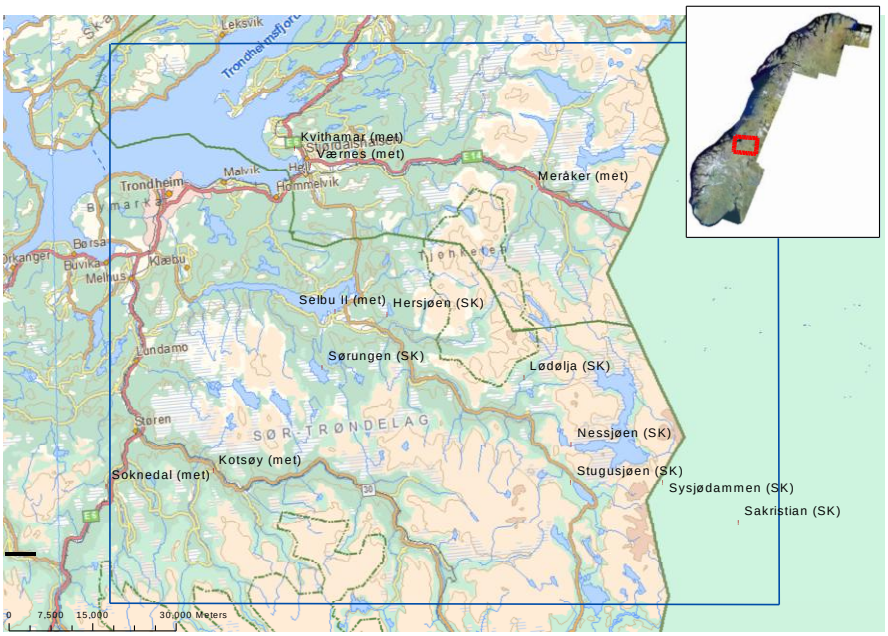
Master Thesis, Department of Geosciences

Backtracking maximum snow distribution from remotely sensed snow coverage

Rune Yoshida TOLLEFSRUD



UNIVERSITY OF OSLO

FACULTY OF MATHEMATICS AND NATURAL SCIENCES

Backtracking maximum snow distribution from remotely sensed snow coverage

Rune Yoshida TOLLEFSRUD

Master Thesis in Geosciences

Discipline: Geomatics

Department of Geosciences

Faculty of Mathematics and Natural Sciences

University of Oslo

30/11/2011

Abstract

In this study a physically-based, distributed snow model (SnowModel) was used to estimate seasonal maximum snow water equivalent (SWE) based on dates of snow cover disappearance. The dates of snow cover disappearance were derived from automatically recorded snow measurements at a local scale and from sequences of remotely sensed snow covered area (SCA) at a catchment scale. The study area comprises the catchment of Nea-Nidelv in Sør- and Nord-Trønderlag. The objective was to assess the accuracy of a backtracking routine using SnowModel, both at point and catchment scale. In addition it was tested whether the reconstructed snow distribution could be used to correct the modeled precipitation distribution, forcing the model results through the maximum SWE value. At point scale, three snow pillows and snow depth sensors were used to determine the date of snow cover disappearance, and validate the back-calculated value. Where snow pillow data registered reasonable values, the discrepancy of the backtracked SWE was between 3 and 19 mm SWE. It is evident that SnowModel does not reproduce mild periods too well, therefore overestimating most snow accumulation in late autumn/early winter, thus giving non-representative correction factors. However, it is still possible to force the model results through observed SWE max which resulted in high correlations with the observations for snow maximum and melting season.

At catchment scale 13 SCA maps could be used for the melting season of 2008, due to cloud cover and non-optimal recording geometry. Using SCA map pairs with a maximum 3 day temporal gap between consecutive recordings, the approximate date of snow cover disappearance was determined by evaluating situations where the SCA for a grid cell dropped from above to below a global SCA threshold value.

The catchment scale results were assessed by comparing runoff from SnowModel with observed runoff for the Kjelstadfoss catchment. Even though observed runoff was unsatisfactorily reproduced using the interpolated precipitation correction fields, application of a linear correction to DEM height resulted in improved performance.

As a concept, the backtracking routine seems promising, but for validating the resulting precipitation correction using runoff, more hydrologic processes should be considered when simulating SnowModel runoff, i.e. evapotranspiration and soil- and groundwater processes, before more solid conclusions can be drawn. Better temporal recording consistency and spatial resolution of the remotely sensed data is believed to give more accurate maximum SWE estimates and accordingly more reliable precipitation correction factors.

Acknowledgements

This work was made possible by the collaboration of Statkraft AS, Dr. Glen Liston (Colorado State University) and the University of Oslo (UiO). The author would like to thank supervisors Thomas Schuler (UiO), Gaute Lappegard (Statkraft AS) and Bjørn Wangensteen (previously at the Norwegian Computing Center [Norsk Regnesentral, NR], now Oslo and Akershus University Collage of Applied Sciences) for their advises, guidance and mentor-ship during the work of this study. Additional thanks to Statkraft AS and NR for access to the data used in this study, Øivind Due Trier (NR) for collecting and processing the MODIS data to SCA maps, and to Dr. Glen Liston for the permission to work with his model and providing the source code.

Many thanks to my co-students Kjersti "Gnisås" Gisnås and Tobias "Door-humper" Litherland for helpful discussions and their feedbacks during the study. Good luck with your Ph.D. and master studies!

I would also like to express gratitude to my family and friends for their support. A special shout out to my homies Anders, Aug and Halvard, just because they wanted to be mentioned by name in the acknowledgements. Thanks to Astrid, Ingvild and Ingvild for company during coffee/lunch breaks and psychological support.

Contents

List of Figures

List of Tables

List of Abbreviations

BOAS	Barnes objective analysis scheme
BRDF	Bidirectional reflectance distribution function
CORINE	Coordination of Information on the Environment (EEA project)
DEM	Digital elevation model
EEA	European Environment Agency
IDW	Inverse distance weighing (interpolation)
KNN	K-nearest neighbor (classification)
MWE	Melt water equivalent
MODIS	Moderate-resolution imaging spectroradiometer
NLR	Norwegian Linear Reflectance-to-snow-cover
NR	Norsk Regnesentral (Norwegian Computing Centre)
SAR	Synthetic aperture radar
SCA	Snow covered area (fraction)
SD	Snow depth
SRTA	Spectral reflectance training areas
SWE	Snow water equivalent

1. Introduction

Snow and its distribution plays an important role in many fields of science, e.g. its high albedo influences the radiation balance, thermal conductivity may effect the distribution and depth of permafrost and the storing of water as snow and the release of the melt water is of great interest for hydrologist in areas with seasonal snow cover. In some regions the magnitude of the snow reservoir and melt water is of great importance for agricultural purposes and/or hydroelectric production, and can account for as much as 50% of the annual runoff. The most basic processes regarding snow is accumulation and melt, which can more or less co-occur (e.g. ephemeral snow covers) or one might have accumulation dominated conditions (e.g. Antarctic ice sheet) (Liston and Elder, 2006a). In addition one might add redistribution as an important factor, such as wind transport and avalanches. The accumulation and ablation of snow varies both spatially and temporally as a product of spatially and temporally varying factors, that is atmospheric conditions and their interaction with surface properties (Liston and Elder, 2006a).

1.1. Distributed snow models

In many situations one is interested in the spatial distribution of the amount of snow, usually in form of snow water equivalent (SWE) and/or snow depth. These data are hard to measure for larger areas, such as whole catchments, as most automatic snow measurement instruments are only representative at a point scale and no remote sensing techniques give accurate enough estimates for SWE. Manual measurements are often time consuming, which leads to large time gaps between each observation. A solution is therefore to use snow models to simulated the snow variables in order to get the desired spatial and temporal coverage. Typically these models take measured meteorological data, e.g. temperature and precipitation, as input and try to "mimic" the natural processes physically or empirically. These models may then, if it is of interest, be incorporated as a module of hydrological model that simulates runoff. The model complexities range from lumped and very simplified (empirical or conceptual) to distributed and physically based. An example of a lumped and (simple) conceptual snow model is the snow routine of the HBV model, Bergstrom (1995) and others, a widely used hydrological model and is the standard model for the Nordic countries. It uses a threshold temperature to determine if precipitation falls as rain or snow and degree-day methods for snow melt and refreezing. For some purposes a lumped, conceptual model might not be desirable since the model does not take into account the spatial distribution of the snow and more complex processes/principles, e.g. energy balance and snow pack evolution. There are some more or less physically based, distributed snow models, such as described by Tarboton et al. (1995) and Winstral and Marks (2002). In this study SnowModel (Liston and Elder, 2006a) is used to simulate the snow. The model consists of four sub models, namely MicroMet, EnBal, SnowPack and SnowTran-3D. They handle meteorological data, energy balance equations, snow pack evolution and redistribution of snow by wind, respectively.

1.2. Snow measurements

Error in input data or simplifications in the model may result in simulations that do not fully match the actual snow situation (Liston and Hiemstra, 2008). This can be accounted for and to some extent corrected by using assimilation or updating routines with observed snow data. There are many methods of obtaining snow data, and the current state of snow data collection methods in the Nordic countries were recently reviewed by Lundberg et al. (2010). The methods range from manual snow pit measurements via ground penetrating radar (GPR) to remotely sensed satellite data.

1.3. Backtracking SWE using physically based snow models

Several studies, e.g. Cline et al. (1998) and Liston (1999), have shown that there is a strong relationship between snow depletion patterns, end of season SWE and snow melt rate. Martinec and Rango (1981) presented a back calculation scheme for SWE using snow melt estimates and snow cover area fraction (SCA) data, and the accuracy of this methodology have been assessed by Cline et al. (1998) and Liston (1999), showing promising results. These methods utilize a SWE-SCA interrelationship in estimating the SWE using the melt rates generated by physically based energy balance model. As only snow melt is modeled and used in the back calculation of SWE, any snowfall during the melting season gives way for errors in the estimate. Liston (1999) shows that this methodology can give, assuming Arctic conditions with clear accumulation and ablation periods, good estimates of sub-grid SWE distribution if SCA observations can be acquired on a daily or more frequent basis.

2. SnowModel description

The snow model used in this study, SnowModel, is described in Liston and Elder (2006a) but with certain updates (not specified here) by Glen Liston over the last years. This section is a short description of the sub-models which make SnowModel; MicroMet, EnBal, SnowPack and SnowTran-3D.

2.1. MicroMet

MicroMet is a meteorological distribution model presented in (Liston and Elder, 2006b), which provides the rest of SnowModel with gridded meteorological variables. The model includes a data pre-processor, which detects missing data, does quality assurance /quality control of the data and fills missing parts of time series, and an algorithm that distributes the meteorological variables if necessary.

The MicroMet model requires air temperature, relative humidity, wind speed, wind direction and precipitation values from at least one point within or adjacent to the simulation domain at all times. These data are then spatially interpolated using an interpolation method that the MicroMet authors have based on Barnes (1964); Koch et al. (1983), called Barnes objective analysis scheme

(BOAS), and physical sub models are used to enhance the parameters. The outputs are distributed surfaces for the variables air temperature, relative humidity, wind speed, wind direction, incoming solar radiation, surface pressure, and precipitation.

Measured air temperatures at each station is first brought down to sea level using a monthly temperature lapse rate ($°C/km^{-1}$). The temperatures are then interpolated using BOAS, creating a gridded dataset. The values for each cell is then brought up to their topographic height according to a DEM with that same lapse rate. Relative humidity measured at the stations are converted to dew point temperature by

$$T_d = \frac{c * \ln(e/a)}{b - \ln(e/a)}, \tag{1}$$

where a is 611.15 Pa, b is 22.452, c is 240.97 °C, from Buck (1981), and e is given by

$$e = \frac{RH * e_s}{100}, \tag{2}$$

where RH is measured relative humidity in percentage and e_s is the saturation vapor pressure calculated by

$$e_s = a \exp(\frac{b * T}{c + T}), \tag{3}$$

where T is measured air temperature, before they are brought down to sea level using a dew point temperature lapse rate ($°C/km^{-1}$). Liston and Elder (2006b) explains that this conversion is done as dew point temperature behaves more linearly with elevation than relative humidity. The values are interpolated and brought up to topographic height, similarly as for air temperature. Lastly the dew point temperatures are converted back to relative humidity.

Precipitation is gridded with BOAS, and so is the measurement station height. The precipitation is then adjusted for the topography using a monthly precipitation adjustment factor (km^{-1}) and the difference between the interpolated station height and DEM.

Wind speed and direction are first converted to zonal and meridional, which are independently interpolated using BOAS. The values are then modified by a simple wind model, adjusting speed and direction according to topographic slope, aspect and curvature, and is described in detail in Liston and Sturm (1998) and Liston and Elder (2006b).

MicroMet simulates the cloud fraction and incoming solar radiation using the topography and potential incoming solar radiation, reducing it by e.g. a simulated cloud fraction. Long-wave radiation is modeled following Iziomon et al. (2003), considering cloud cover and elevation. For time steps above three hours, the incoming solar radiation is calculated for every three hours and assimilated to the model resolution. Surface pressure is modeled by a time independent distribution, dependent only on elevation above sea level. Observed short-wave and long-wave radiation and surface pressure may be assimilated into the sub models. Detailed descriptions of the interpolation routine, physical sub models and assimilation schemes of MicroMet are given in Liston and Elder (2006b).

2.2. EnBal

The EnBal model uses the meteorological forcings from MicroMet and calculates a surface energy balance for the snow pack;

$$(1 - \alpha_s)Q_{si} + Q_{li} + Q_{le} + Q_h + Q_e + Q_c = Q_m \tag{4}$$

where α_s is surface albedo, Q_{si} is solar radiation reaching the surface, Q_{li} is long-wave radiation, Q_{le} is emitted long-wave radiation, Q_h is turbulent exchange of sensible heat, Q_e is turbulent heat exchange, Q_c is conductive energy transport and Q_m is the energy available (Liston and Elder, 2006a). The equation is solved for the snow surface temperature (T_0), with melt energy equal to 0 °C, using Newton-Raphson method. A resulting T_0 above 0 °C for snow covered areas is indicating energy available for snow melt, and the energy balance is then solved for melt energy with $T_0 = 0$. Details for each term in Equation 4 and its solution can be found in Liston (1995), Liston and Hall (1995) and Liston et al. (1999).

2.3. SnowPack

This sub model, SnowPack, handles the evolution of the snow pack as a response to precipitation and melt from MicroMet. SnowPack follows Liston and Hall (1995), which in turn closely follows Anderson (1976). The model handles all inputs and outputs from the snow pack, in addition to compaction of the snow due to melting and settling. New snow is added to the snow pack with a given "new snow" density, following Anderson (1976). When melting energy is available from EnBal, SnowPack converts the energy available to a "melting water equivalent" (MWE), by the equation

$$MWE = \delta t * Q_m/(\rho_{water} * L_f), \tag{5}$$

where δt is seconds per time step, Q_m the available melt energy from EnBal and L_f is the latent heat of fusion.

SnowPack then reduces the snow depth accordingly, using the modeled snow density. The melt water is redistributed and a new snow density is calculated. If the new density is above a maximum density threshold (default value $550.0 kg/m^3$) any excess water is released from the snow pack so that the density settles at the maximum value. If rain falls onto the snow pack, the SWE of the snow pack is raised accordingly, and the density recalculated. Similarly to the melt situation, any excess water is released if the new density is above the threshold. The model also handles canopy interception and the release of this snow from this storage, but this is not too relevant for this study as only areas above the tree line is considered. Over time the snow pack is compacted due to settling, which is modeled by the equation

$$\Delta\rho_{snow} = dt * C_1 * 0.5 * d_{snow} * \rho_{snow} * \exp(-0.08(T_f - T_s)) * \exp(-C_2 * \rho_{snow}), \tag{6}$$

where $\Delta\rho_{snow}$ is the change in snow density, dt seconds in the model time step, d_{snow} is the snow depth, ρ_{snow} the snow density before reduction, T_f is 273.16 °K, C_1 is the fractional density in-

crease per time unit, which is set to 0.0013 in the code, C_2 an empirically estimated constant set to 0.021 and T_s the snow pack temperature estimated, assuming ground-snow interface temperature of -1 °C, by

$$T_f = 0.5 * ((T_f - 1) - T_{sfc}), \tag{7}$$

where T_{sfc} is the modeled snow surface temperature.

SnowPack also considers sublimation from the snow pack, but the modeled sublimation is of such a small magnitude for the study areas that is makes no major impact. It is, however, still considered as a process to remove snow from the snow pack. A notable process that is missing in this version of SnowPack is modeling of re-freezing of liquid water in the snow. Details regarding this models may be found in Liston and Hall (1995).

2.4. SnowTran-3D

Redistribution by wind of the snow is governed by the three dimensional sub model SnowTran-3D as described by Liston and Sturm (1998) and further in Liston et al. (2007). This model has previously been tested on several sites, including a Norwegian alpine catchment (Bruland et al., 2004). The change in snow depth ($\frac{\delta \zeta_s}{\delta t}$) is modeled by the equation

$$\frac{\delta \zeta_s}{\delta t} = \frac{1}{\rho_s}[\rho_w P - (\frac{\delta Q_s}{\delta x} + \frac{\delta Q_t}{\delta x} + \frac{\delta Q_s}{\delta y} + \frac{\delta Q_t}{\delta y}) + Q_v], \tag{8}$$

where ρ_s and ρ_w is snow and water density, Q_s transport by saltation, Q_t transport by turbulent suspension, Q_v sublimation from transported snow and P is precipitation. Depending on the surface layer input to the model, each cell will have a snow retention capacity where only a snow depth above this threshold will be available to transport by wind. The model is described in detail by Liston and Sturm (1998), and several improvements to the model are presented by Liston et al. (2007). The improvements include a new wind model, two-layer parametrization of threshold friction velocity for the snow pack and an implementation of the Tabler (1975) model for snow drift profiles.

2.5. SnowAssim

The snow assimilation subroutine may be used with SnowModel if there are snow observation data available, and is described in Liston and Hiemstra (2008). A simple assimilation scheme was chosen in order to not increase the processing time of SnowModel simulation too much (Liston and Hiemstra, 2008). The scheme assumes that the differences between the model simulated values and the "true" observed values are mainly caused by errors in either precipitation forcings or the modeled snow melt. For each observation point the difference in simulated and observed SWE is calculated. In order to determine which of the two processes (precipitation or melt) that is governing in the evaluated period, the relative contribution (R) of precipitation (P) and snow melt (M) is calculated by

$$R_{prec} = \frac{\sum P}{\sum P + \sum M} \tag{9}$$

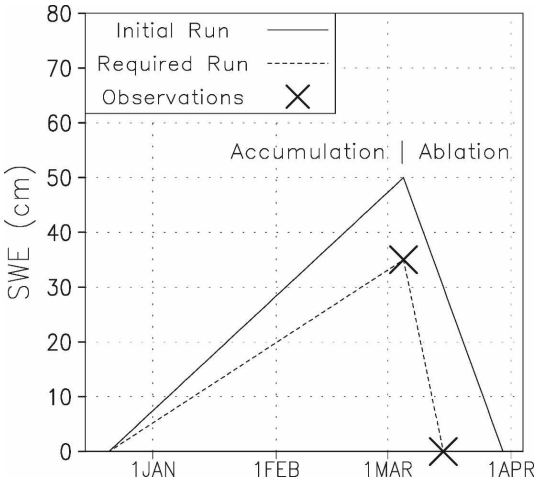


Figure 1: Conceptual figure of the assimilation routine with observations at SWE peak and end of season. From Liston and Hiemstra (2008)

$$R_{melt} = 1 - R_{prec} \tag{10}$$

for the time period in question. The process with the greatest relative contribution is then scaled using a correction factor. The precipitation correction factor is calculated by

$$P_{fact} = 1 + \frac{(SWE^t_{obs} - SWE^t_{sim}) - (SWE^{t-1}_{obs} - SWE^{t-1}_{sim})}{\sum P}, \tag{11}$$

where subscripts *obs* and *sim* represent observed and simulated values respectively. The model is then re-run from the beginning of the season, or last observation point, to the observation registration time with the corrected precipitation data. A similar procedure is done for snow melt if R_{melt} is greater than R_{prec}. Figure 1 illustrates the concept of SnowAssim.

3. Remote sensing of snow by satellite

Remote sensing is often used to acquire information on surface properties over vast areas. There are two main categories of remote sensing techniques based on the part of the electromagnetic wave spectrum they utilize; optical, which records visible, near-infrared etc. frequencies, and radar sensors, e.g. synthetic aperture radar (SAR). Both classes have both passive and active systems. Passive optical sensors systems have often been used to map snow cover distribution, but with the significant drawback that no measurements can be made through cloud covers. On the other hand, radar waves can penetrate cloud cover, but is only able to map wet snow and is very sensitive to vegetation and snow free spots, which are wrongly classified as snow free (Storvold et al., 2005). As the snow cover to snow free state transition is of uttermost importance for this study, radar based products were not used. There are several operational snow mapping products derived from satellite imagery, and also several operational SWE mapping services, but these data are of rather coarse spatial resolution, and therefore not well suited for regional modeling.

In this study, a snow cover area (SCA) product from the Norwegian Computing Centre (Norsk

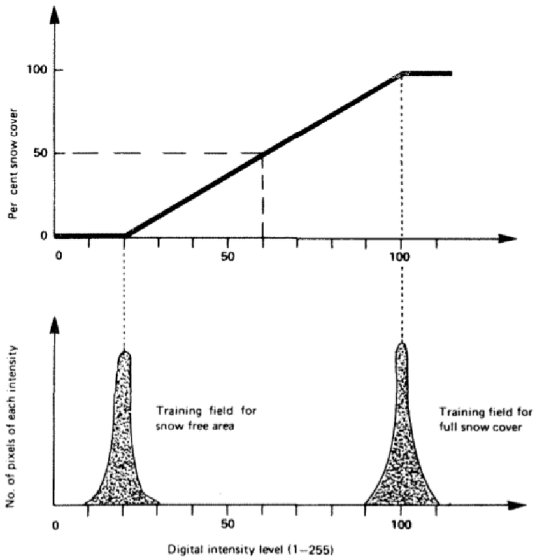


Figure 2: Illustration of the NLR algorithm. A pixel with DN along the X-axis is assigned a fractional snow cover, Y-axis, from the linear transformation between the two training clusters. From Andersen (1982)

Regnesentral, NR) is used as remotely sensed observation of the snow cover. The optical observations are processed using NRs optical snow algorithm, presented in Solberg et al. (2006). It is based on the Norwegian Linear Reflectance-to-snow-cover (NLR) algorithm presented by Andersen (1982) and Solberg and Andersen (1994), which was originally developed for NOAA AVHRR data, but modified for MODIS data by NR. For the fractional snow cover algorithm, several training areas are selected for calibration of the linear model. 100% SCA is determined by a linear transformation of the mean of the high-level spectral reflectance training areas (SRTA), and 0% by the mean of the low-level STRA (2). The high-level SRTA are set at glaciers and low-level in boreal forests with little or no anthropogenic surfaces. For each pixel that is not excluded by the cloud detection algorithm, i.e. cloud and sea pixels, SCA is calculated according to the linear transformation of the training areas (Figure 2).

NRs algorithm includes a cloud detection routine based on K-nearest neighbor classification (KNN), using the MODIS bands 1, 4, 6, 19, 20, 26 and 31, which includes wavelengths from 0.545 to 11.280 μm, and sun angle at the observation time. Band 6 is most important regarding separation of snow and cloud cover, since clouds have high reflectance and snow has low (Solberg et al., 2006). KNN is a simple, supervised, image classification algorithm, which looks at the K nearest trained neighbors in the multidimensional parameter space and assigns the pixel in question to the class that is in a majority. The classes used in the classification are cloud, land, ocean and snow cover.

Solberg et al. (2006) points out several weaknesses of the traditional NLR algorithm; that is does not take into account how topography affects reflectance of solar radiation, does not consider the decay of reflectance of the snow throughout the season due to snow age and pollutants, and that it does not handle the variation bare ground surface reflectance. NR has improved on the NLR algorithm by introducing a bidirectional reflectance distribution function (BRDF) model grid.

The change in reflectance is modeled with a BRDF model described in Solberg et al. (2006), with further details in Solberg (2004) and Solberg (2005). The model features two BRDF grids, one for snow covered areas and one for snow-free areas, based on observations from many snow seasons. The BRDF grids accounts for changing illumination angles and different acquisition angles. The fractional snow cover algorithm works iteratively, doing a spectral unmixing based on the BRDF grids, modeling snow metamorphosis and impurities until the change between two iterations are insignificant or until a maximum number of iterations are reached.

Further details on the above mentioned remote sensing algorithms and discussions regarding these can be found in Solberg et al. (2006), Storvold et al. (2006) and Solberg et al. (2006).

4. Study area and input data

The study area is centered in the catchment of the river Nea in Mid-Norway, which runs out to the sea at the city of Trondheim (Figure 3). In SnowModel the area is represented by a digital elevation model (DEM) with original spatial resolution of 50 m and an vegetation map based on CORINE (Coordination of Information on the Environment) Land Cover data with 100 m resolution. CORINE Land Cover is an European project, now under the European Environment Agency (EEA), for gathering environmental information, and provides land cover information based on satellite data. The Trondheim fjord is located in the north-west of the area, and the elevation increases east- and southwards. According to the CORINE data, the north-western part is dominated by agriculture and coniferous forest, and the valleys in the inland have deciduous forest. As elevation increases, bog-lands and marshes are dominant and the vegetation is lower and more spare the higher the elevation. The vegetation map used in SnowModel is shown in Figure 4, the classes are based on the description of the CORINE data. Although the data have many specific classes, the most important is to be able to distinguish between forested and non-forested areas as this will determine whether canopy interception, solar radiation reduction etc. will be considered in the code.

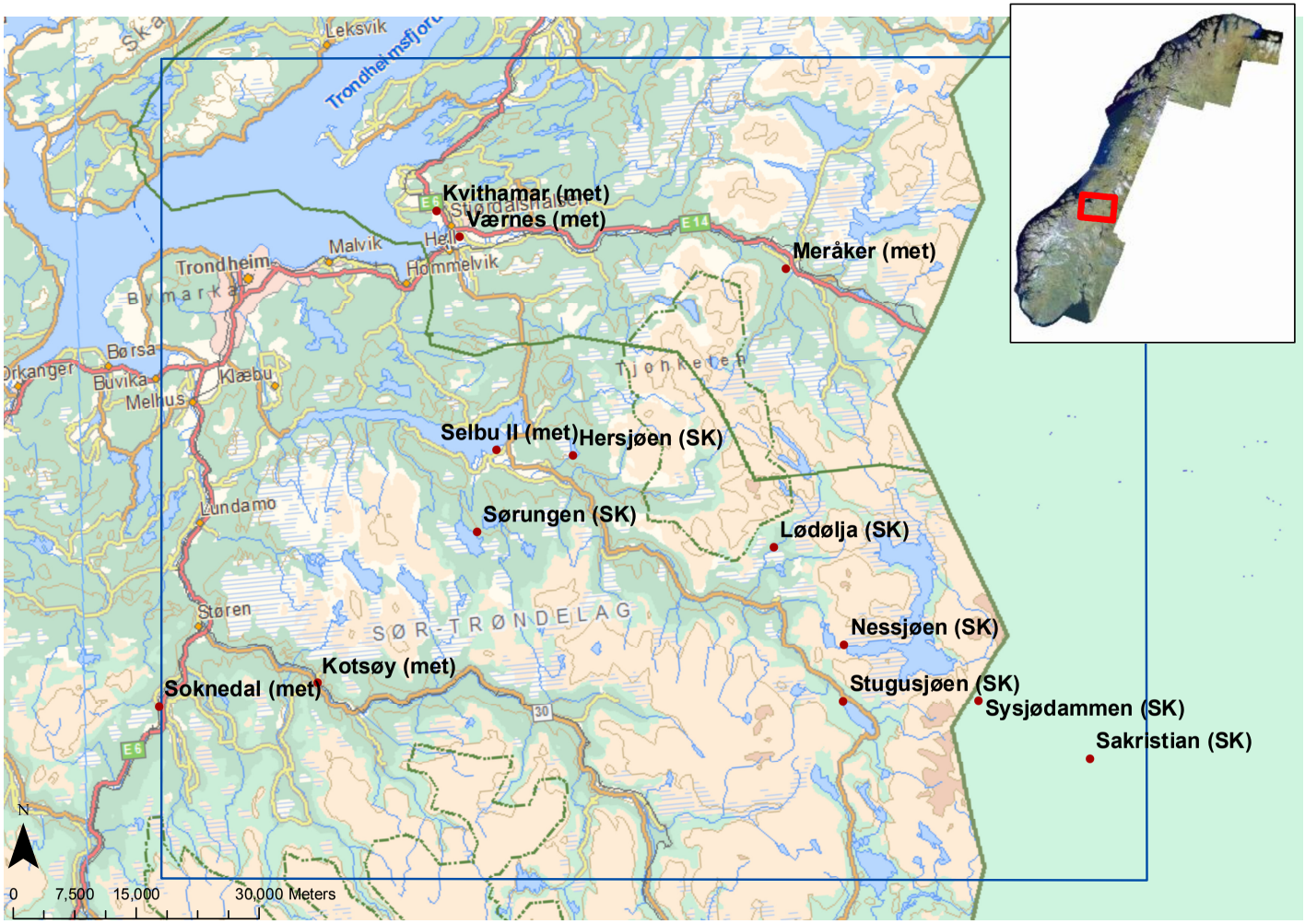


Figure 3: Map and overview over the study area (marked with blue rectangle). Meteorological stations are marked with station name. Data origin/owner indicated in parenthesis; *met* meaning data from the Norwegian Meteorological Institute and *SK* data from Statkraft AS

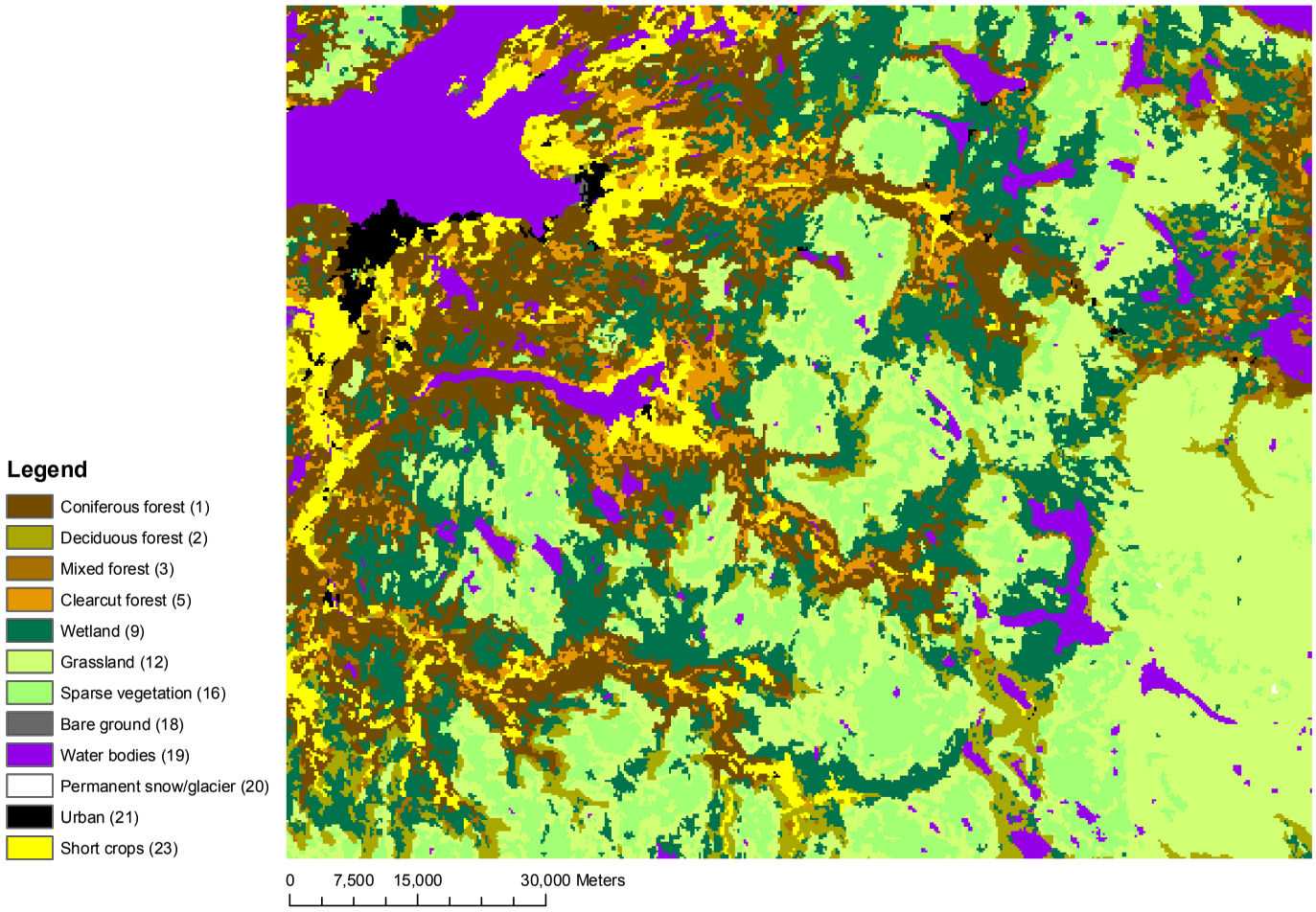


Figure 4: Vegetation map used in the study, based on CORINE data for the area and translated to SnowModel vegetation classes by description. SnowModel vegetation class given as numbers in parenthesis, described in Appendix B

The meteorological data to drive the model were acquired from stations owned by Statkraft AS and the Norwegian Meteorological Institute located in the region (Figure 3). Hourly data for 2007-2010 were downloaded for each of the stations listed in Table 1, which also shows which variables that are were uploaded from each station. In addition to meteorological variables, three of Statkraft AS' stations have functioning snow pillows combined with snow depth ultrasonic sensor.

Hourly runoff values were retrieved for Kjelstadfoss station (Figure 5), which is a measuring station in the river Kjelstafossen, which flows into Selbusjøen from the north-east. In order to asses the model quality, the catchment for this station was delineated using a 250 m DEM, and direct runoff from SnowModel summed up for all grid cells within the catchment, i.e. assuming direct response. The position of the measuring station and the catchment is shown in Figure 5. The delineated catchment has an area of 150 km^2, and has a few small lakes and some wetlands or marshes.

Table 1: List of meteorological stations and variables that are provided in hourly resolution at each station.

Station name	**M.a.s.**	**Temp.**	**Rel. hum.**	**Wind spd.**	**Wind dir.**	**Precip.**	**SWE**	**Snow depth**
Kotsøy	127	X	X	X	X			
Kvithamar	40	X	X			X		
Meråker	169	X	X	X	X			
Selbu II	299	X	X	X	X			
Soknedal	12	X	X	X	X			
Værnes	420	X	X	X	X			
Hersjøen	420	X	X	X		X	X	X
Lødlja	540	X	X	X		X		
Nessjøen	725	X	X	X		X	X	X
Sakristian	860	X		X	X	X		
Stugusjøen	630	X	X	X		X		
Sylsjødammen	840	X	X	X	X	X		
Sørungen	460	X	X	X	X		X	X

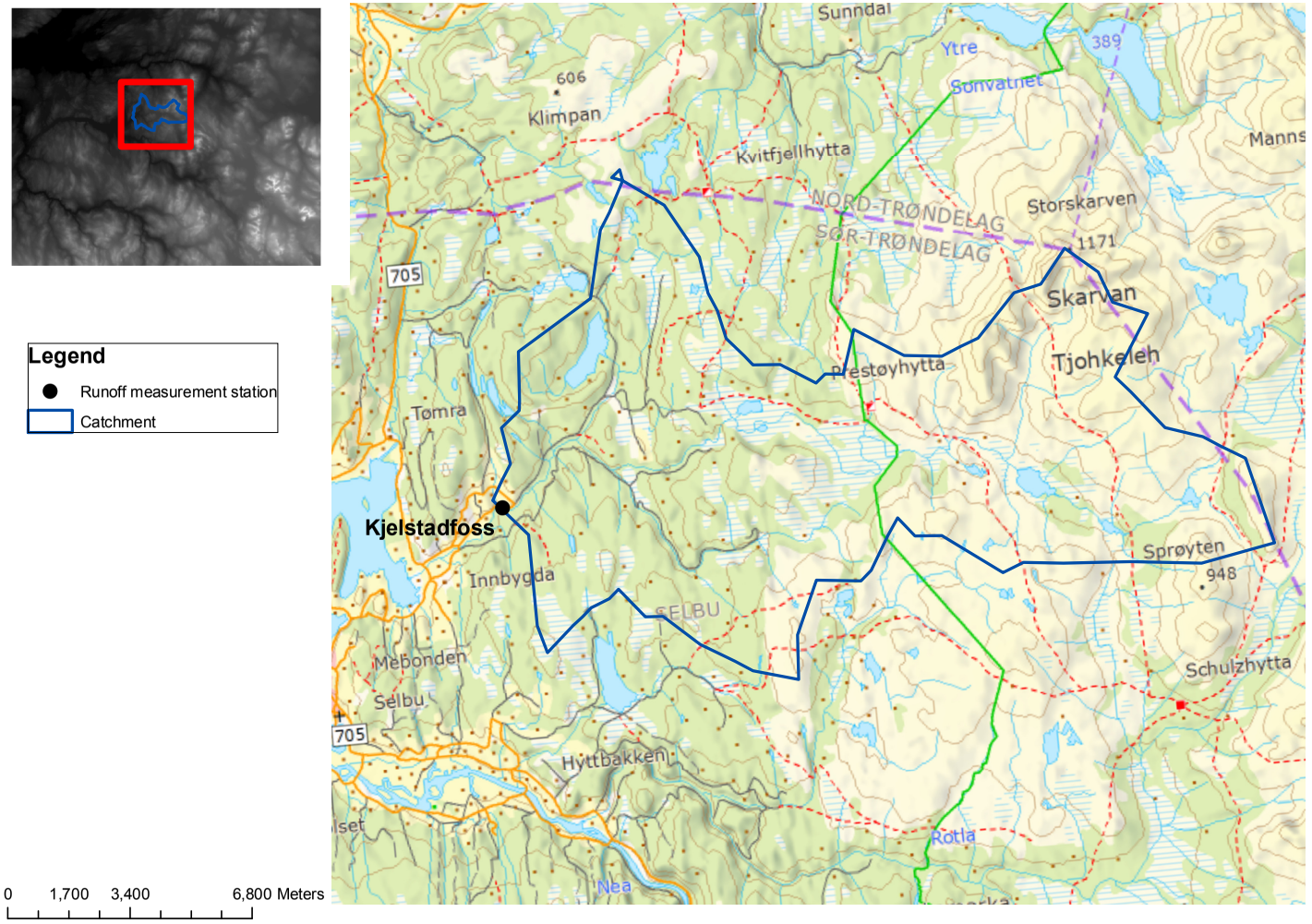


Figure 5: Map showing position of runoff measuring station Kjelstadfoss, and the catchment delineated from a 250 by 250 m DEM.

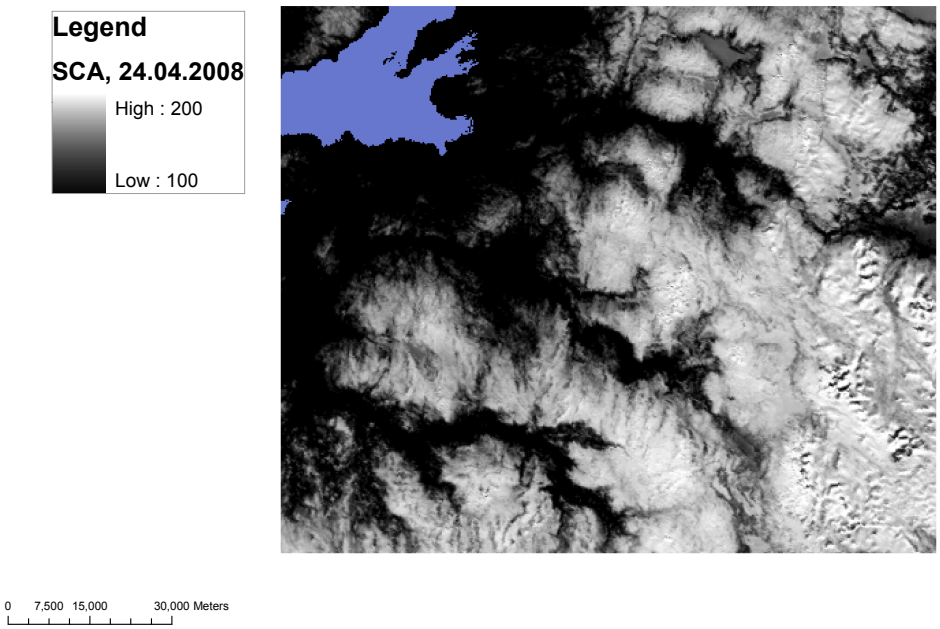


Figure 6: Example of SCA from NR, transformed and clipped to study area. Value 100 equals 0% snow cover, 200 equals 100%. Map from 24. 04. 2008

Raw images from MODIS-Terra from March to July 2008 were visually inspected in order to find suitable images in terms of coverage, cloud cover and recording geometry. The selected MODIS images were downloaded and processed at NR. The SCA-products arrived as a map of the whole of Norway in UTM33 WGS84 projection, so the maps were reprojected to UTM32, WGS84 and clipped to fit the study area (Figure 6). Although the SCA algorithm takes recording geometry into consideration, an inspection of the data showed that two recordings close in time, e.g. one repeat cycle apart, with different viewing geometry gives some percentage of difference in the SCA, sometimes increasing the values in the melt season. Since this increase the possibility of false detections of zero SWE dates, observations with high viewing angles were excluded from the data set. The weather along the coast of Norway is often cloudy, giving rather sparse amount of data over time. The recording dates that were found suitable for the analysis is listed in Table 2.

As the study is heavily based on MODIS images with a 250 m spatial resolution, all spatial data were aggregated to this resolution using averaging for the DEM and nearest neighbor method for the vegetation data, as these are discrete values. All vegetation classes were reclassified to SnowModel-compatible variables based on the CORINE description of each class. SnowModel was ran both 24- and 3-hour time resolution, and thus all measured meteorological variables were aggregated up to these values. The two different time resolutions were chosen in order to compare simulations where diurnal variations are modeled and where they are not in order to assess the importance of time resolution in the study. Hourly simulations require much longer processing time, and was therefore not tested in this study.

The SnowModel code used in the study was uploaded by Glen Liston 19. January 2011, and

Table 2: MODIS image recoding times

Date	Time
24. 04. 2008	10:51
26. 04. 2008	10:40
05. 05. 2008	10:35
08. 05. 2008	11:05
10. 05. 2008	10:51
28. 05. 2008	10:40
29. 05. 2008	11:22
29. 05. 2008	11:25
02. 06. 2008	11:00
04. 06. 2008	10:45
07. 06. 2008	11:16
02. 07. 2008	11:10
04. 07. 2008	11:00

was at that time the latest version of the code.

5. Methodology

As the MODIS products have a spatial resolution of 250 m, the same resolution was used for the SnowModel simulations. SnowModel is mainly designed to work at spatial scales less than 200 m (Liston and Elder, 2006a), but may be used at lower resolution at the cost of high resolution information loss (Liston, 2004). When running at this scale, the snow transport model is thought to no longer be relevant or representative, and in fact an initial test of running the model with and without the SnowTran-3D sub model shows little to no difference, at least when comparing at the snow pillow locations. The absence of snow transport by wind is also favorable in regard to the fact that the assimilation scheme adjusts the precipitation locally for each grid cell, so it is assumed there is no inter-pixel flux. Additionally, runs using the model default and no precipitation lapse rate showed very little difference in the snow accumulation, and the difference in accumulation compared to the snow pillows are of no significance. Therefore all simulations use a lapse rate of 0.0 in the code for all months, so that the correction field calculated from the snow melt values will give an indication of how the (snow) precipitation factors should be spatially distributed. Other than turning SnowTran-3D off and setting precipitation adjustment factors to zero, the other parameters were kept at default values for all simulations, except those associated with input data and simulation time. The complete list of SnowModel user set parameters are given in appendix C. All simulations showed that for both snow pillow and runoff data comparison, classifying the CORINE class for clear cut forest as the corresponding class in SnowModel, i.e. SnowModel calculated canopy interception and solar radiation reduction etc. for these areas, gave worse simulations compared to simulations where this class was set to some other non-forest SnowModel class. Therefore all simulations presented in this paper are similar to the map shown in Figure 4, except that the clear cut forest class is reclassified to a similar class that is not handled as forest in the code.

5.1. Backtracking seasonal maximum SWE using SnowModel

It is proposed that, knowing the date when an area is totally snow free, one can trace back the snow melt simulated in SnowModel which will result in a maximum SWE value for this area, and previous studies have shown that it is possible (e.g. Liston (1999)). Most of these studies have only relied on simulated snow melt from physically based energy balance models to backtrack the SWE-values, making late season snow precipitation a source of error in areas where this occurs to some degree. By including sub-models which will give input as snow precipitation during the melt, evolution of the snow pack depth/density throughout the season etc. it is expected to be possible to not only give a better estimate of the maximum SWE value, but also find a precipitation adjustment factor field that can improve the SnowModel simulation to, similar to the methodology in SnowAssim (Section 2.5). In contrast to the previous studies using remotely sensed SCA evolution at each grid cell, this study simplifies the methodology and focuses only on the point in time where the SCA reaches a zero SWE situation and assuming this to be representative for the SWE modeled for grid cells, i.e. not actually considering the sub grid snow distribution at all and assuming a uniform snow cover over the whole grid cell in the simulated values.

All SnowModel modules are associated with the melting dynamics; MicroMet grids and models all meteorological variables used in the energy balance solving, precipitation in rain or snow form, etc, EnBal solves the energy balance and calculates energy available to melt snow and sublimation, and finally SnowPack melts or accumulates snow, as described previously. In late autumn/early winter one might have both accumulation and melting of snow, giving a fluctuating snow cover, and even situations where the previously accumulated snow is completely melted. This may influence the snow melt value in cases where the snowfall is over-estimated, because the model will allow a lot more melt if more snow is available. Therefore, only data from after 1. January is considered in the backtracking calculations, i.e. considering the main accumulation season. In the ripening phase of the snow pack, SnowPack melts out the MWE (Equation 5) and redistributes this water evenly in the snow pack. This reduces the snow depth and thus the snow density is increased. When the snow pack is ripe, this melt will result in runoff from the snow pack. Since the model does not handle liquid water content in the snow pack, and does not handle refreezing, this algorithm will in a lumped sense refreeze all melted snow pre-output phase, in that the MWE is not removed from the snow pack until the density threshold is reached. Because of this, in order to backtrack the maximum SWE one must look at the simulated runoff/output from the snow pack during the melt season rather than the MWE. During the melt season it may also fall precipitation as snow, adding additional melt able snow to the snow pack which is not to be included in the maximum SWE value. The back-calculation is very similar to the methodology used by Liston (1999), but with a slight modification in that it accounts for snowfall during the melting season and uses runoff rather than melt energy;

$$SWE_{max} = \sum_{t_{max}}^{t_0} (Q_{runoff}(t) - P_{snow}(t)), \tag{12}$$

where t_{max} is the point in time where the simulated SWE is at maximum value, t_0 the observed time where SWE = 0, Q_{runoff} the runoff produced from the snow pack due to melting, and

P_{snow} precipitation as snow added to the snow pack during the output phase, all values in water equivalent depth. This method differs from those of previous studies mentioned in Section 1.3 in that it does not only take into consideration the energy in the melting season, but extracts data from a simulated snow pack. This procedure does not take into account any rain retained in the snow pack during output phase as any new additional snow is assumed to have low retention capacity and a ripe snow pack will in this algorithm not retain any rainwater. Equation 12 will work in situations where the model overestimates the precipitation, but will not work in cases where it is underestimated. The reason for this is that SnowModel stops calculating melt energy as soon as the snow cover is gone. For this reason, parallel simulations from 1. January until 1. Sept were ran with an initial snow cover of 5 meters depth at 1. January. This will allow the melting calculations to continue further in time than the original run, and the maximum SWE is then given by

$$SWE_{max} = \sum_{t_{max}}^{t_{sim=0}} (Q_{runoff,1}(t) - P_{snow}(t)) + \sum_{t_{sim=0}}^{t_0} Q_{runoff,2}(t), \tag{13}$$

where $t_{sim=0}$ the point in time where this simulation is snow free, $Q_{runoff,1}$ is runoff due to snow melt from the original simulation, and $Q_{runoff,2}$ runoff due to melt from the exaggerated snow cover simulation. For situations where the simulated zero SWE situation is later than what is observed, Equation 12 is used, and Equation 13 for when snow free condition is reached too early.

The precipitation correction scheme is based on SnowAssim in that it calculates a correction factor that is used to force the model to better fit the observations. In this study, all uncertainties associated with the snow cover is assumed to come from uncertainties in the measurement and gridding of snow precipitation. As snow precipitation is the only means in the model that will add SWE to the snow pack in each grid cell and the above mentioned runoff due to snow melt is output, the water balance of the snow storage is $P_{snow} = Q_{runoff}$. The snow precipitation correction factor is therefore estimated by the formula

$$P_{corr} = \frac{\sum_{t=t_j+1}^{t_{zero}} Q_{runoff}(t)}{SWE(t = t_j) + \sum_{t=t_j+1}^{t_{zero}} P_{snow}(t)}, \tag{14}$$

where t_j is 1. January, t_{zero} is the time where a zero SWE situation is observed and Q_{runoff} is the runoff due to snow melt, with added values from the parallel run if zero SWE occurs too early, $SWE(t)$ the SWE storage in the snow pack and P_{snow} the precipitation as snow.

5.2. Multi-temporal SCA maps as zero snow indicators

As mentioned, optical remotely sensed data from satellites have a major disadvantage in that information below cloud cover is irretrievable. This makes it highly improbable that one will get observations at all points in the study area at a daily basis, and it is therefore unlikely to get a continuous grid of zero SWE situation observations. Non-optimal viewing geometry may also add to the length of the time gaps between each SCA observation. However, if two observations are close enough in time, one may use the difference in SCA values to estimate the point in time that the grid cell is virtually snow free. It is proposed that below a certain threshold of SCA the SWE

remaining snow is so sparse and thin that its SWE is insignificant. The zero SWE observations are therefore found by simply extracting pixels where the SCA value drops below a certain, global threshold value between two consecutive observation dates. The recording time of the last of the SCA map pairs will then be used as the date of snow cover disappearance observations.

The maximum amount of days between observations to determine zero SWE situations was set to three days. This rather coarse lower temporal resolution may give a lot of uncertainty, as much melting may have occurred over three days, resulting in inexact melt out observation. In the SnowModel 3 h simulations, large melt out events are typically in the range of 30 to 40 mm on daily basis. However, since the melt-out of pixels follows the topography to some extent, one may filter out the pixels that are most likely a true zero SWE observation at the last observation date from those that became snow free at some earlier point in time between the two observations. This was done by running a 3x3 window over each of the zero SWE observations resulting from an observation pair. The window checks if the pixel has any neighbors with a SCA value higher than the threshold in the original SCA map, excluding all observations that does not pass this test. This will result in a narrow band of zero SWE observations on what is assumed to be the snow line at the last of the two observation dates. As the snow cover melts, the roughness of the area within each pixel and influence of wind transport will give different melt patterns. A completely flat area with no wind and equal distribution of all variables will in theory go from 100% to 0% snow cover almost immediately, and represents how SnowModel in a sense simulates the snow melt in these runs. However, as this is very rarely the case for natural surfaces, the observed SCA from MODIS will gradually go from completely snow covered towards a snow free state. Although previous studies, e.g Liston (1999), utilize a relationship between SCA and SWE can be established, it is likely that this relationship will vary from grid cell to grid cell especially at such coarse spatial resolution as the one used in this study. As there is no data to base this relationship on in this study area, only a melt out threshold is used. It is obvious that, depending on the topography at sub-grid scale, at some SCA threshold above 0% an 250 by 250 m grid cell may be considered practically snow free. For example a grid cell with 10% SCA and assuming that the remaining snow has a depth of 50 mm SWE, in terms of a SnowModel simulation, this is equivalent to model a state of 5 mm snow cover for this pixel. This snow cover is so thin that the model is most likely not able to catch it at this temporal and spatial resolution. It is therefore thought that, although very simplified, there is some threshold of SCA where the remaining observed snow cover is so small that it is negligible. Obviously the SCA threshold at where one considers a pixel snow free is a critical parameter in this analysis.

An analysis using a SCA threshold value close to 0 will be sensitive to noise and varying viewing geometry, as well as small errors in the estimation of SCA, and may not be representative of a zero SWE state in the model, as discussed previously. In addition, low SCA values represents a very patchy snow cover, and in terms of SWE this might already be as good as a "no snow" situation. Because of the uncertainties associated with this threshold value, different threshold values were tested and the model results compared to relevant observed values in order to test its sensitivity and best-fit value. Observations in forested areas were ignored, as there is a lot of uncertainties associated with remote sensing of SCA in forested areas.

Two SCA observations relatively close in time will give a belt of zero SWE observations along

the snow-line. Since one pair will only give observations more or less following the topography, a correction calculated for these grid cells will not represent the region as a whole. If one is able to acquire such image pairs somewhat regularly throughout the melting season, one will get better coverage and better representation of the study area. An image pair give one zero SWE observation map, and all observations done through a season is gathered in one single map, where the pixel value corresponds to the date at which the zero SWE situation was observed.

Based on this snow free date map, one can calculate correction factors for each grid cell by extracting simulated values from SnowModel based on the corresponding date, and using the equations in Section 5.1. The resulting map is, however, not continuous as the correction factors can only be calculated for cells where the melt out time is known. In order to make a correction map that can be used to correct SnowModel, the values were interpolated over the whole study region using Inverse distance weighing (IDW) to a 1 km grid, and then scaled down to 250 m in order to smooth the correction field and reduce the influence of outliers and noise. A few other methods to represent the whole area are also tested in this study.

5.3. SnowModel setup and changes to the source code

SnowModel has many parameters that may be set/changed by the user in order to fit the model to different simulation conditions.

The original MicroMet code uses, as mentioned, monthly values for the temperature lapse rates, vapor pressure coefficient and precipitation adjustment factor (values in Liston and Elder (2006b)). Since the precipitation is the parameter which will be corrected in this study, the precipitation correction factor was set to 0 so that a resulting correction map will not be influenced by potentially erroneous precipitation correction factors.

Investigating the air temperature and dew point temperature lapse rates using the meteorological station height and observations, it was evident that the default monthly lapse rates used in SnowModel does not fit well with the climate at this study area, e.g. the temperature gradient is much steeper in the spring months, most likely due to bare ground in the lower parts and snow cover in the upper parts. Both the temperature lapse rate and vapor pressure coefficient will most likely vary at a time scale much finer than monthly time steps. Because of this, new values were calculated for each time step in the meteorological input files. For the temperature lapse rates, this was simply done by fitting a linear curve to the temperature and station elevation, using the least square method. In the original code, the vapor pressure coefficient is never directly used, but is converted to a dew point temperature lapse rate. As the dew point temperature lapse rates are assumed to behave linearly in MicroMet, the dew point temperatures were calculated for each time step using equations 1 to 3. The lapse rate was estimated using a linear fit and added to the MicroMet input files along with the calculated temperature lapse rates. Small modifications were made to the MicroMet source code in order to handle these new lapse rates.

In addition, some changes were made to the code in order to extract the variables of interest for both snow pillow and catchment areas.

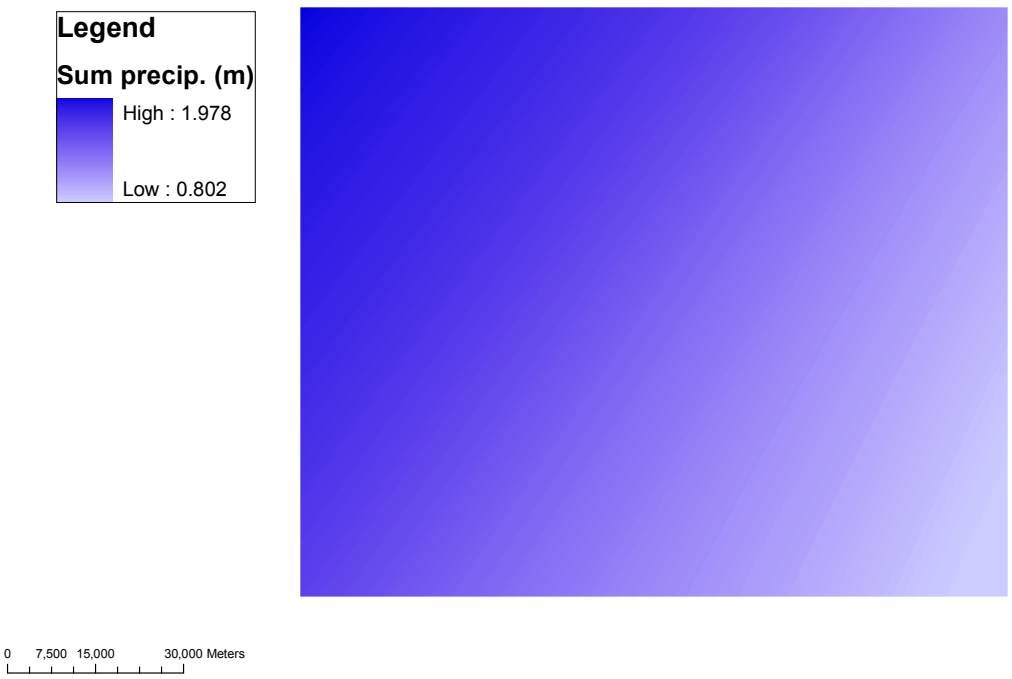


Figure 7: Interpolation of precipitation without lapse rate, annual sum (Hydrologic year 2007-2008).

6. Results

6.1. SnowModel meteorological variables

For this study, no precipitation adjustment by height is done. The consequence of this is that the interpolation of precipitation results in a very smooth precipitation field (Figure 7). The precipitation field shows a decrease in precipitation from north-west to south-east, probably a result of increasing distance from the ocean. If a precipitation lapse rate is used in MicroMet, the precipitation in each grid cell is scaled according to the difference between topography and interpolated station height (Figure 8). This difference field, showed in Figure 9, gives an indication of how a lapse rate would increase or decrease the precipitation in Figure 7. Figure 10 shows the average temperature field of the study area for 1. Sept 2007 to 31. August 2008. Since this variable used a lapse rate calculated at each time step, the distribution closely follows the topography.

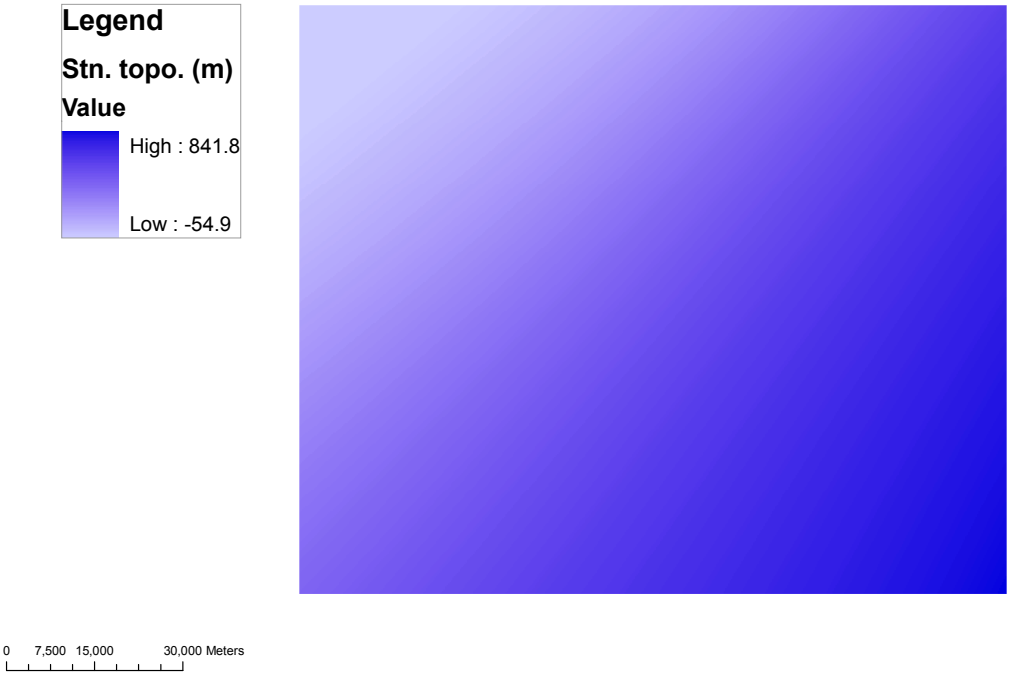


Figure 8: Interpolated station elevation, used in cases where precipitation adjustment rate is given.

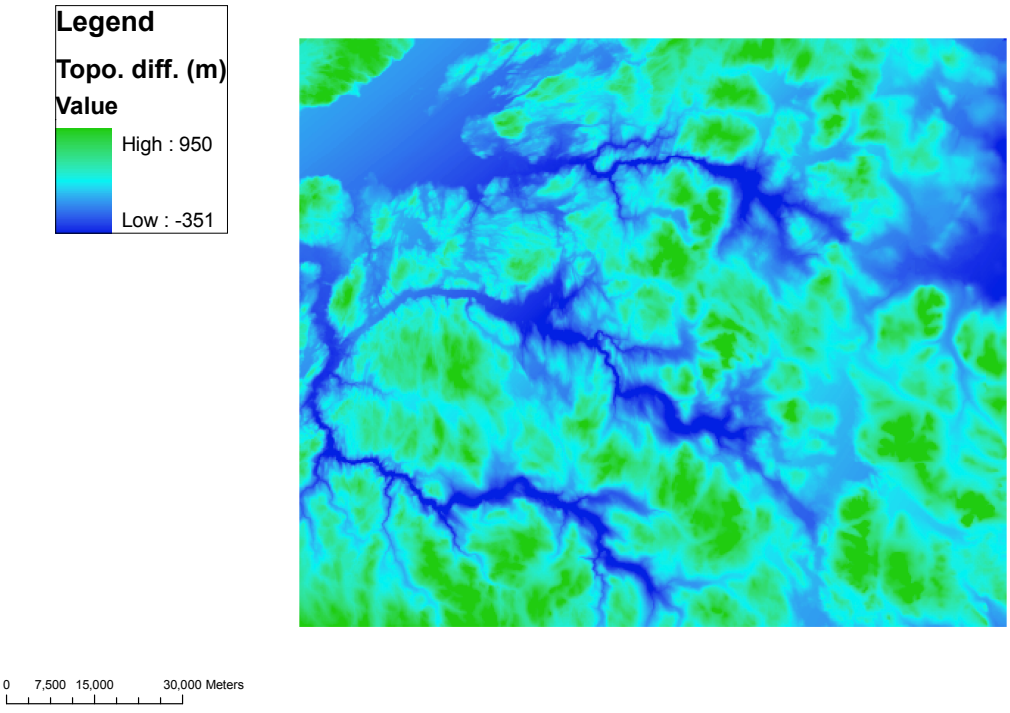


Figure 9: Difference [DEM] - [interpolated station elevation], gives indications of how precipitation would be scaled if a precipitation lapse rate is given. Negative values would decrease and positive increase precipitation.

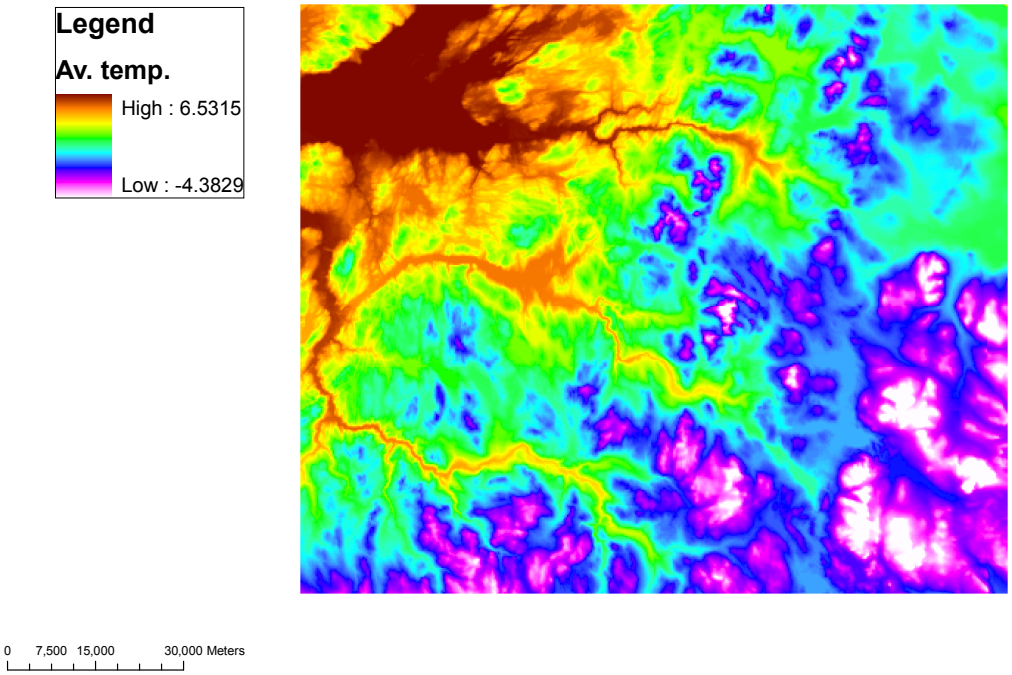


Figure 10: Interpolation of temperature, annual average temperature (Hydrologic year 2007-2008).

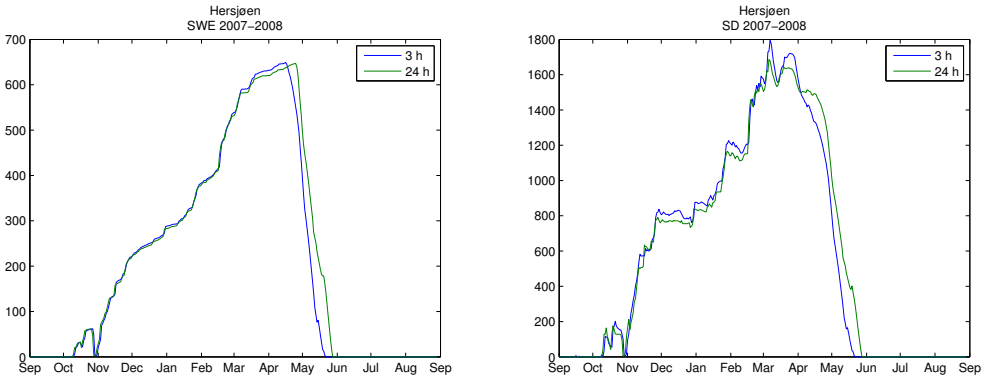
6.2. Test of concept: backtracking SWE using SnowModel

6.2.1. Initial runs

In order to test the concept of backtracking the seasonal maximum SWE, values were extracted from the simulation results at points where working SWE and snow depth measurements are located (Hersjøen, Nessjøen and Sørungen (Figure 3 and Table 1). These values were compared to the observed values in order to assess the viability of the concept. The meteorological variables used by the model at these points are influenced by the interpolations done in SnowModel, which most notably influences the precipitation distribution, as it was not adjusted to the topography and thus only a smooth field (Figure 7).

Initial runs showed that the density SnowModel uses as maximum snow density before output occurs is too high compared to the density at which the measured data starts melt-water output. Averaging the values for all three stations when output occurs gives a maximum snow density value of 488 kg/m^3, and this value was substituted into the source code. Stations and seasons where the SWE has unrealistic values when compared to the measured snow depth were ignored in this calculation. From the snow-pillow and snow depth data, the point in time where the measured snow cover reaches zero was found and used in the backtracking routine, using equations 12 and 13, as a test of concept. For cases where there are a mismatches in the zero date between observed SWE and snow depth, the latter observation series was used.

The snow pillow and snow depth data for 2009-2010 show that in the beginning phases of the melting season the SWE and snow depth relationship give snow densities much lower than what



(a) SWE

(b) SD

Figure 11: Comparison of initial simulations of SWE and snow depth evolution at 3 h and 24 h temporal resolution for 2007-2008, showing how the output phase starts later in the 24 h run

is expected of a ripe snow pack, e.g. the density is at about $300kg/m^3$ at Nessjøen, where the maximum density in the SnowPack code is $488kg/m^3$ (2.3). Such erroneous SWE evolutions are common for snow pillows, and the maximum observed snow values are most likely inaccurate for some of the seasons. The best set of observations considering both SWE and snow depth is for 2007-2008, and the majority of the analysis is done for this season.

The test was done on the hydrological years (1. Sept. to 31. Aug.) from 2007 to 2010. Table 3 shows the backtracked and observed SWE maximum values at the snow pillows at 24 and 3 hour increment runs for 2007-2008. The daily resolution runs seem to greatly underestimate the maximum SWE, while the 3 hour resolution runs are much closer to the observed values, meaning that EnBal and SnowPack as of this version produces too little melt at daily temporal resolution. This is also evident in Figure 11, as both 3 h and 24 h runs reaches the melting season with similar snow depth and SWE values, but the 24 h simulation uses far more time to melt away the snow, and the output phase starts later for the 24 h simulation. Based on these results, the rest of the simulations were done on a 3 hour time resolution, as the daily simulations does not seem to be able to reconstruct the maximum SWE value by using this particular backtracking scheme.

Results from the two parallel simulations, i.e. normal run and run with exaggerated snow cover, were extracted at each snow pillow point and the correction factor calculated using Equation 14, results are shown in Table 3. Based on the station elevation it might seem that the correction factor increases with elevation.

The accumulation starts very late at Hersjøen and Sørungen during the last year. At Nessjøen, which is located higher, it was registered an ephemeral snow layer rather early that completely melts out and there is a long period of little to no snow before the accumulation starts again in 2010. In the initial run for all three stations the accumulation starts way too early, and the snow pack is thus greatly overestimated the whole season. Since the effect of the early accumulated snow that is actually melted out, the correction of precipitation will be too low as it considers the simulated SWE at 1. January and all snow that falls after that. This would therefore mean

Table 3: Backtracking of maximum SWE, initial runs. Only 2007-2008 was ran on both temporal resolutions

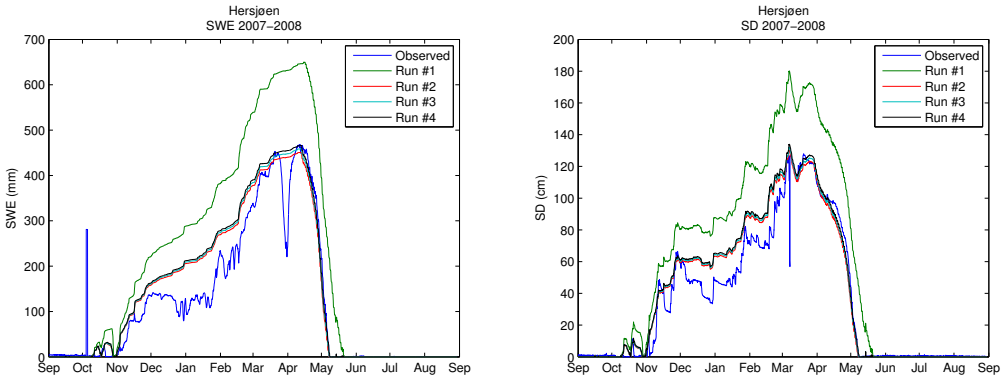
Station	Backtr. SWE (24 h) (mm)	Backtr. SWE (3 h) (mm)	Observed SWE (mm)	Corr. factor (3)
2007-2008				
Hersjøen	296.5	430.1	460.1	0.6756
Nessjøen	543.2	891.5	822.8	1.7094
Sørungen	344.2	446.7	453.0	0.6509
2008-2009				
Hersjøen	-	411.9	396.4	0.7370
Nessjøen	-	535.0	504.0	1.0951
Sørungen	-	658.4	618.7	1.1869
2009-2010				
Hersjøen	-	339.0	251.1	0.6857
Nessjøen	-	405.7	201.5	0.8671
Sørungen	-	194.5	192.6	0.4166

that all the falsely accumulated simulated snow per 1. January is considered to contribute to the seasonal maximum snow cover. Thus SWE and snow depth during the snow accumulation period in the first correction run is highly overestimated up until the maximum period. By the end of the accumulation the two values get closer to what is observed.

6.2.2. Iterative correction runs

For the stations where the model overestimated the SWE amount the backtracked SWE was too low, and visa versa. This might be a consequence of the fact that increased/lowered snow pack needs more/less energy to reach the output phase. For example a backtracking of snow-melt runoff from an underestimated snow cover will require less energy to reach the output phase than the actual, deeper snow cover. Thus more energy should have been used to further ripen the snow pack rather than produce runoff, and the backtracked SWE value will be too high. This is particularly important in cases where modeled and observed SWE values have large differences. Additionally, the simulated snow maximum point in time is also shifted. Because of this, three iterations of snow precipitation correction runs were run, and the maximum SWE backtracking and correction factors were assessed after each iteration. The idea is that the SWE values after one correction run will be closer to the observed values, but not optimal because of the above mentioned factors. Additionally, doing it iteratively will give some insight to how the correction would behave if a precipitation adjustment rate initially was better estimated, i.e. considering first correction run as initial run, and looking at whether the second correction run improves the backtracking.

The iterative backtracking results are listed in Table 4 and SWE and snow depth evolutions for all stations for all simulations years are plotted in Appendix A. After each run, a new set of correction factors were calculated and the model re-started using the original data and the product of the current and previous correction factor values, e.g. for Hersjøen, the second snow precipitation correction run used a correction factor of $0.6756 * 1.0195 = 0.6888$ (Table 4). The best fit with observed data is achieved in the 3^{rd} correction runs. Figure 12a and 12b shows the observed and simulated SWE and snow depth (SD) values from the iterative runs at Hersjøen



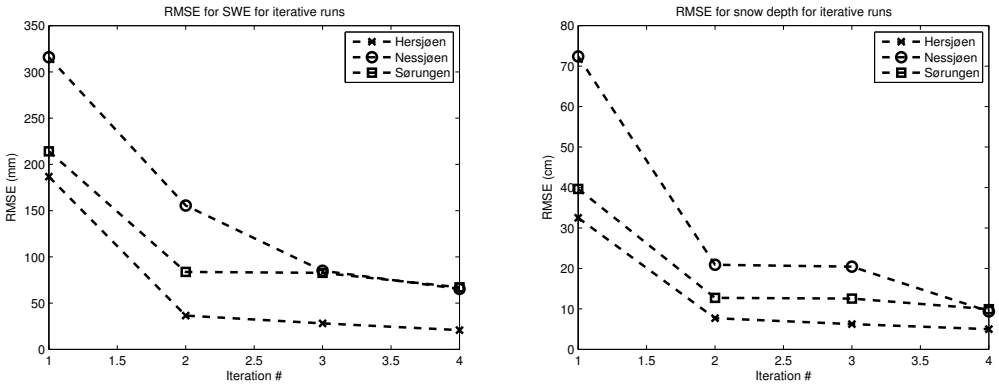
(a) (b)

Figure 12: Observed and simulated SWE (a) and snow depth (b) for iterative runs at Hersjøen 2007-2008

station for 2007-2008. This station at this year had the best fit of all backtrackings, with an error of 3.2 mm SWE in the 3^{rd} correction run, and it is evident from the figure that both the simulated SWE and SD closely follows the observed values from late accumulation season until the melt out at this station. When only considering observation and simulations that coincide with the onset of output from the snow pack, i.e. Hersjøen 2007-2008, Nessjøen 2007-2008 and 2008-2009, and Sørungen 2007-2008, the average error is 15.85 mm. The other backtrackings could not be assessed, as observed SWE values were most likely wrongly measured.

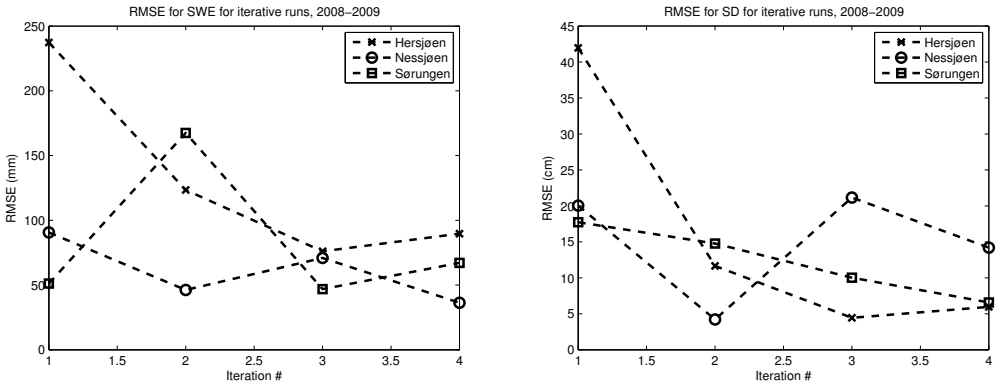
Figure 13 show the root mean square error (RMSE) for SWE and snow depth, respectively, for the three stations in the season 2007-2008, and it is evident that the first correction run improves the simulations a lot, while the following iterations continues to reduce the errors, but the improvements are much smaller. The RMSE for the seasons 2008-2009 and 2009-2010 (Figure 14 and 15) has no clearly present trend. For 2007-2008 the snow depth RMSE values are much lower, so that the variation looks larger than for 2007-2008, and since the initial error is low the correction does not improve, or increase during the iterations. The first correction run of Sørungen this year has a much higher RMSE than the initial run, but is brought down in the last two iterations. In the 2009-2010 simulations the SWE and snow depth simulations vary much at Sørungen, and the magnitude and the RMSE are high for the SWE values, probably due to erroneous SWE observations at SWE maximum.

When plotting the simulated values one can see that for most stations the main snow accumulation season starts too early compared to the measured SWE and snow depth. In some cases SnowModel, in contrast to the observed data, simulates a short accumulation and ablation period prior the main accumulation. At Hersjøen 2007-2008 (Figure 16a) the early accumulation/ablation events are never removed, thus influencing the total snow precipitation correction for this period as more snowfall events contribute to the maximum snow value. Figure 16a also illustrates how the iteration runs seem to converge toward a stable simulation after several correction runs. Additionally, when comparing these SWE values to the snow depth values (Figure 16b) it is evident that the errors in SWE are closely related to errors in simulated snow depth. In 2008-2009 the



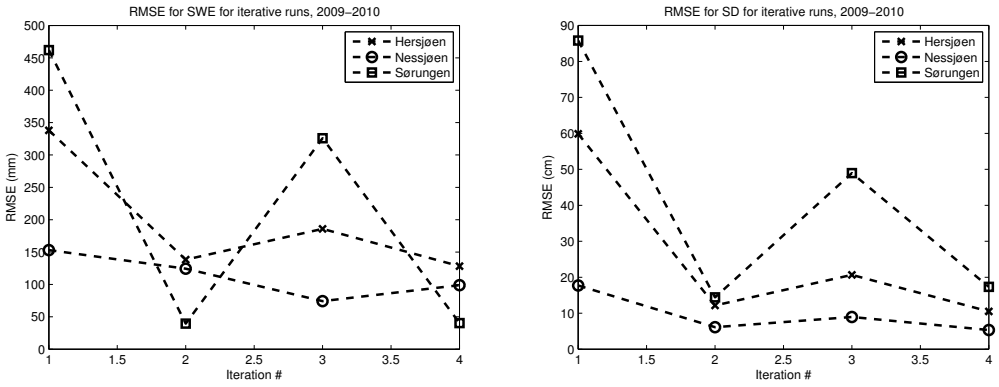
(a) SWE (b) SD

Figure 13: RMSE for SWE (a) and snow depth (b) for iterative correction runs 2007-2008. Run no. 1 is the initial run.



(a) SWE (b) SD

Figure 14: RMSE for SWE (a) and snow depth (b) for iterative correction runs 2008-2009. Run no. 1 is the initial run.



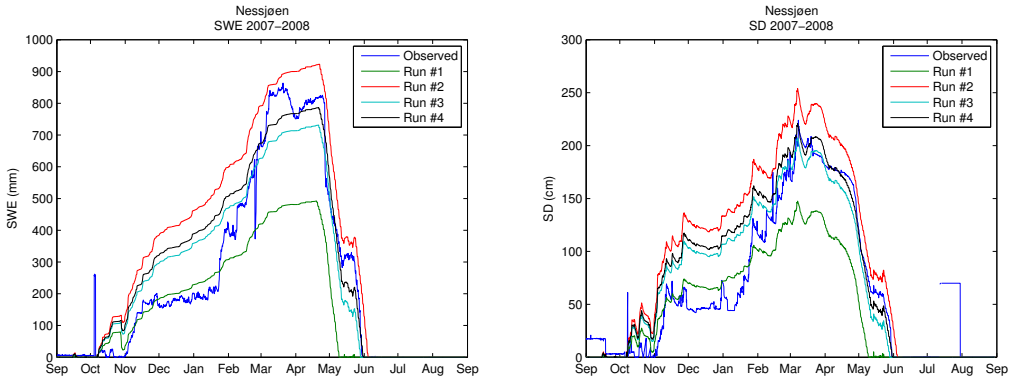
(a) SWE (b) SD

Figure 15: RMSE for SWE (a) and snow depth (b) for iterative correction runs 2009-2010. Run no. 1 is the initial run.

Table 4: Backtracking of maximum SWE, iterative snow precipitations correction factor runs

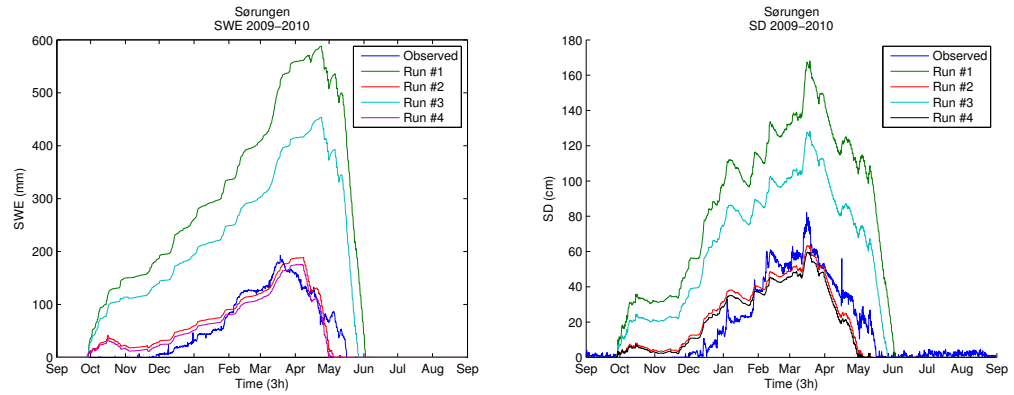
Season Run no.	Station	Back.tr. SWE (mm)	Observed values	New corr. fac. (mm)
2007-2008				
2	Hersjøen	449.9	460.1	1.0195
2	Nessjøen	747.0	822.8	0.8131
2	Sørungen	471.2	453.0	1.0036
2008-2009				
2	Hersjøen	387.2	396.4	0.9099
2	Nessjøen	454.7	504.0	0.8506
2	Sørungen	560.2	618.7	0.8448
2009-2010				
2	Hersjøen	391.4	251.1	1.0687
2	Nessjøen	379.4	201.5	0.9147
2	Sørungen	351.1	192.6	1.7642
2007-2008				
3	Hersjøen	456.9	460.1	1.0173
3	Nessjøen	796.0	822.8	1.0661
3	Sørungen	472.7	453.0	1.0633
2008-2009				
3	Hersjøen	401.8	396.4	1.0290
3	Nessjøen	490.3	504.0	1.0636
3	Sørungen	582.5	618.7	1.0347
2009-2010				
3	Hersjøen	363.3	251.1	1.0687
3	Nessjøen	400.7	201.5	0.9147
3	Sørungen	219.2	192.6	1.7642

modeled accumulation starts too early at all three stations, and in the last season (2009-2010) the start of the accumulation period is far too early compared to the observations. The fact that the model starts the main accumulation too early makes the snow precipitation correction factors wrong, as more precipitation is considered to fall as snow and contribute to the SWE, rather than as rain. This will in turn make the snow accumulations wrong as all snow precipitation events will be scaled down or up, so although the model forces the data through the assumed maximum SWE values from the backtracking, the snow evolution pre-melting season may not fit the observed data. All in all, the correction runs does not seem to improve the accumulation period too much in most cases, but the maximum snow amount and melting season is always improved. Also notable is the simulation for Sørungen 2009-2010, where the correction factors are never able to fit the simulations (Figure 17a and 17a) to the observed values, even after several correction iterations.



(a) (b)

Figure 16: Observed and simulated SWE (a) and snow depth (b) for iterative runs at Nessjøen 2007-2008



(a) (b)

Figure 17: Observed and simulated SWE (a) and snow depth (b) for iterative runs at Sørungen 2009-2010

6.2.3. Melt energy and ripening of the snow pack

Figures 18 and 19 shows simulated SWE, snow depth and accumulated melt at Hersjøen for the snow season of 2007-2008, both for the initial and 3^{rd} assimilation run. This is the station which gave the best fit after the assimilation for SWE and snow depth especially from the beginning of the melting season and to the melt out, meaning that the model as a whole works very well at this point at this time after correction. It is clear from the figures that some amount of the melting thats occurring in the beginning of the melting season (ripening phase) goes to reduce the snow depth and increase the snow density, rather than leaving the snow as runoff. As the snow depth and SWE values are very similar to the observed values (Figure 12a and 12b), one can assume that the interrelationship between melt, snow depth and SWE is accurately simulated. In Figure 18 and 19 the point in time where simulated SWE is at maximum (right vertical line) and the point in time where steady melting is starting, i.e. energy available to melt snow, (left vertical line), i.e. the ripening phase, is marked. In this period 159.6 and 129.6 mm MWE (Equation 5) is simulated for the initial and 3rd run respectively, and the total amount of MWE from the steady melt start to melt out is 890.1 and 616.5 mm MWE for the two simulation runs, thus the relative proportion of total MWE used to ripen the snow pack are 18% and 21% (only melt from the steady melting period is considered).

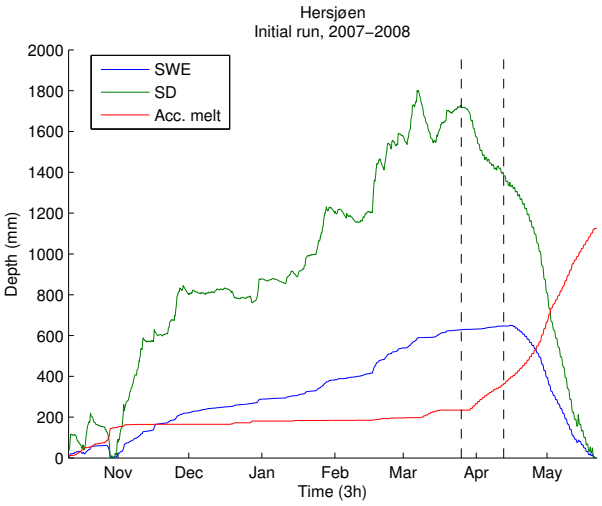


Figure 18: Initial simulation of SWE, snow depth (SD), and accumulated MWE (Acc. melt) for snow season at Hersjøen, 2007-2008

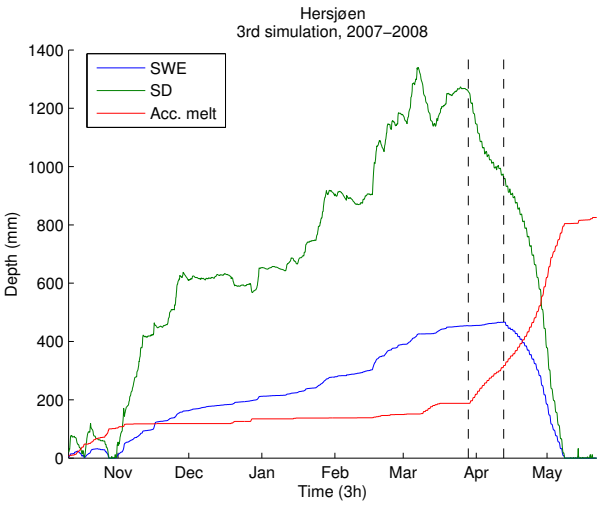


Figure 19: 3rd simulation of SWE, snow depth (SD), and accumulated MWE (Acc. melt) for snow season at Hersjøen, 2007-2008

6.3. Using SCA maps as zero SWE situation indicators

6.3.1. Spatial distribution of observations

Processing the SCA images gave snow free time observations and thus basis for calculation of maximum seasonal SWE and correction factors over a large portion of the study area for the 2008 melt season at different elevations. Figure 20 shows the distribution of observations over the study area. The most optimal situation would be to have one observation at each grid cell, giving a continuous correction field for the whole area. As this is very unlikely to achieve in the cloudy, coastal climate of Norway, some areas have a large temporal spread in observations. The lowest laying areas have few to no observations as the data used are from some time into the snow melting season, and all forested grid cells are filtered out. The exact location of the zero SWE observations will differ as a function of the SCA threshold used; in general the observations move higher up in the topography with increasing threshold. Histograms of the DEM height of the zero SWE observations found using a SCA threshold of 25% (Figure 21) show how the zero SWE observations move higher up through the season and that for all observation dates, the histograms have a tail towards right, i.e. right skewed distributions. These tails can very well be the result of areas that are wind swept and have a thinner layer of snow relative to the surrounding grid cells, and thus give more bare ground earlier in the melting season compared to the surrounding cells. Figure 22 shows details from the north eastern corner of the study area with zero SWE observations from 08. May, 07. June, and 04. July 2008. The observations closely follows the topography and reflects the snow line at each point in time. However, at more steeper, higher laying parts of the study area, as shown in Figure 23, there are many observations from the earlier date close to observations from later dates, giving correction factors as different as 0.1 and 0.9 in neighboring grid cells when processed. The histograms in Figure 21 also show this effect in that all observation dates have a tail towards higher elevations. It is hard to tell whether these early observations at this height are results of wrong classification due to complex recording geometry, or very wind swept areas with thin snow layers that are early melted. At any rate, although these observations may be "true" in terms of SCA value, the general correction factor at this area will be influenced by these outliers. If there are little to no other observations near these observations, they will give a non-representative correction factor for that whole area. As IDW interpolations are sensitive to outliers, these outliers will also increase the probability of "duck eggs"-effect in the resulting grid.

From the interpolated correction factor fields it is evident that the correction factors have a correlation with the topography, giving increasing values with increased height. This is not surprising as the flat precipitation interpolation does not take into account any precipitation correction by difference in height.

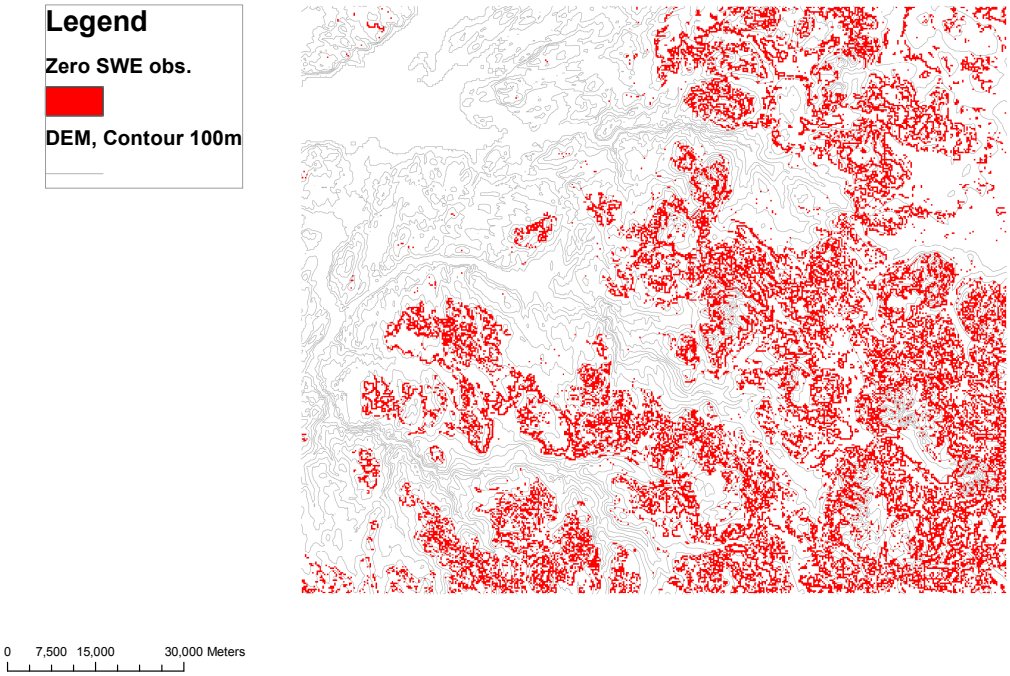


Figure 20: The spatial distribution of zero SWE observations extracted from MODIS SCA maps using a SCA threshold of 25% for the melting season of 2008

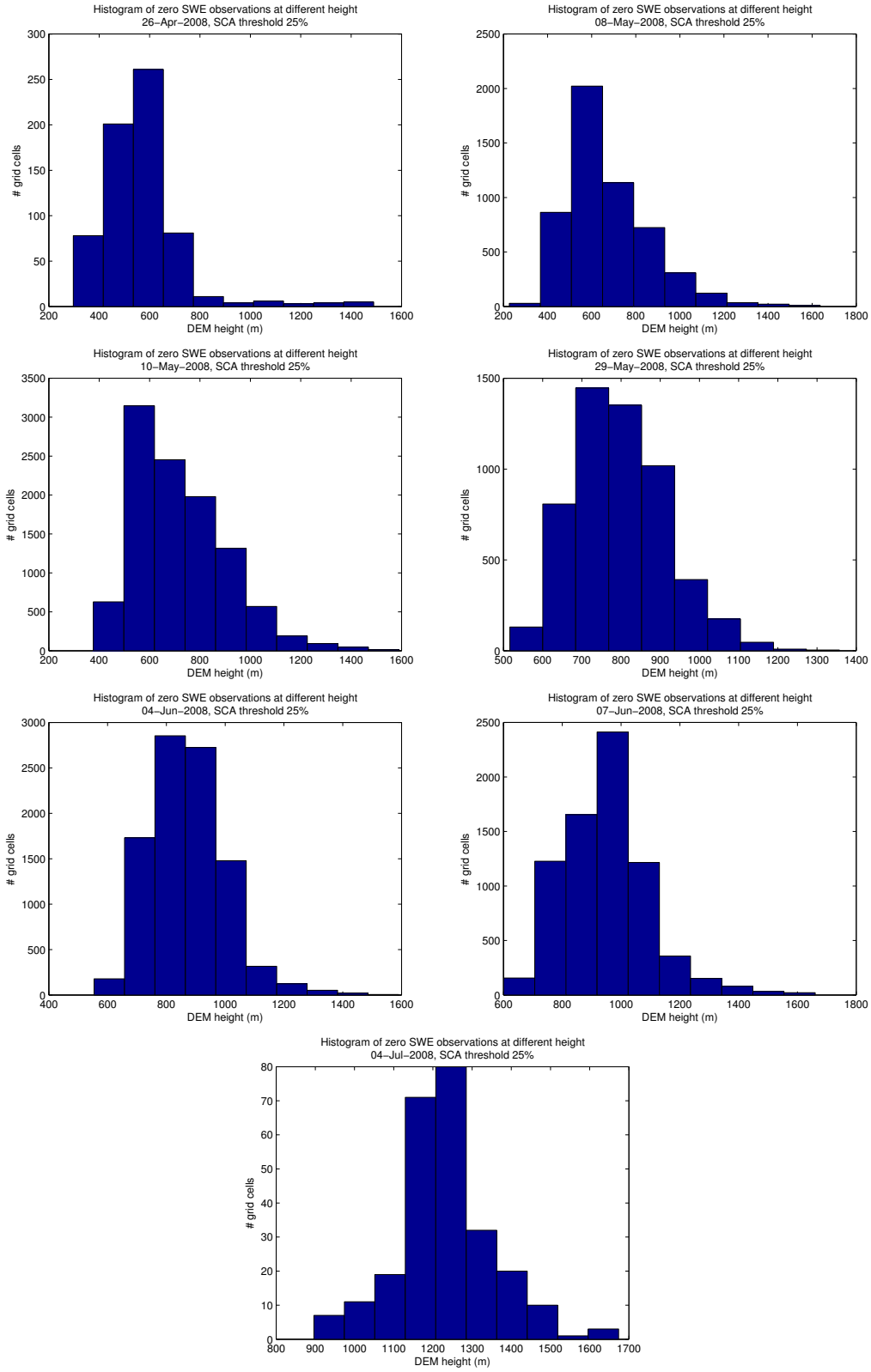


Figure 21: Histograms of zero SWE observations at elevations from SCA maps using 25% SCA threshold at different dates.

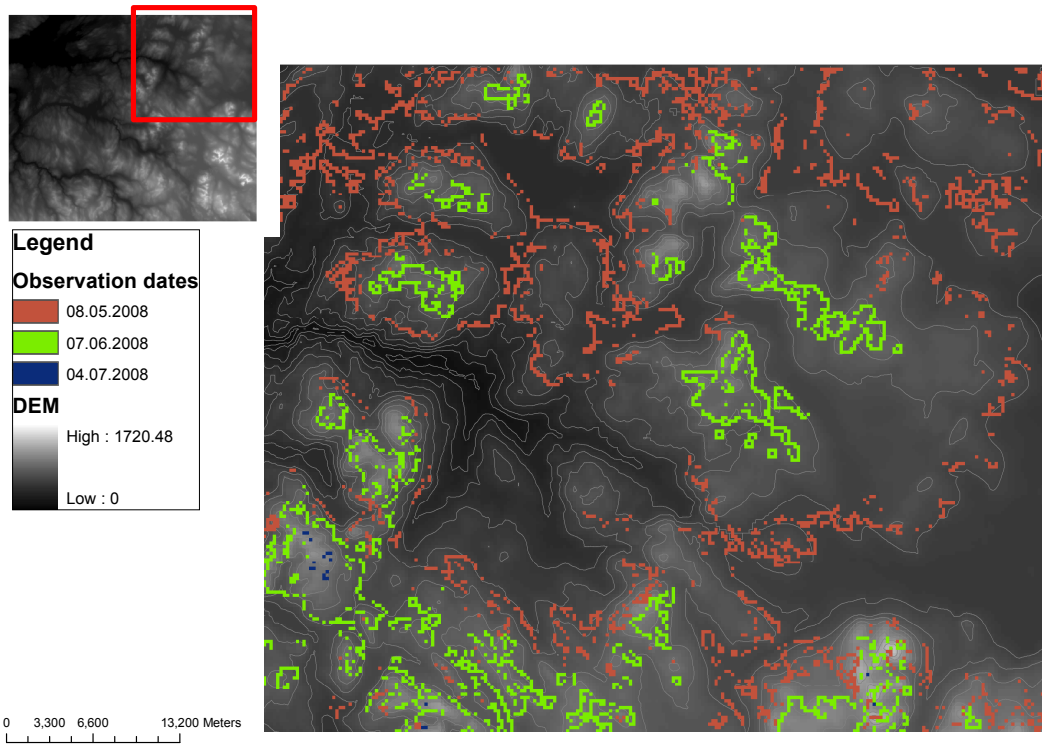


Figure 22: Zero SWE observations at three dates in 2008, in north eastern corner of study area, showing how the observations seem to follow the topography

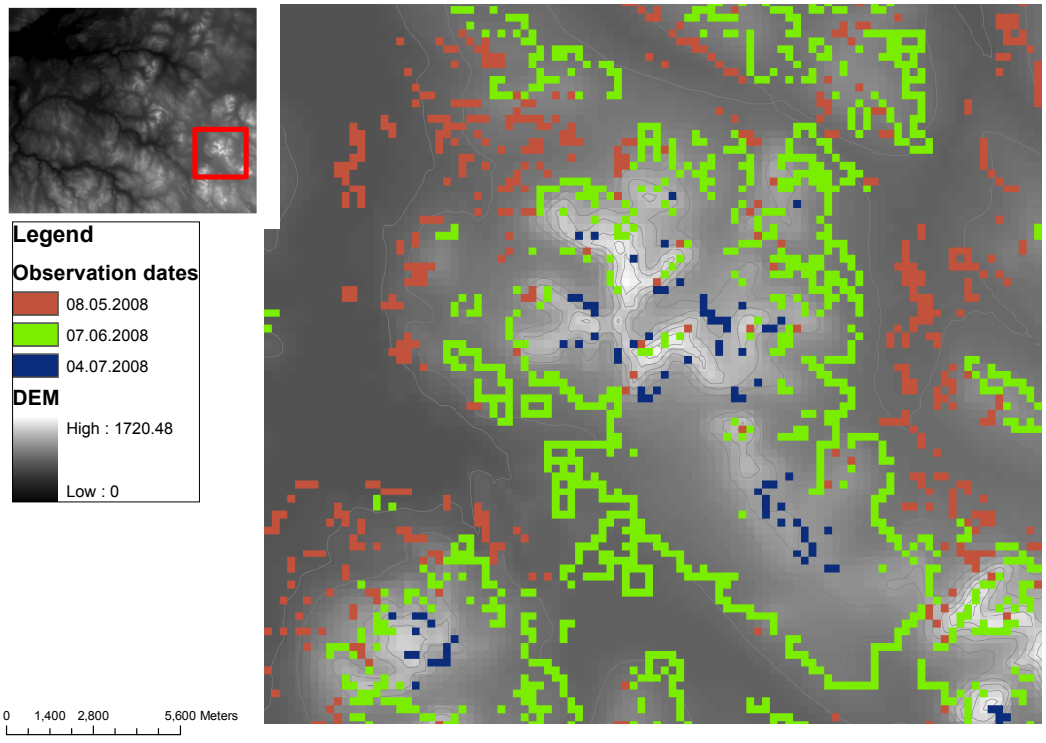


Figure 23: Zero SWE observations at three dates in 2008, in steep terrain showing how some high laying areas get zero SWE observations much earlier that neighboring grid cells

6.3.2. Gridding of correction factors

As shown in Section 6.2, in most cases the correction runs does not give a good fit for SWE and snow depth in the accumulation period, but the snow maximum and ablation season the fit is improved. A calculation and gridding of correlation factors will therefore force the model through the snow maximum value based on runoff and zero snow situation time, and thus possibly improve the simulated runoff values. Correction factors were calculated for all zero SWE observations found at the different MODIS image pairs and interpolated to cover the whole study area at different SCA threshold values. The number of observations were around 23,000 depending slightly on the SCA threshold used, which is a rather large amount of data to process. Several interpolation methods were tested for these datasets; the BOAS algorithm included in SnowModel gave some interference-like distortions in the field, and when doing a krieging analysis, the semi-variogram scatter did not seem to flatten out to a sill, even with trend removal, thus indicating that the dataset is not suited for these interpolation methods. Spline interpolation gave areas with negative correction factors, and were therefore not usable. IDW seemed to give local spikes and pits (so called "duck-eggs"), but it was possible to reduce these effects by interpolating to a grid of 1000 by 1000 m, doing a 3x3 window low pass (mean) filtering, and then re-sampling down to 250 by 250 grid using bilinear interpolation. The prediction errors of krieging and IDW were very similar. When considering processing time and accuracy, the IDW method seemed to be best suited for interpolation of the correction factors. Figure 24 shows the result of the interpolation of the correction factors derived from the observations in Figure 20 using the above mentioned IDW method and smoothing.

The interpolated correction factors were extracted for the grid cells that include the snow pillows and compared to the correction factors found in the test of concept study. The correction factors for Hersjøen, Nessjøen and Sørungen are shown in Table 5. It seems that for the grid cell covering Hersjøen the correction factor is varying randomly around 0.6, without any trend with threshold value, and the same goes for Sørunger, although the values varies more here. However, at Nessjøen the correction factor has a clear negative trend with the SCA threshold, decreasing as the threshold value increases. A threshold value of 0% seems to give much too high correction factors, and might be a result of noise in the SCA maps. When comparing these correction values to those found by using the zero time extracted from snow depth observations, the 10% threshold is closest. In this comparison there is a significant mismatch in scale, as the correction factors based on the snow pillow and/or depth radar is representative for an area of a few meters or less, the SCA data have spatial resolution of 250 by 250 m, so a zero SWE situation at the snow pillow is not necessarily true for the whole 250 by 250 m grid cell. Figure 25 is a cumulative plot of the correction factors at different SCA threshold values, showing the area fraction for the correction factors. As the threshold rises, the correction factors are lower. This is not surprising considering that a higher threshold would mean early melt out and therefore lower accumulated melt energy compared to the accumulated snow, giving lower correction factors.

The correction fields from the 0% and 5% thresholds gave some false observations in the lower laying areas late in the melting season, resulting in very large correction factors giving out-layers that creates erroneous correction fields, e.g. very high correction factors at sea level. These errors might come from uncertainties in the classification algorithm and/or noise in the MODIS images.

As the precipitation correction factor is set to zero for all the simulations and correction calcula-

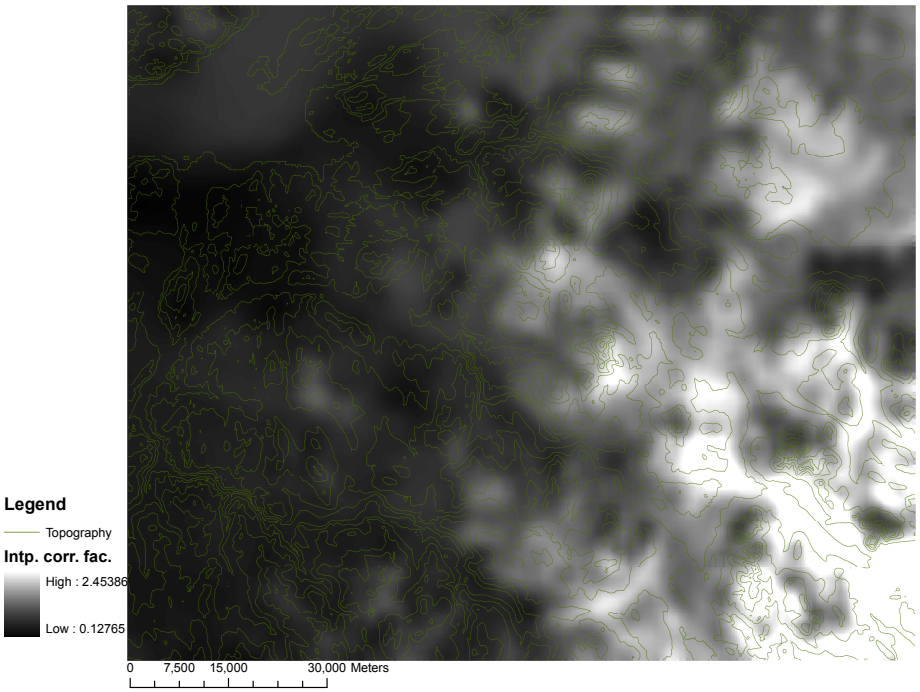


Figure 24: Map of interpolated correction factors from Figure 20 using IDW and smoothing, melting season 2008.

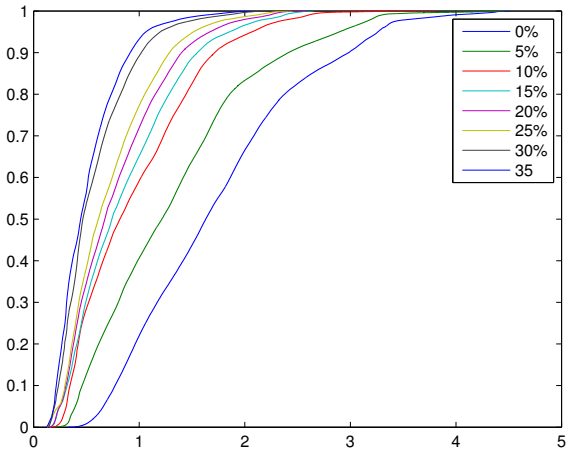
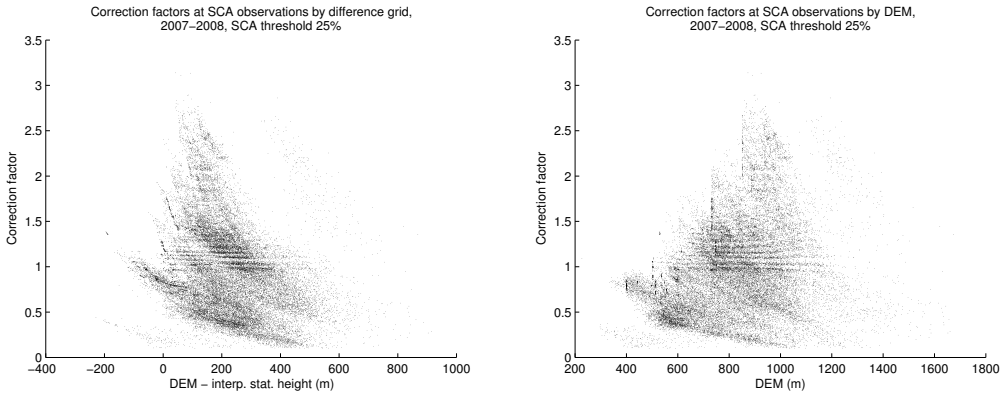


Figure 25: Area fraction of interpolated correction factors for study area at different SCA threshold values

Table 5: Correction factors at snow stations using SCA maps to determine zero SWE situation. Stn. row is values calculated using station observations to determine zero SWE in the initial run for the season 2007-2008

SCA threshold (% snow cover)	Hersjøen	Nessjøen	Sørungen
Stn.	0.6756	1.7094	0.6509
0	0.9876	2.0344	0.8717
5	0.6116	1.8937	0.4358
10	0.6222	1.7114	0.3701
15	0.5784	1.6220	0.3001
20	0.6006	1.4459	0.3703
25	0.6047	1.2820	0.3599

tions, it is expected that the distribution of correction factors should have a linear relationship with the difference between interpolated precipitation station elevation and the DEM, assuming that all other factors are correctly distributed and modeled. Figure 26a and 26b shows scatter-plots of the correction factors and the height differences for pre-interpolation and interpolated values, respectively. The scatter clouds are very similar for all threshold values, except for the spread in magnitude in the correction factors, which is higher for lower SCA thresholds and lower for higher thresholds (also evident in Figure 25). The clouds show that the highest correction factors are found at around 150-200 m difference. Neither the observation points nor the interpolated data show any linearity with the difference in height (DEM - interpolated station height). Figure 27 and 28 shows the same correction factor data plotted against the DEM topography only. In this case both the data sets show a positive relationship with the topography, and the interpolated data seem to show some correlation with the topography. In addition it was investigated if a multiple regression using x (Easting coordinate), y (Northing coordinate), and z (DEM height) could represent the field in a general way. However, since the NW corner of the study area have practically no coverage, it is doubtful that the resulting field is representative for the whole area. The multiple regression shows some correlation with x and y coordinates, but these may as well be results of the observation distribution and correlation between x, y, and z. When comparing the different correction grids fields for high areas in eastern and middle part of the study area (e.g. the highest portion of the Kjelstadfoss catchment), the correction factors are reduced quite significantly, due to a rather strong relationship with x-coordinates. It is therefore thought that, at least when trying to get the best representation of this particular catchment, either the interpolated correction factor field or the linear fit with DEM height will give the best results when comparing runoff values.



(a) (b)

Figure 26: Scatter plot of calculated correction factors against difference between DEM and interpolated station heights (a) and DEM heights (b) for season 1. Sept. 2007 to 31. Aug. 2008, using a SCA threshold of 25%

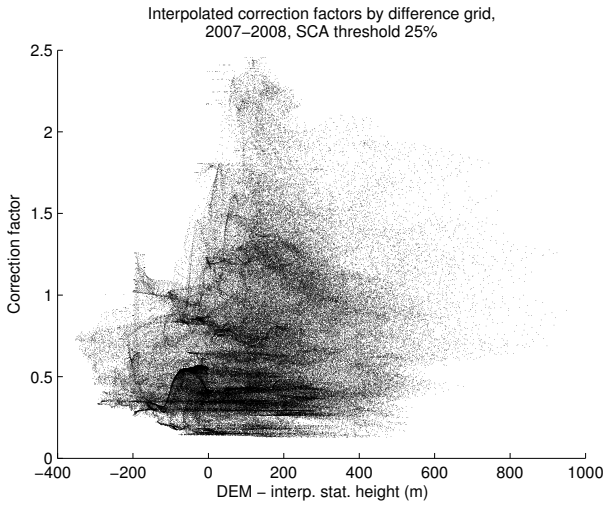


Figure 27: Scatter plot of interpolated correction factors against DEM values and interpolated station heights for season 1. Sept. 2007 to 31. Aug. 2008, using a SCA threshold of 25%

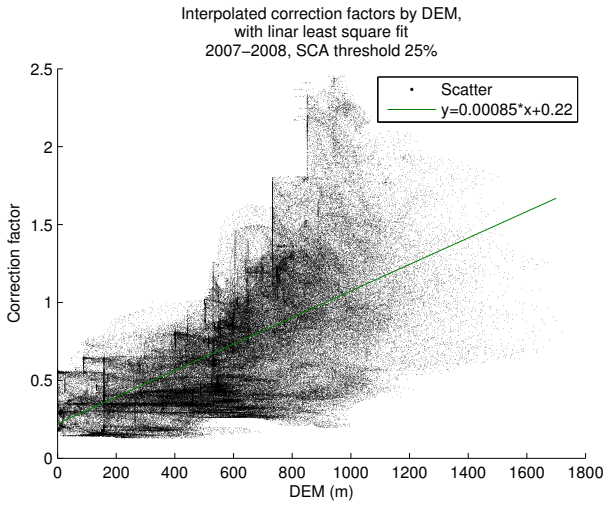


Figure 28: Scatter plot of interpolated correction factors, interpolated for the whole study area, against DEM values and interpolated station heights for season 1. Sept. 2007 to 31. Aug. 2008, using a SCA threshold of 25%

6.3.3. Validation using runoff

Figure 29 shows the observed runoff at Kjelstadfoss for the period 1. September 2007 to 31. August 2008 and the simulated runoff from SnowPack for an initial run for the same catchment (delineated from 250 by 250 m DEM) 3 hour resolution aggregated to daily values. It is evident that the runoff produced outside the snow season (due to rain) does not fit well with the observed values. Additionally, the observed runoff during the snow accumulation period shows several episodes of high runoff values that does not correlate with the runoff from SnowModel. This is most likely errors in measurements due to ice in the measurement station, since these high runoff events occur during the coldest period of the year with little to no rainfall or snow melt. Considering the snow melt season, the model seem to catch most of the melt episodes, but the magnitude of the melt is in general too large. It was decided to focus on the runoff season, as this is the time where snow melt has the most influence on runoff. The runoff produced for simulations using correction fields calculated from some of the threshold values in Figure 25 are plotted in Figure 30, and shows that for all runs with thresholds below 30% the simulated runoff is larger than the observed values the whole snow melt season, but somewhat reduced as the threshold is increased. For SCA thresholds of 30% and higher some of the melting episodes are underestimated. Assuming the melting energy to be accurately modeled over the whole catchment and that all snow melt goes directly to runoff in the stream, this means that the simulated amount of snow is too large in the catchment for all correction factors lower than 30%. As shown in Section 6.2, the correction factors based on the initial runs might not be accurate due to large initial errors in amount of snow, and iterative correction runs are sometimes needed. Tests show that doing a second correction run with the interpolated correction fields would reduce the runoff values, similar to when the SCA threshold values were raised.

Figure 28 showed that it seems to be some relationship between the topographic height and correction factor. It is possible that the fitted line may give some insight to what the precipitation

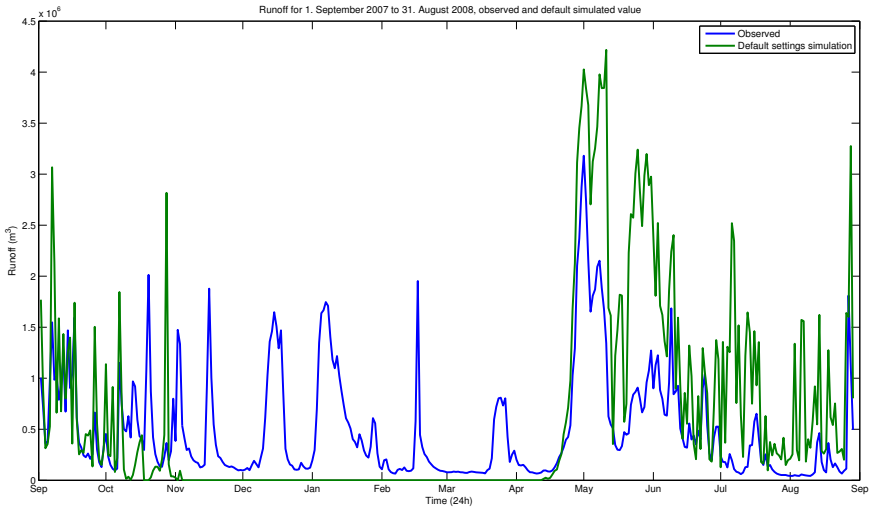


Figure 29: Runoff 1. September 2007 to 31. August 2008, observed values and simulated values using default settings and no precipitation correction

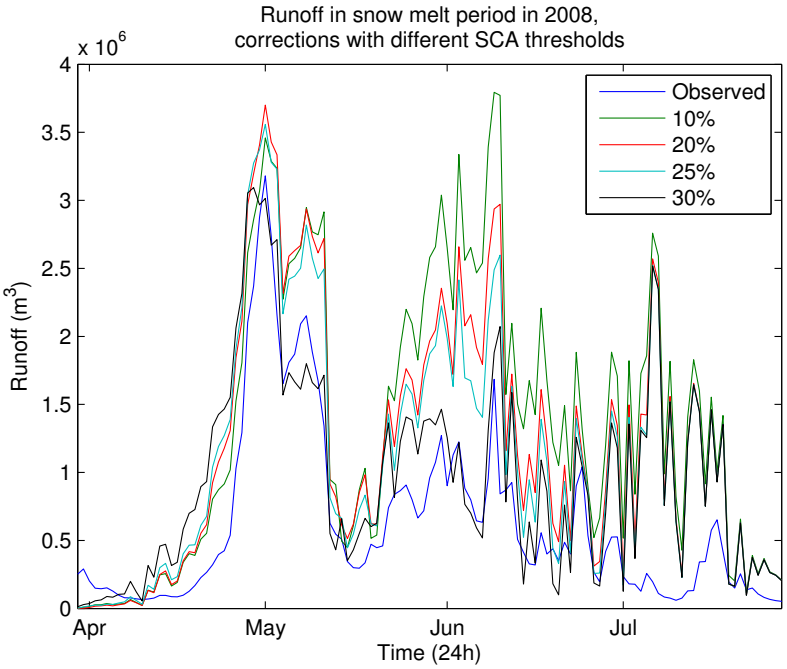


Figure 30: Runoff from simulations using correction fields found from different SCA threshold values

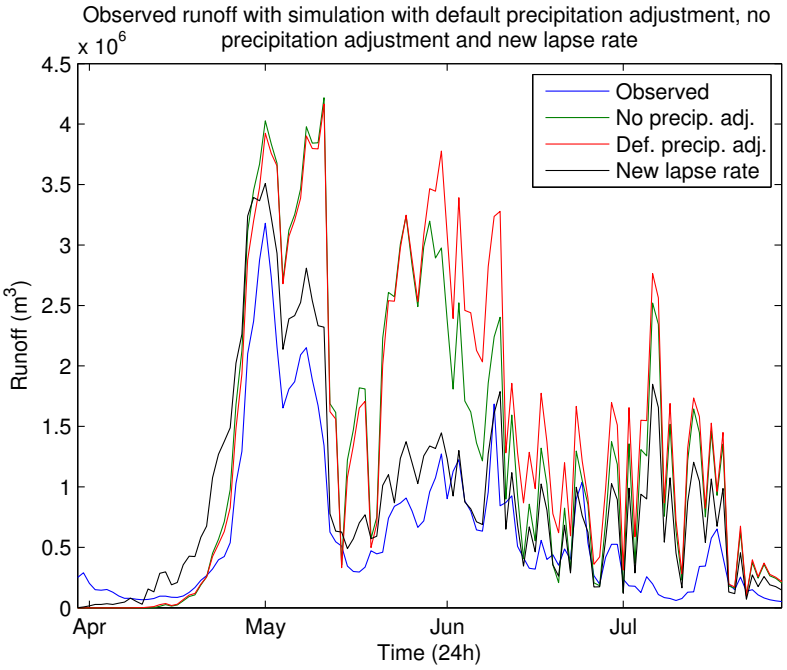


Figure 31: Runoff from simulations using default precipitation correction, no precipitation correction, and using the linear fit from interpolated SCA based corrections (Figure 28)

adjustment factor should be close to for snow/winter precipitation. Since the simulation using the correction factors derived from the 25% SCA threshold observations seem to give the best correction without underestimating any of the runs (Figure 30), this threshold was used in order to determine a linear precipitation lapse rate. The linear fit from Figure 28 gives a function for precipitation correction, P_{corr},

$$P_{corr} = 0.00085 * DEM + 0.022, \tag{15}$$

where DEM is the topographic height at a grid cell. Figure 31 shows the runoff for the melting season at Kjelstadfoss for simulation with the default precipitation adjustment factors (2.1) and methodology used in SnowModel, with no precipitation correction, and a simulation using Equation 15. It is evident that while the two first simulations overestimates the runoff quite significantly, the last run fits the observed runoff much better at most melting events. The new adjustments does however give more runoff in the beginning of the melt seasons. At any rate, the snow distribution is much closer to what the runoff observations indicate.

Using the results from the new lapse rate simulation, a new correction field was calculated. The resulting correction run show that the two first peaks in the runoff have a better fit, but the melt events in late May and early June is more overestimated (Figure 32), meaning that the the correction run estimates too much snow in the higher area of the catchment. All in all, the correction run does not seem to improve the simulated runoff.

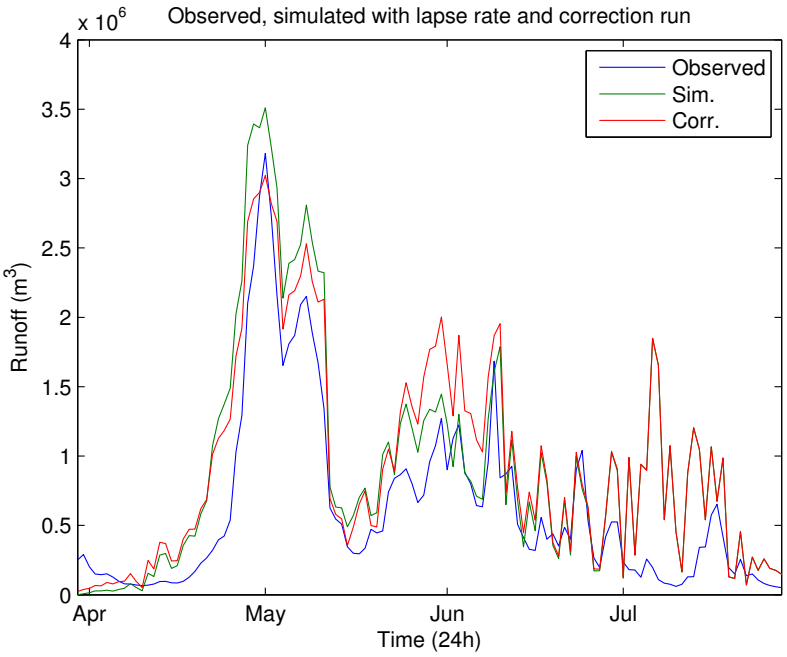


Figure 32: Observed and simulated runoff as in Figure 28 with runoff from correction run (25% SCA threshold)

7. Discussion

7.1. Test of concept

As a test of concept, SnowModel simulation results for grid cells that include automatic SWE and snow depth observations were extracted and compared with the observed data. The initial SWE backtracking assessment showed a large difference between the two different temporal resolutions (24 and 3 hours) tested. For daily time steps (24 h) the backtracked SWE missed with as much as several hundred millimeters of water equivalent, while the three hour resolution runs gave values much closer to those from the snow pillows. This indicates that SnowModel, as of this version, is not able to correctly mimic the physical processes governing snow melt at such a coarse time resolution. It is probably that the model fails to replicate the snow melt due to the fact that a daily resolution will not catch the diurnal cycles of meteorologic variables, e.g. high day temperatures gets smoothened by low night temperatures giving a steady, but too low, snow melt over the twenty-four hours time step, while at a finer temporal resolution a more intense melt is simulated during the day. Taking into consideration how much better fit backtracking the 3 hour simulations gave than the daily simulation, it is imperative to catch the diurnal cycles in order to realistically simulate the melting of the snow pack when using the current versions of EnBal and SnowPack.

7.1.1. Backtracking seasonal maximum SWE with point observations

Backtracking the maximum SWE value using the 3 hour simulation, seemed to give a good maximum SWE value compared to the observed values; the backtracked values are much closer to the

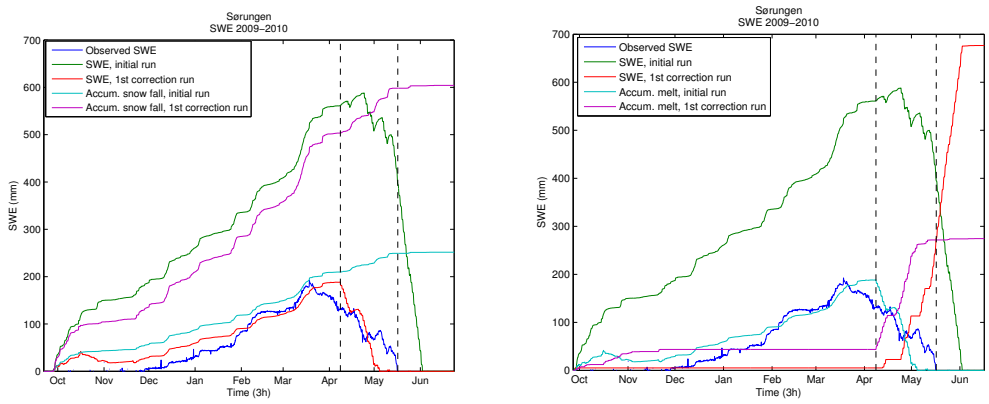
observed maximum values that the initial runs both with and without the default precipitation correction. Basing the backtracking on the initial runs seemed to over- and under-estimate the SWE value according to whether the initial simulation under- or over-estimated the snow cover, and iterative correction runs reduced the RMSE in general, although the largest improvement was achieved in the first correction run.

For the hydrologic years 2007-2010 several correction run iterations were done in order to try to minimize the error caused by different amounts of energy needed to ripen the snow pack. Table 4 shows that the backtracked SWE values and snow precipitation correction factors seem to stabilize quickly for both Hersjøen and Sørungen in 2007-2009; both stations have correction factors close to 1.0 already after the second run. Simulations at Nessjøen station needed a few runs to get the correction factor close to 1.0, and even after the third run it still indicates need for more correction. Since zero SWE situation is reached almost simultaneously for the last correction run and the observed data, it seems that the sum of melt runoff from the simulation is too small to replicate the SWE/snow depth. The melting season at this station also comes to a halt after an intense period of melt, with precipitation adding snow to the snow pack before the melting continues. The errors might therefore be a consequence of the fact that the model does not handle periods where there is some uncertainty in whether the precipitation is solid or liquid (i.e. around 0°C), and is discussed later in this section. Figure 37 and 38 show that a very good fit was achieved for Herjsøen, when considering only the late part of the accumulation and melting period, both for SWE and snow depth and is the simulation that fits best with the observed data.

The results of the iterative runs shows that for all runs the snow depth is overestimated until the melting begins, while the simulated SWE is eventually overestimated compared to the observations. In the melting phase, the snow depth of the simulations fit well with the observations, raising doubt to the accuracy of the SWE observations. As there seem to be a close relationship between snow depth and SWE error (e.g. Figure 16a and 16b), it might be possible to assimilate the snow depth observations, as these data are often very accurate, to correct for the errors in SWE during the melting season, assuming constant snow density.

7.1.2. Erroneous snow accumulation and correction factors

For all three stations the SWE and snow depth evolution during the very beginning of winter and the majority of the accumulation period the simulations seem to over-estimate the values. For the errors in the beginning of the seasons, it might be that some parameters needs to be changed in order to fully capture the snowfall-melt cycles. For all runs the SWE values were too high in this period, and at Nessjøen 2007-2008 the early-accumulated snow is never entirely melted away as the observations suggests, which influences the rest of the simulation period. This will in turn affect the precipitation correction factors that are based on SWE from 1. January and snow precipitation from after that date. This problem is also very prominent in the simulations for 2009-2010, where the snow accumulation starts much later, and maximum snow depth is much thinner than the previous years. In this season, the model starts the accumulation much too early compared to the observations, resulting in correction factors that are not suited to correct the snowfall events that actually contribute to the maximum SWE value. For Sørungen in the 2009-2010 season (Figure 17a and 17b) the correction factor is never able to fit the model to the observations even for the



(a) SWE

(b) SD

Figure 33: Accumulated snow fall (a) and accumulated runoff (b) with simulated and observed SWE at Sørungen, 2009-2010, clearly showing how the correction run greatly scales down all snowfall events and increase the melt. Vertical markers on correction run maximum snow and observed melt out.

melting season. After close investigation of the simulated values, it is evident that because the simulated snow accumulation starts much earlier than what is observed, the total amount of snow is much larger and thus the correction factor very low. As the correction factor is applied to all snowfall events, the total amount of snow is scaled down, but each event is much more reduced than what is realistic. This will, for the correction run, have a great impact in that snowfall events during the melting season are much more reduced. This will add less new snow to the snow pack, and thus it is faster ripened and more energy is used to melt away the existing snow. In addition, since less snow is added to the snow pack during melt out, the snow pack melts out faster. In the initial run, the simulated snow fall after the onset of the melting season is about 100 mm, but after the correction run it is reduced to about 40 mm for the same period, and the difference is very evident when plotted (Figure 33a). Figure 33b shows the runoff and simulated and observed SWE at Sørungen, 2009-2010, and it is evident that the runoff is much higher for the same time window in the correction run. This explains why the corrections does not converge towards the right values. The error is mainly due to the early simulated accumulation start, and seems to be a re-occurring problem for all simulation years, but is especially evident at Sørungen, where there are many snowfall events after the maximum snow peak. The other stations have poor SWE observations at the snow maximum for this season, and could not be assessed. It should also be mentioned that the correction runs use a flat correction factor through the whole snow season, and that the simulations do not use any precipitation adjustment, so that temporal changes in precipitation lapse rate are also not considered.

Many of the simulations and results previously discussed indicate that some important parameters in the late autumn and early winter season are not accurately parametrized, making the model not able to simulate the processes correctly. Figure 34 shows a snow pack accumulation and ablation early in the season at Nessjøen in 2009-2010. It is evident that, although the accumulation of snow in terms of SWE and snow depth fits somewhat with the observed values in the beginning, the model does not catch the melt out of this snow pack. The increasing error in the

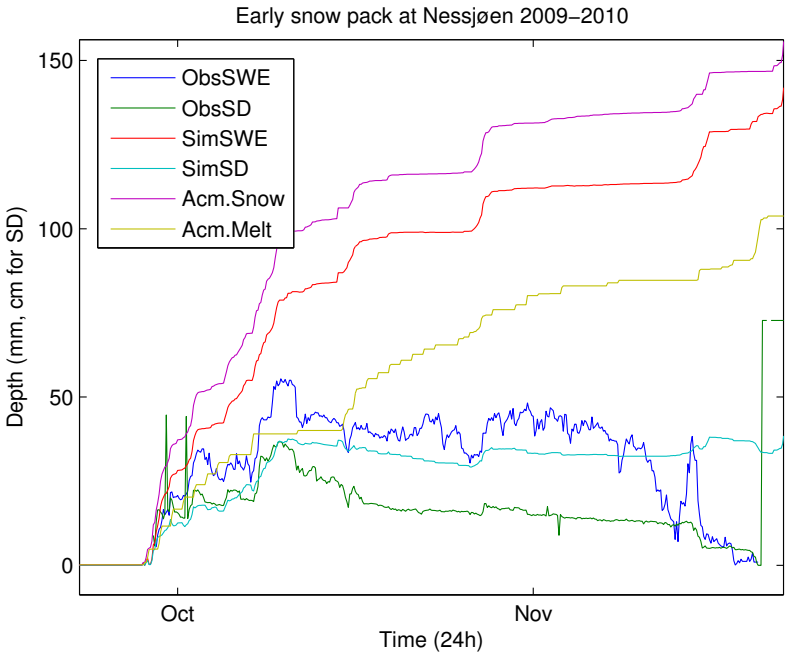


Figure 34: Detail from Nessjøen simulation 2008-2010 showing an early snow cover that is completely melted away in reality, but the model just continues to accumulate more snow rather than melt.

accumulation is mostly due to error in the precipitation adjustment factor. After the observed SWE peak is reached the simulation continues to accumulate the snow further, and the melt out in the end of the season is not caught at all. Judging from the simulated SWE there is a lot of precipitation at the same time as the observed snow melts out (Figure 34). It is therefore likely that the threshold temperature determining whether precipitation falls as rain or snow (2°C) is too high, adding precipitation that is actually falling as rain to the snow pack as snow instead. In addition, the little melt that occurs during this period is too small to remove the snow pack, in fact there is no runoff produced. In cases where precipitation is added as snow instead of rain, it might be possible to correct the precipitation state using snow depth measurements. One can for example look at the change in snow depth in the same period as there is precipitation at temperatures between 0 and 2 °C.

It is certain that the corrections will be better if one is able to determine the point in time that the main accumulation period starts, forcing the model to simulate a zero SWE situation here and do the correction factor calculations based on this period rather than just looking at the time from 1. January to melt out. It might be possible to extract these accumulation start observations from the MODIS based SCA or other remotely sensed images if one is able to get images with relatively short time between the recording times, e.g. a few of days, and do a similar analysis as the one done in this study using situations when the observed SCA goes from a $SCA = 0$ to a stable $SCA > 0$ state instead.

7.1.3. On energy used on ripening the snow pack

As shown in the example at Hersjøen 2007-2008 (Section 6.2.3), the amount of melt used to ripen the snow pack increase with the amount of SWE, the error in not considering this factor will be small for thin snow packs or for snow packs where the snow pack is already much compacted before the onset of the main melting season, but for the example of Hersjøen, including all melt in both the ripening and output period would give a backtracked SWE of 1.3 times that found by only considering the output phase. This shows the importance of including a sub model that handles the snow depth/snow density.

The energy required to ripen a snow pack before the output phase begins increases with snow pack thickness and this amount of melt will lead to an overestimation of maximum SWE value that increases with snow pack depth. It is not trivial to estimate the amount of pre-output melt needed as it depends on snow pack depth, which evolves during the snow season due to gravitational compaction and, in some climates, short melting episodes during the accumulation period.

7.1.4. On the observed snow data

The most prominent features in the snow depth evolution, which is the most reliable of the two observations, seem to be caught by the model and the values eventually fit, most notably when the ripening and output phase starts. In the accumulation period it seems to be a common error that the model does not catch many drops in the snow depth. This might also be an error caused by error in precipitation and the correction factor used. As most of the uncertainties in these simulations seem to lay in the accumulation period of the snow model, further studies must be done to better understand which factors and parameterizations that influence this phase of the snow pack in SnowModel. Additionally, transport by wind is most likely of importance and should be included in future studies involving snow pack accumulation simulations, especially for high spatial resolution runs.

The simulated SWE evolution during the accumulation period does not fit too well with the observed data, in some years not even at the SWE maximum and melting season. This might very well be due to melting episodes or other events that may create ice lenses or strong layers that can take the weight off area above the snow pillow, or increase the pressure by relieving the weight of surrounding snow. This is a common problem when using snow pillows, as mentioned by e.g. Lundberg et al. (2010). These layers might break, thus giving unrealistic increase in SWE, or increasing the SWE if supporting surrounding snow. There are several periods that seem to fit with this explanations, i.e. the observed increase of SWE and snow depth over a period of time give a density higher than water for the "new fallen" snow, and the model simulated much less snowfall and no melt or rain. Other suspicious cases are found where measured SWE and snow depth giving unrealistically low snow densities prior to the output phase. It is therefore hard to assess the model performance regarding backtracking of SWE for some periods and sometimes during the snow melt, but having snow depth measurements to compare to makes it possible to point out the most unrealistic events in the observed SWE evolution. For backtracking and validation purposes, the snow depth sensors seem to give much better and reliable data than the

snow pillows, but has a major drawback in that one is not able to provide maximum SWE for validation.

As a concept, backtracking seasonal maximum SWE using SnowModel and zero SWE date observations, seems promising as the backtracked values were close to the observed values. Most parameterizations in the melting part of the model seem to be rather good as the backtracked values are close to the observed, and the melting curves of the model seem to fit as well with the snow pillow observations after doing a correction run. A close investigation of the melting phase of the snow pack shows that the model, when run on a 3 hour temporal resolution simulates the variations of both SWE and snow pack on a high temporal resolution very well. The different dependencies in the snow evolution algorithm makes it hard to correctly backtrack to a good value using only an initial run when the model over- or under-estimates with a large amount of SWE during the accumulation period, seeing that the energy used to ripen the snow pack will be different for different thicknesses of the snow pack. A larger initial miss will give way to a larger error in the backtracking, but it seems to be possible to reduce this particular error by iteratively run the model with updated snow precipitation correction factors. In some cases the RMSE possibly showed that the values will converge towards an improved value using several iteration of correction runs as the SWE evolution seems to improve after a second correction run, but this approach will be a more time consuming process, as the simulation time will increase with n times, where n is the number of iterations to reach a stable SWE value. A simple estimate of how much more/less ripening energy that will be required for a thicker/thinner snow pack might give an even better first estimate of the maximum snow cover. Another weakness of this methodology is that precipitation as rain during the different phases are not considered and might therefore be a source of errors, since rainwater may speed up the ripening of the snow pack, and any retained rain is also added to the SWE. The methodology does not consider sublimation either, but analysis of the initial simulation showed that both of these parameters are of a smaller magnitude compared to those included in the algorithm, so the errors due to these simplifications are probably not too great.

Some processes are not accounted for by this version of SnowPack. Firstly, it does not handle any negative net energy fluxes, e.g. energy available to refreeze liquid snow in the snow pack. It does not keep account for free water content in the snow pack either, which makes it difficult to make calculations based on the total energy- and water-balance of the snow pack at all times. Some improvements to the SnowPack module might give better backtracking and correction maps for the area.

7.2. Correction using SCA maps

7.2.1. Temporal and spatial coverage using MODIS-Terra

The weather at the coast of Norway prevents satellite-borne optical remote sensing from providing daily images regularly due to frequent cloudy weather. Using a maximum time gap of three days between observations, only seven observation dates were usable through the whole snow melt season of 2008 to determine zero SWE situations. While the use of radar-based sensors might also give a SCA estimate, studies show that this method is sensitive to bare spots (Storvold et al., 2006)

and is therefore possibly not very usable for this kind of analysis, although it gives the opportunity to collect SCA during cloudy conditions. When filtering the registered zero SWE observation by requiring that they have at least one neighbor with SCA larger than the threshold value, one got observations that seemed to mark the snow line at that particular time, thus reducing the risk of registering zero SWE situation that occurs earlier than the last recording date when the time gap is larger than one day. It is possible that this moving window method will give zero SWE observation estimations when used on only one image only if all pixels with value lower than the threshold is processed using the above mentioned method, although this was not tested in this study.

Even thought the remotely sensed data could not give a continuous set of zero SWE observations, the spatial coverage was rather good, at least above the tree line. Values for lower laying areas were excluded, mainly due to uncertainties in SCA spectral unmixing for forested areas, resulting in poor data density for interpolation in the lowest regions and in the valley bottoms. Correction fields were created using the remotely sensed data and different SCA thresholds. It was shown that for higher threshold, the correction value spread was decreased and average value lowered, and is as expected because a higher threshold will give earlier zero SWE situation in time, and thus less accumulated melt energy and less precipitation is needed to force the model through the backtracked maximum SWE value. For thresholds lower than 10% many false observations occurred for the lower portions of the study area, giving unrealistically high correction factors. As shown in Section 6.3, some zero SWE observations were generated at high altitudes even early in the snow melting season. It is impossible to conclude whether these are real observations or misclassifications due to noise and/or complex recording geometry in the original MODIS image with the data available in this study. If they are real observations it would mean that these areas have significantly less snow accumulation relative to their neighboring grid cells, and might be a consequence of wind erosion or other means of snow transport. If the satellite images gave a continuous grid of zero SWE observations and assuming the early observations to be true, they could possibly give an estimate of whether the wind transport parameters in SnowModel are sensible for a simulation, if the SnowTran-3D sub-model was to be used. However, because the precipitation distribution is not well known and SnowModels default values does not work well for this area, calculating a precipitation factor at these grids will give values much lower than the surrounding observations, and in observations sparse regions they may greatly influence the interpolation results. A possible solution to this problem in terms of estimating the correction factor may be to restrict the zero SWE observations at each date according to the height of each observation grid cell, e.g. only grid cells with height within two standard deviations from the mean height of the observation grids at particular date.

Scatter plots of correction factors at observation points and topography or the topographic difference between DEM and interpolated station height showed that there is little linearity for the difference height (Figure 26a and 26b). This may be due to the distribution of meteorological stations; the stations that provide precipitation are more or less positioned along a north-west to south-east axis, giving interpolated precipitation and station fields with gradients along this axis (Figure 7 and 9). This gives poor representation of the north-eastern and south-western portions of the study area, and the interpolated values may not be too good at these parts. The

interpolated correction factor scatter showed some linearity with the DEM heights, though with a large spread from the linear fit (Figure 28).

7.2.2. Validation using runoff data

As there were no observation data to directly assess the quality of the backtracked SWE or the precipitation correction factors, the runoff produced from the SnowPack sub-model was compared to measured runoff at Kjelstadfoss station. A large uncertainty in this comparison is obviously that the simulated runoff is from the bottom of the snow pack, while the measured runoff is affected by many other processes than just snow melt, i.e. evapotranspiration and recharge of soil- and groundwater. It is thought, however, that the two data sets might give some information on whether SnowModel simulates correct runoff during the melting season. The initial runs both with and without the precipitation correction factors gave runoff values much higher than the observations. When the correction fields derived from the SCA data were applied to the simulations, the runoff decreased for most melting events, and a threshold between 25% and 30% the model went from over- to underestimate the runoff at some instances. All the thresholds also gave a much earlier output start compared to the initial simulations and observed data, which might be correct as this water will probably go to recharge of soil- and groundwater. For the catchment of Kjelstadfoss, it seems that the best threshold of to separate snow and insignificant/no snow is between 25% and 30%. The high threshold may be due to uncertainties in the spectral unmixing, and some processes concerning the snow reflection properties might still be unaccounted for, or some parameterizations might not be representative for this area. The threshold is used globally, i.e. same threshold for all grid cells, but it is reason to believe that this threshold will vary across the study area as a function to topographic roughness at a sub-grid level and wind erosion and deposition. In order to better assess the relationship between SCA and SWE, more information at a finer scale is thought to be needed in addition to snow data that represents whole grid cells at the remote sensing product. One might possibly run SnowModel on a higher spatial resolution and compare the snow distribution with wind transport in the melting with the SCA observations from the MODIS images.

As discussed previously, the interpolated correction fields seem to be quite affected by outliers and has sparse observation density in the lowermost areas. Figure 35 shows the observation point around the Kjelstadfoss catchment, and it is evident that the data is not very representative for the lower regions of the catchment. None of the correction fields gave a very good fit with the observed data, either over- or underestimating melting events. The investigation of the scatter plots of the correction factors and topography showed that there might be some relationship between the two variables. Using the linear fit for the correction factors calculated from a 25% SCA threshold gave a runoff value that were much closer to the observed values for the whole period compared to simulations where the interpolated correction fields were used, although still slightly overestimated. Taking these results into consideration, it might seem that the interpolated correction fields are very sensitive to noise, melt out events that occurs earlier or later than the surrounding area, or in general outliers and gaps in the data. Using a more general correction based on the trend in the correction field, i.e. the linear lapse rate with topographic height, gave rather good correction. The improvement in runoff simulations when using the new lapse rate is most likely due to the fact

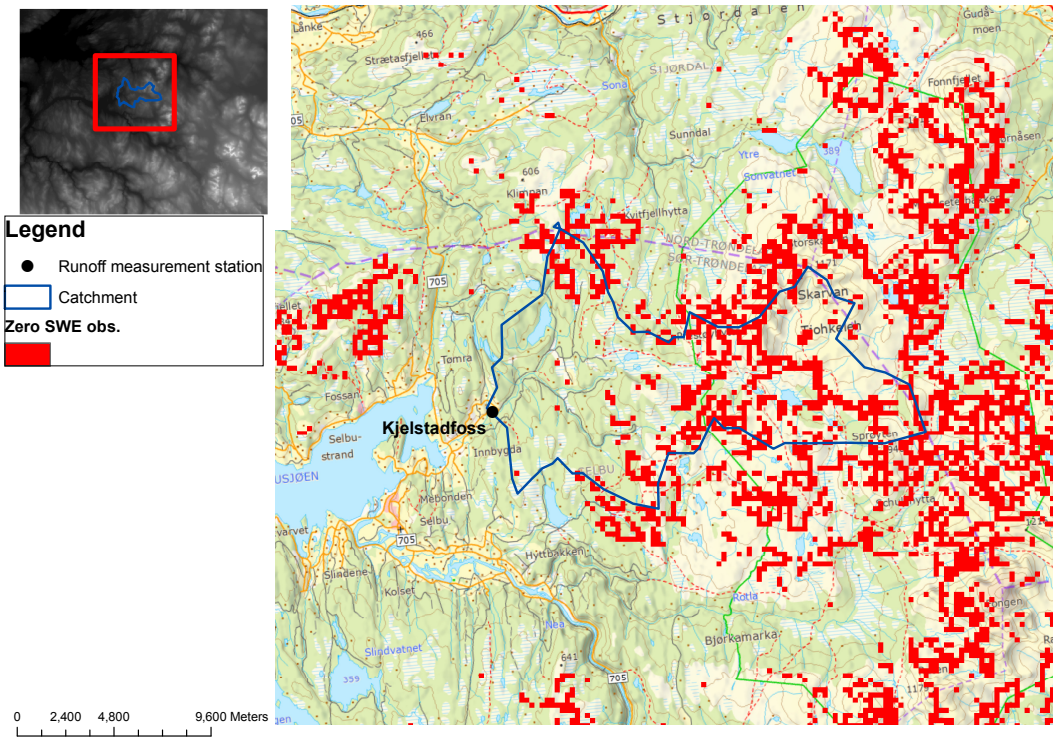


Figure 35: Zero SWE observations around catchment. Note the low density of observations in the lower most area of the catchment

that the initial/default precipitation correction factor is not representative for this climate/region, and thus gives a snow distributions that does not give the right amount of runoff when melting begins.

In all correction runs, the snow melt output seems to start too early compared to the observations. There are several reasons that may cause this; one is that the early runoff should be used to recharge soil moisture and/or ground water in a sub-model, rather than directly going to the river runoff. Another factor might be that due to the sparse observations at lower levels which will give large uncertainties in the interpolation, and the lower most grids are also those that are subjected to melting first. The fact that the runoff is mostly overestimated during the melt season may also be due to the fact that water is not removed by any means of evapotranspiration processes.

Another factor that may give high runoff values is the lack of redistribution of snow on sub-grid level. As the SCA-maps show, the snow covered area fraction is gradually reduced, and with a lower portion of the area covered in snow means lower surface that can produce melt water. In the case of SnowModel simulations, however, the whole area is considered snow covered, giving runoff according to the melting energy for the whole grid cell.

It should also be noted that the simulated runoff seem to be more off from the observed values the more the runoff is dominated by the precipitation than the snow melt, as shown in Figure 36. Here it is evident that as the snow melt goes towards zero as the snow disappears, the more mismatch there is between simulated and observed runoff.

Further assessment of the methodology may be done if one is able to use finer scale SCA or

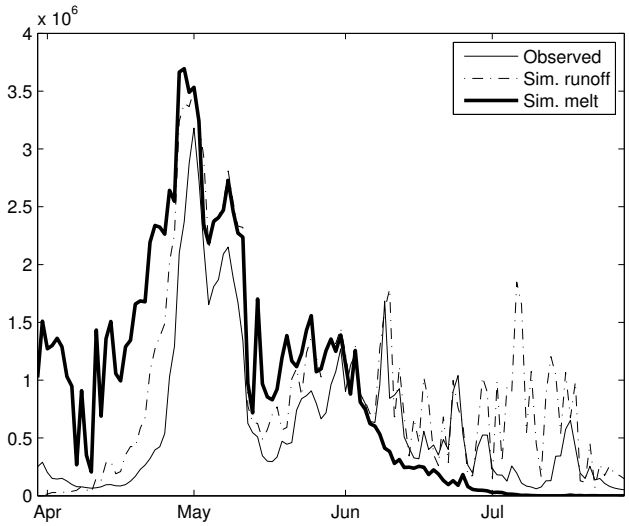


Figure 36: Observed and simulated runoff with simulated melt for run with linear lapse rate found from scatter (note: snow melt is either used to just compact snow pack or to give output)

snow/no snow binary maps that are more or less continuous in time. The temporal gaps in the study gave rise to spatial gaps in the zero SWE observations, and therefore high sensitivity to erroneous observations, outliers, uncertainties in interpolation, and possibly wind blown grid cells. If one does try this methodology at a fine spatial scale, it is most likely that wind transport and deposition will be much more important, and the sub-model SnowTran-3D must be used in addition to the modules used in this study. This will, however, probably introduce more uncertainties in the form of more parameters that must be considered to give a correct distribution of the snow. If spatially representative SWE data at snow maximum is available this could also be used to verify the backtracking methodology more correctly, but because of the coarse resolution of the simulations in this study it is very hard to gather such data. It could, however, be done on smaller scale simulations.

8. Conclusions

The concept of backtracking a maximum seasonal SWE and using a precipitation correction factor to force the model through these values were tested on both point and catchment scale observation of the point in time of a snow free condition. When calculating the backtracked SWE from point observation based zero snow situation the values got close to those from the snow pillows, although the snow pillow data were often subject to possible measuring errors when comparing to the observed snow depth evolution. Iterative correction runs seemed in some cases to improve both the snow parameter evolution for the whole melting season, but sometimes the best results were achieved after only one correction. The best maximum SWE result had an error of 3.2 mm compared to the snow pillow and the worst estimate, when ignoring erroneous snow pillow measurements, of a couple of centimeters of SWE. For most simulations the iteration runs were

able to closely simulate the snow depth, which is measured by a more reliable devise than snow pillows, indicating that the SWE is in fact close to what one might expect at times where the snow pillows fail to measure realistic values. It is though that it might be possible to assimilate the observed snow depth values during the melting season to correct for errors in SWE in the simulation, assuming a constant snow density.

From the simulations it is evident that SnowModel as of this implementation does not handle periods with temperatures around zero too well. It is thought that there is at least one major oversimplification that may cause these erroneousness snow pack evolutions, which is the constant 2°C threshold to determine solid or liquid precipitation. More work is needed to give a better estimate on this. The correction factors are therefore seldom representative, as the snow that is accumulated to reach the maximum SWE includes early snow events that are actually melted out before the main accumulation period. It might be very possible that the errors associated with these mild periods also influence the backtracking of maximum SWE to some degree, as snow precipitation is subtracted from the runoff in the melt out period.

Because of the weather systems at the study site, the optical remote sensing data were spread quite a lot in time, and thus the zero snow observations were spread in space. Since the forested areas were filtered out when detecting zero snow situations, the spatial coverage is very poor in the lower laying areas, and the corrections are mostly only representative above the tree line. Although the majority of observations based on the satellite images follow the topography, there is some noise in the form of early observation in the high mountains. This may be an effect of misclassification of SCA values or wind swept areas that are earlier melted bare. Simulations including wind transport (SnowTran-3D) and sub-grid distribution or finer spatial resolution is needed in order further investigate this effect.

Comparing the runoff from SnowModel for different SCA thresholds to observed runoff at Kjelstadfoss showed that the best threshold is between 25% and 30% (over- and underestimates runoff, respectively), which is higher than expected. This might be a consequence of uncertainties in the spectral unmixing when calculating the SCA from MODIS-Terra images. When investigating the relationship between topography and correction factor, it seemed to be no clear relationship with the topographic difference between DEM and interpolated station height, which is the variable SnowModel uses to adjust precipitation to topography. However, some relationship between correction factor and DEM height was found. Based on this relationship a simulation with a linear lapse rate was ran. From runoff comparisons it seemed that the correction fields were able to improve the runoff to some degree, but the best fit was achieved when only using the general trend. It is possible that the interpolation method (IDW) was not optimal and that noise and gaps in the observations gave sub-optimal correction fields. Since the point observation data simulations showed a good fit, it is thought that remotely sensed data with better spatial resolution combined with finer resolution simulation of SnowModel may give better correction fields. Additionally, more frequent remotely sensed observations are needed to better represent the area, where the optimal case is a zero snow observation at each grid cell in the model with no need for interpolation. A realistic solution for this is stationary digital cameras, which will give high temporal and spatial resolution, but at the cost of less spatial coverage.

In order to validate the SCA-based correction, the runoff from the snow pack was compared

to observed runoff in the Kjelstadfoss catchment. It is obvious that this comparison is crude as the liquid water output from SnowModel is not ran through any evapotranspiration or soil- and groundwater models. Therefore it is impossible to draw any final conclusions regarding the SCA threshold using MODIS-Terra based SCA maps to correct SnowModel retroactively before SnowModel is coupled with other hydrological modules, if one wants to validate against runoff data. At finer spatial scale resolution simulations, manual and automatic SWE observations may be used as verification.

As SnowModel is a physically based model, no calibration was needed to fit the model to the results, except the maximum snow density, and only the zero snow time was used to correct the model. As snow cover can be observed by most optical remote sensing techniques, from stationary cameras to satellite imagery, it has the potential for rather accurately estimate snow coverage in remote areas, as long as the meteorological gridding is accurate. Although the model does not preform well for precipitation in mild periods, it is thought that this method can be used to investigate maximum SWE distributions in retrospect for a snow season and also used to test response in snow cover parameters for different climate scenarios, as pointed out by Cline et al. (1998). It is important emphasize out that the accumulation periods were seldom close to the observed values, and the methodology is, as of now, only applicable for the snow maximum and ablation period.

References

Andersen, T. (1982). Operational snow mapping by satellites. In J. W. Glen (Ed.), *First Scientific General Assembly of the International Association of Hydrological Sciences (IAHS)*, Exeter, UK, pp. 350. International Association of Hydrological Sciences.

Anderson, E. A. (1976, Feb.). A point energy and mass balance model of a snow cover. Technical Report NWS-19, NOAA.

Barnes, S. L. (1964). A technique for maximizing details in numerical weather map analysis. *Journal of Applied Meteorology 3*(4), 396–409.

Bergstrom, S. (1995). The hbv model. In P. Singh Vijay (Ed.), *Computer models of watershed hydrology.*, pp. 433–476. Water Resources Publications Highlands Ranch CO United States.

Bruland, O., G. E. Liston, J. Vonk, K. Sand, and A. Killingtveit (2004). Modelling the snow distribution at two high arctic sites at svalbard, norway, and at an alpine site in central norway. *Nordic Hydrology 35*(3), 191–208.

Buck, A. L. (1981). New equations for computing vapor pressure and enhancement factor. *Journal of Applied Meteorology 20*, 1527–1532.

Cline, D., R. Bales, and J. Dozier (1998, MAY). Estimating the spatial distribution of snow in mountain basins using remote sensing and energy balance modeling. *WATER RESOURCES RESEARCH 34*(5), 1275–1285.

Iziomon, M. G., H. Mayer, and A. Matzarakis (2003). Downward atmospheric longwave irradiance under clear and cloudy skies: Measurement and parameterization. *Journal of Atmospheric and Solar-Terrestrial Physics 65*(10), 1107–1116.

Koch, S. E., M. Desjardins, and P. J. Kocin (1983). An interactive barnes objective map analysis scheme for use with satellite and conventional data. *Journal of Climate and Applied Meteorology 22*(9), 1487–1503.

Liston, G. (1999, OCT). Interrelationships among snow distribution, snowmelt, and snow cover depletion: Implications for atmospheric, hydrologic, and ecologic modeling. *JOURNAL OF APPLIED METEOROLOGY 38*(10), 1474–1487.

Liston, G. E. (1995). Local advection of momentum, heat, and moisture during the melt of patchy snow covers. *Journal of Applied Meteorology 34*(7), 1705–1715.

Liston, G. E. (2004). Representing subgrid snow cover heterogeneities in regional and global models. *Journal of Climate 17*(6), 1381–1397.

Liston, G. E. and K. Elder (2006a). A distributed snow-evolution modeling system (snowmodel). *Journal of Hydrometeorology 7*(6), 1259–1276.

Liston, G. E. and K. Elder (2006b). A meteorological distribution system for high-resolution terrestrial modeling (micromet). *Journal of Hydrometeorology 7*(2), 217–234.

Liston, G. E., R. B. Haehnel, M. Sturm, C. A. Hiemstra, S. Berezovskaya, and R. D. Tabler (2007). Instruments and methods simulating complex snow distributions in windy environments using snowtran-3d. *Journal of Glaciology 53*(181), 241–256.

Liston, G. E. and D. K. Hall (1995). An energy-balance model of lake-ice evolution. *Journal of Glaciology 41*(138), 373–382.

Liston, G. E. and C. A. Hiemstra (2008). A simple data assimilation system for complex snow distributions (snowassim). *Journal of Hydrometeorology 9*(5), 989–1004.

Liston, G. E. and M. Sturm (1998). A snow-transport model for complex terrain. *Journal of Glaciology 44*(148), 498–516.

Liston, G. E., J. G. Winther, O. Bruland, H. Elvehoy, and K. Sand (1999). Below-surface ice melt on the coastal antarctic ice sheet. *Journal of Glaciology 45*(150), 273–285.

Lundberg, A., N. Granlund, and D. Gustafsson (2010). Towards automated 'ground truth' snow measurements-a review of operational and new measurement methods for sweden, norway, and finland. *Hydrological Processes 24*(14), 1955–1970.

Martinec, J. and A. Rango (1981). AREAL DISTRIBUTION OF SNOW WATER EQUIVALENT EVALUATED BY SNOW COVER MONITORING. *WATER RESOURCES RESEARCH 17*(5), 1480–1488.

Solberg, R. (2004). A high-precision dynamic and adaptive BRDF and fractional snow-cover monitoring algorithm. In *IGARSS 2004: IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM PROCEEDINGS, VOLS 1-7 - SCIENCE FOR SOCIETY: EXPLORING AND MANAGING A CHANGING PLANET* , IEEE International Symposium on Geoscience and Remote Sensing (IGARSS), pp. 3721–3724. IEEE International Geoscience and Remote Sensing Symposium, Anchorage, AK, SEP 20-24, 2004.

Solberg, R. (2005). Model-driven retrieval of fractional snow cover area. In *IGARSS 2005: IEEE International Geoscience and Remote Sensing Symposium, Vols 1-8, Proceedings*, IEEE International Symposium on Geoscience and Remote Sensing (IGARSS), pp. 1943–1946. 25th IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2005), Seoul, SOUTH KOREA, JUL 25-29, 2005.

Solberg, R., J. Amlien, H. Koren, L. Eikvil, E. Malnes, and R. Storvold (2006). Multi-sensor/multi-temporal approaches for snow cover area monitoring. *EARSeL LIS-SIG Workshop, Berne, February 21-23, 2005*.

Solberg, R. and T. Andersen (1994). *AN AUTOMATIC SYSTEM FOR OPERATIONAL SNOW-COVER MONITORING IN THE NORWEGIAN MOUNTAIN REGIONS*. Igarss '94 - 1994 International Geoscience and Remote Sensing Symposium Volumes 1-4 - Surface and Atmospheric Remote Sensing: Technologies, Data Analysis and Interpretation. New York: I E E E.

Solberg, R., H. Koren, and J. Amilien (2006, 15.12.2006). A review of optical snow cover algorithms. Technical Report SAMBA/40/06, Norsk Regnesentral (Norwegian Computing Center).

Storvold, R., E. Malnes, Y. Larsen, K. A. Hogda, S. E. Hamran, K. Muller, and K. A. Langley (2006). SAR remote sensing of snow parameters in norwegian areas - current status and future perspective. *Journal of Electromagnetic Waves and Applications 20*(13), 1751–1759.

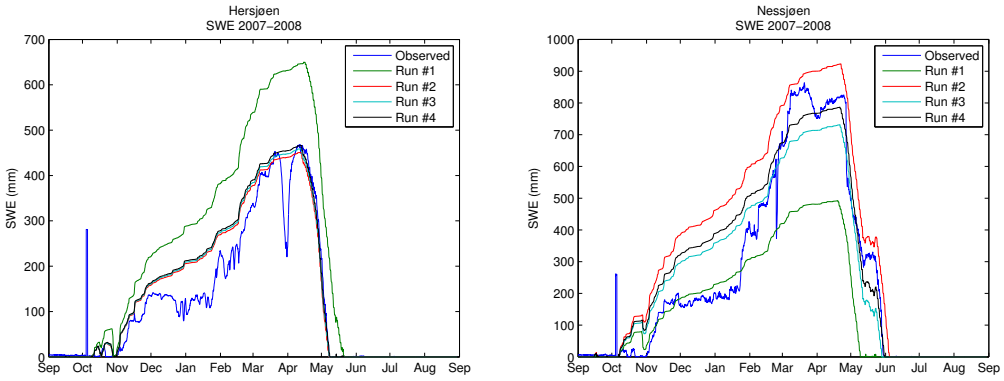
Storvold, R., E. Malnes, and I. Lauknes (2005). Using envisat asar wideswath data to retrieve snow covered area in mountainous regions. *EARSeL eProceedings 4*(2), 150–156.

Tabler, R. D. (1975). Predicting profiles of snowdrifts in topographic catchments. In *Proceedings of the 43rd annual western snow conference*, pp. 87–97.

Tarboton, D. G., T. G. Chowdhury, and T. H. Jackson (1995). A spatially distributed energy balance snowmelt model. In K. A. Tonnessen, M. W. Williams, and M. Tranter (Eds.), *Biogeochemistry of Seasonally Snow-Covered Catchments*, Iahs Publications, pp. 141–155.

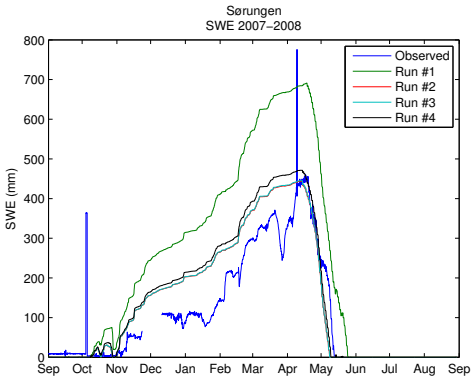
Winstral, A. and D. Marks (2002). Simulating wind fields and snow redistribution using terrain-based parameters to model snow accumulation and melt over a semi-arid mountain catchment. *Hydrological Processes 16*(18), 3585–3603.

A. Observed and iteratively corrected simulated SWE and snow depth for all three station, all three years



(a)

(b)



(c)

Figure 37: SWE of iterative correction runs compared with observed SWE values, 2007-2008

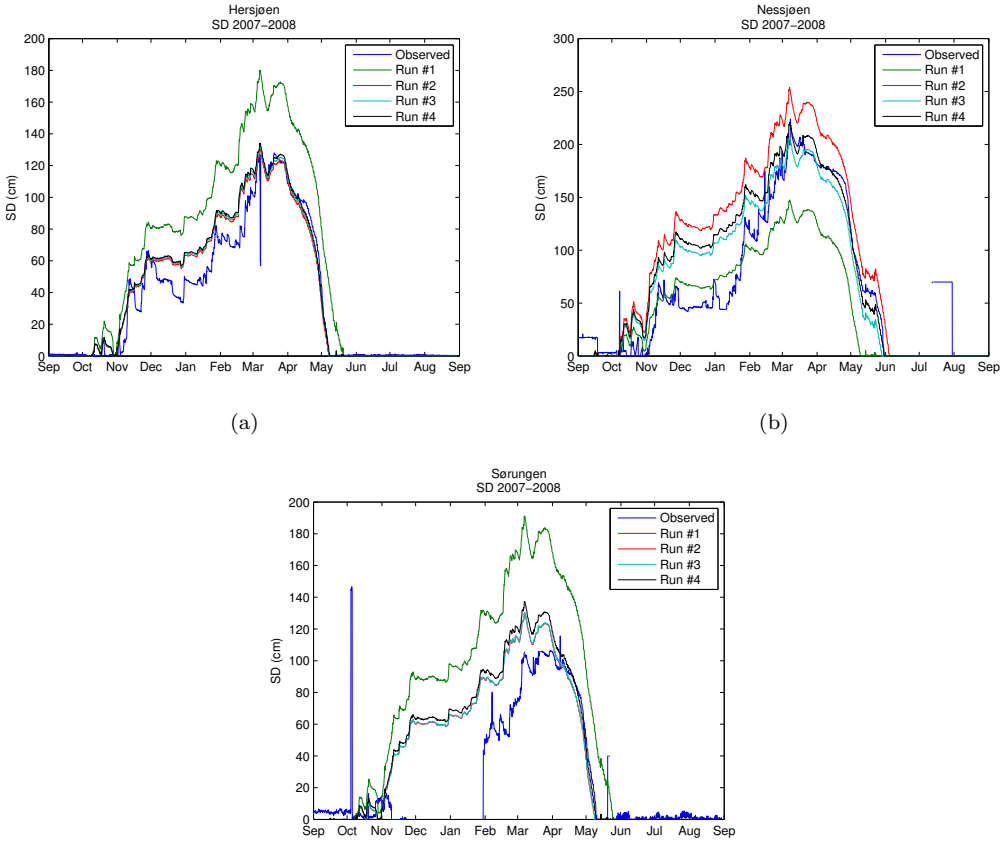


Figure 38: SD of iterative correction runs compared with observed values, 2007-2008

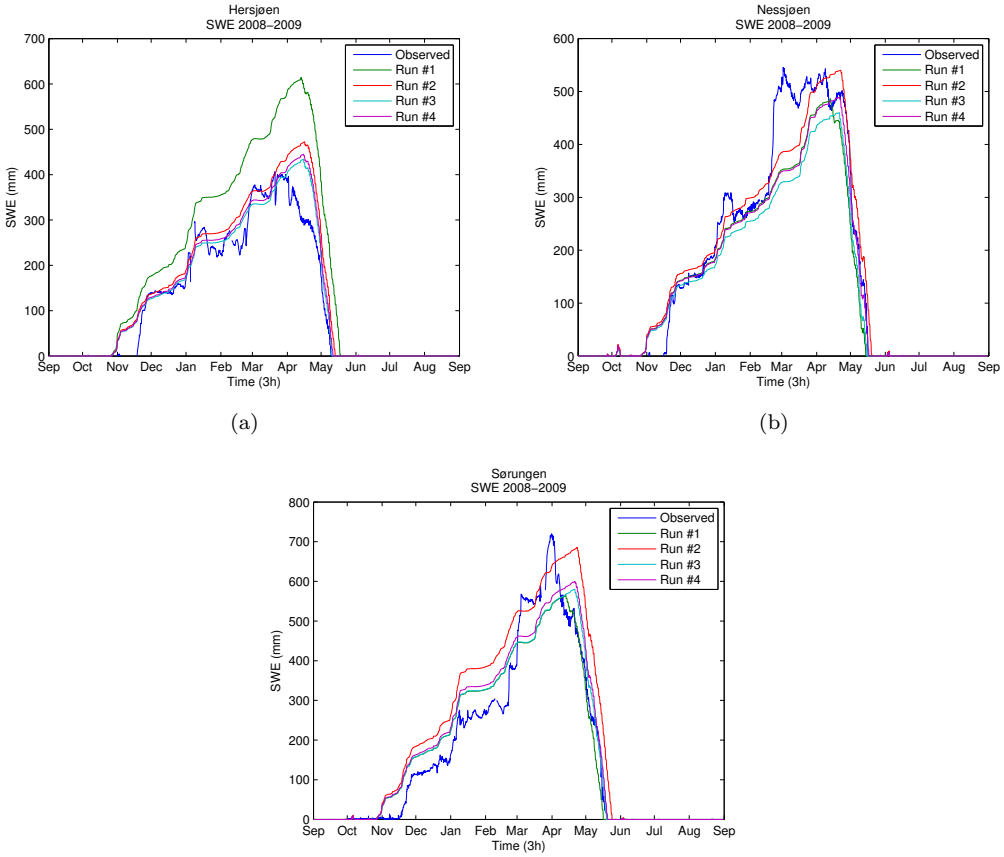


Figure 39: SWE of iterative correction runs compared with observed SWE values, 2008-2009

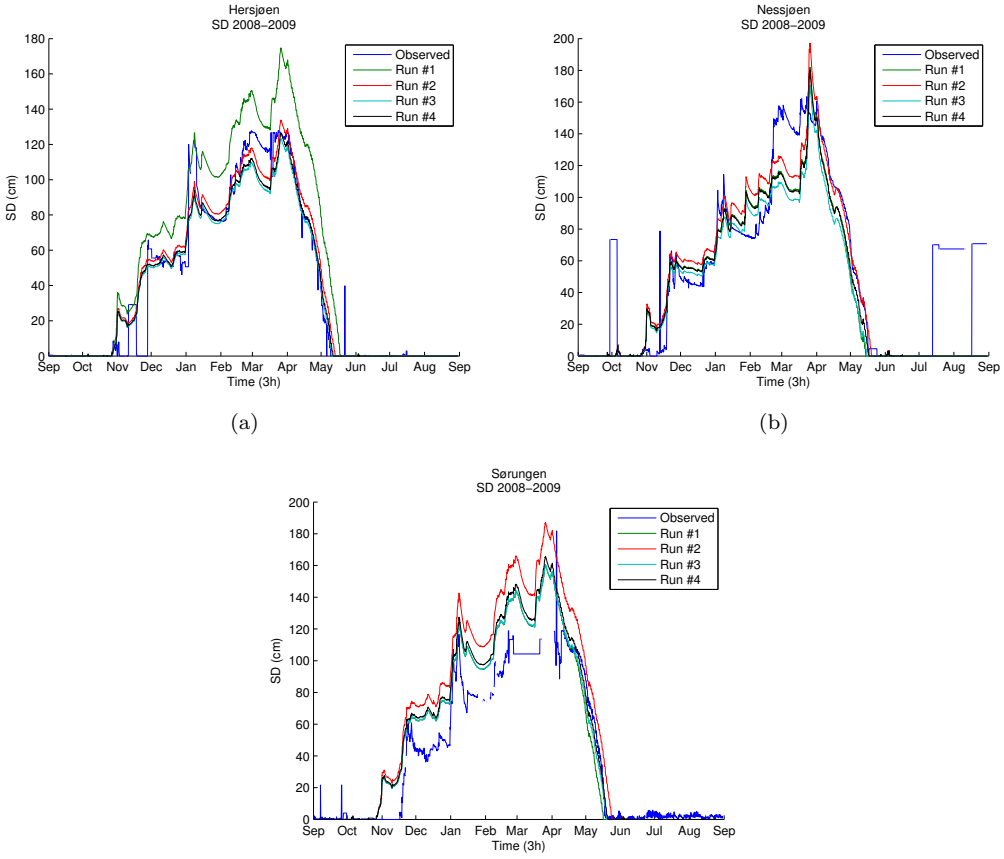


Figure 40: SD of iterative correction runs compared with observed values, 2008-2009

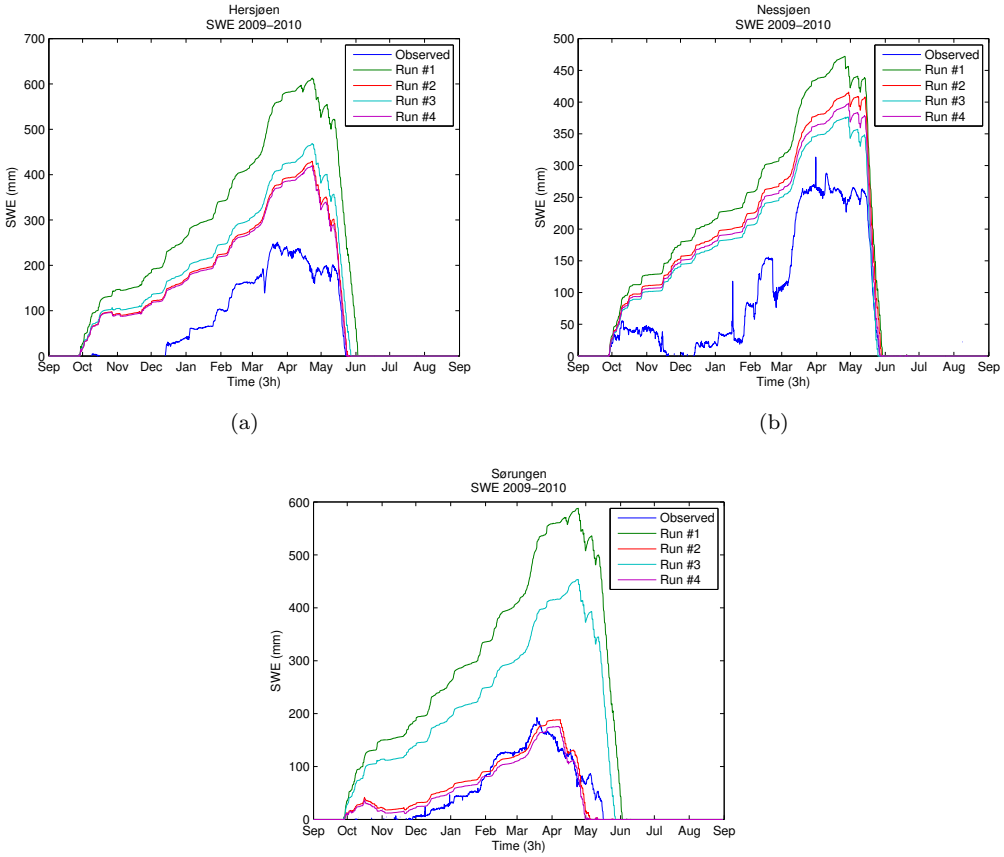


Figure 41: SWE of iterative correction runs compared with observed SWE values, 2009-2010

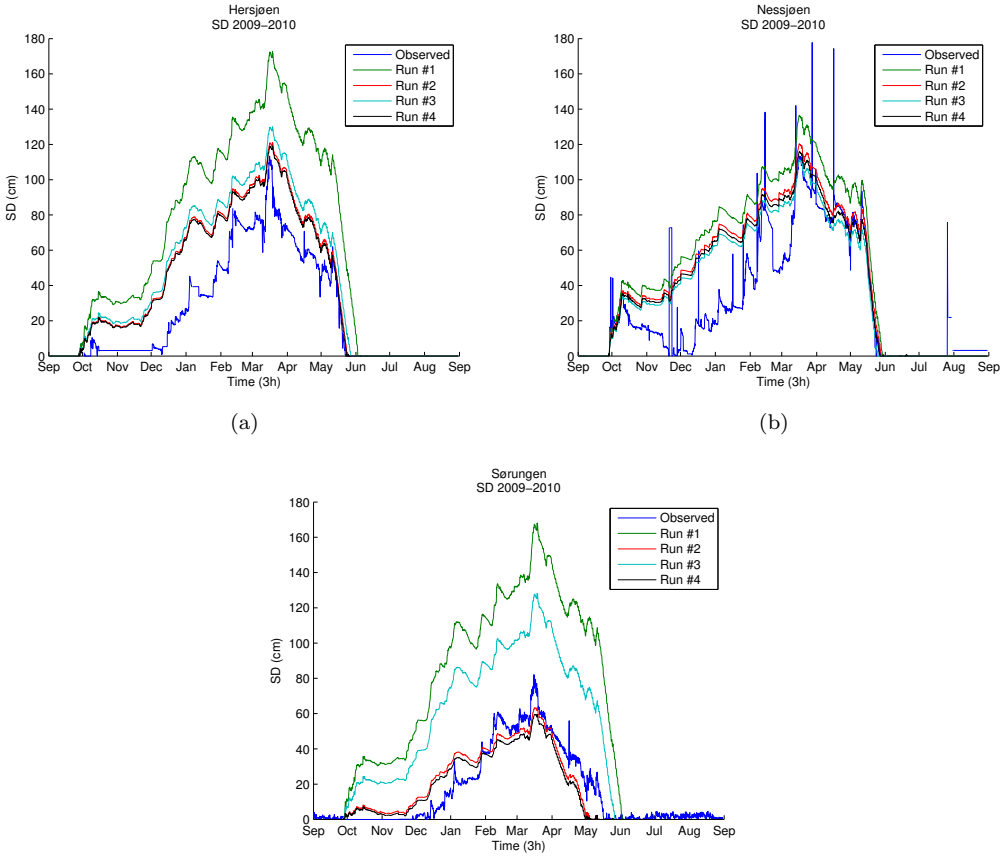


Figure 42: SD of iterative correction runs compared with observed values, 2009-2010

B. SnowModel vegetation classes

```
! The vegetation types are assumed to range from 1 through 30.  The
!   last 7 types are available to be user-defined vegetation types
!   and vegetation snow-holding depths.  The first 23 vegetation
!   types, and the associated vegetation snow-holding depths
!   (meters), are hard-coded to be:
!
! code description           veg_shd  example                    class
!
!  1  coniferous forest       15.00  spruce-fir/taiga/lodgepole  forest
!  2  deciduous forest        12.00  aspen forest                forest
!  3  mixed forest            14.00  aspen/spruce-fir/low taiga  forest
!  4  scattered short-conifer  8.00  pinyon-juniper              forest
!  5  clearcut conifer         4.00  stumps and regenerating     forest
!
!  6  mesic upland shrub       0.50  deeper soils, less rocky    shrub
!  7  xeric upland shrub       0.25  rocky, windblown soils      shrub
!  8  playa shrubland          1.00  greasewood, saltbush        shrub
!  9  shrub wetland/riparian   1.75  willow along streams        shrub
! 10  erect shrub tundra       0.65  arctic shrubland            shrub
! 11  low shrub tundra         0.30  low to medium arctic shrubs shrub
!
! 12  grassland rangeland      0.15  graminoids and forbs        grass
! 13  subalpine meadow         0.25  meadows below treeline      grass
! 14  tundra (non-tussock)     0.15  alpine, high arctic         grass
! 15  tundra (tussock)         0.20  graminoid and dwarf shrubs  grass
! 16  prostrate shrub tundra   0.10  graminoid dominated         grass
! 17  arctic gram. wetland     0.20  grassy wetlands, wet tundra grass
!
! 18  bare                     0.01                              bare
!
! 19  water/possibly frozen    0.01                              water
! 20  permanent snow/glacier   0.01                              water
!
! 21  residential/urban        0.01                              human
! 22  tall crops               0.40  e.g., corn stubble          human
! 23  short crops              0.25  e.g., wheat stubble         human
```

C. SnowModel parameters

```
! snowmodel.par file.
!
! Define any required constants specific to this model run.
!
! The following must be true:
!
!   All comment lines start with a ! in the first position.
!
!   Blank lines are permitted.
!
!   All parameter statements start with the parameter name, followed
!   by a space, followed by an =, followed by a space, followed by
!   the actual value, with nothing after that.  These statements can
!   have leading blanks, and must fit within 80 columns.
!
!   Also note that all of the input numbers follow standard
!   fortran_77 convention where anything starting with the letters
!   "i" through "n" are integers, and all others are real numbers.

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!  GENERAL MODEL SETUP
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! Number of x and y cells in the computational grid.
!      nx = 240
!      ny = 200
       nx = 480
       ny = 400

! deltax = grid increment in x direction.  Meters.
! deltay = grid increment in y direction.  Meters.
```

```
!      deltax = 500.0
!      deltay = 500.0
       deltax = 250.0
       deltay = 250.0

! Location (like UTM, in meters) value of lower-left grid point.
!   xmn = value of x coordinate in center of lower left grid cell.
!     Meters.
!   ymn = value of y coordinate in center of lower left grid cell.
!     Meters.
      xmn = 560000.
      ymn = 6960000.

! Model time step, dt. Should be the same increment as in the input
!   data file. In seconds.
!   One day.
!      dt = 86400.0
!   Six hours.
!     dt = 21600.0
!   Three hours.
     dt = 10800.0
!   One hour.
!     dt = 3600.0

! Start year of input data file. Four digit year. Integer.
      iyear_init = 2007

! Start month of input data file. One or two digit month. Integer.
      imonth_init = 9

! Start day of input data file. One or two digit day. Integer.
      iday_init = 1

! Start hour of input data file. Local time, like in solar time.
!   Decimal hour. Each day the clock runs from 0.00 through 23.99.
!   Real.
      xhour_init = 0.0

! Number of model iterations defines how many times to process.
      max_iter = 2912

! MicroMet requires a met input file that has data in a specific
!   format (see the MicroMet documentation). For the case of
!   inputs from more than a single station, the input met file
!   requires an initial number indicating the number of valid
!   stations at a given time step. For the case of a single
!   station, this number can be dropped (it doesn't have to be
!   though). Identify whether to process as a single station
!   without the valid station count (isingle_stn_flag = 1), or
!   or to process with the count included (for multi stations
!   and single stations with the number of stations included;
!   isingle_stn_flag = 0).
      isingle_stn_flag = 0

! For the case of isingle_stn_flag = 1, the input file can be a GrADS
!   binary file. Identify whether that is the case (=1, else =0).
      igrads_metfile = 0

! Define the meteorologial input file name. Note that this input
!   file has very specific input format requirements (see the
!   MicroMet preprocessor). The required met variables are:
!   Tair (deg C or K), rh (%), wind speed (m/s), wind direction
!   (0-360 True N), and precipitation (mm/time_step).
      met_input_fname = metfile

! The number used as an undefined value for both inputs and outputs.
      undef = -9999.0

! Define whether the topography and vegetation input files will
!   be ARC/INFO ascii text (grid) files, or a GrADS binary file
!  (ascii = 1.0, GrADS = 0.0).
      ascii_topoveg = 1.0

! Define the GrADS topography and vegetation input file name
!   (record 1 = topo, record 2 = veg). Note that if you are using
!     ascii files, you still cannot comment this line out (it is okay
!     for it to point to something that doesn't exist) or you will
!     an error message.
!     topoveg_fname = xxxxxx
!     topoveg_fname = topo_veg/topovegascii250/nea.topoveg.500m.gdat
```

```
      topoveg_fname = topo_veg/regrid/nea.topoveg.100m.gdat

! For the case of using ascii text topography and vegetation files,
!   provide the file names.  Note that if you are using a grads
!     file, you still cannot comment these two lines out (it is okay
!     for it to point to something that doesn't exist) or you will
!     an error message.
      topo_ascii_fname = topo_veg/topoveg_ascii250/nea_dem_250m.txt
      veg_ascii_fname = topo_veg/topoveg_ascii250/recl_veg250.txt

! Define whether the vegetation will be constant or defined by the
!   topography/vegetation input file name (0.0 means use the file,
!   1.0 or greater means use a constant vegetation type equal to
!   the number that is used).  This will define the associated
!   veg_shd that will be used.  The reason you might use a constant
!   vegetation type is to avoid generating a veg-distribution file.
!     const_veg_flag = 12.0
      const_veg_flag = 0.0

! For the case where you have vegetation height data, and want to
!   use that instead of the tables above, you must also provide a
!   file that defines the vegetation height (cm) of each grid cell.
!   iveg_ht_flag = 0 means the file is not provided and the required
!   array will be generated by filling it with the values listed in
!   the above vegetation summary.  iveg_ht_flag = -1 means the data
!   file is a grads binary file with the name veg_ht.gdat.
!   iveg_ht_flag = 1 means the data file is an ascii text file with
!   the same format that ascii topo and veg files would have and with
!   a name of veg_ht.asc.  Note that the .gdat and .asc files are
!   y-reversed (see ascii topo/veg notes).  It is assumed that the
!   file, if provided, is located in an 'topo_veg/' directory off
!   of the main model directory.  Note that this height information
!   is primarily used as the vegetation snow-holding depth, thus the
!   model is expecting average vegetation heights for each grid cell,
!   not maximum heights.
      iveg_ht_flag = 0

! The latitude of domain center (decimal degrees).  This only needs
!   to be approximate, since it is used to define some of the solar
!   radiation calculations.
      xlat = 63.0

! For the case where your domain spans enough latitude that there
!   is significant variation in solar radiation from the top to
!   the bottom of the domain, you must also provide a file that
!   defines the latitude (in decimal degrees) of the center of
!   each grid cell.  lat_solar_flag = 0 means the file is not
!   provided and the required array will be generated by filling
!   it with the constant xlat value listed above.  lat_solar_flag
!   = -1 means the data file is a grads binary file with the name
!   grid_lat.gdat.  lat_solar_flag = 1 means the data file is an
!   ascii text file with the same format that ascii topo and veg
!   files would have and with a name of grid_lat.asc.  Note that
!   the .gdat and .asc files are y-reversed (see ascii topo/veg
!   notes).  It is assumed that the file, if provided, is located
!   in an 'extra_met/' directory off of the main model directory.
      lat_solar_flag = 0

! For the case where your domain spans enough lonitude that there
!   is significant variation in solar radiation from the side to
!   the side of the domain, you must also provide a file that
!   defines the lonitude (in decimal degrees) of the center of
!   each grid cell.  In this case it also no longer makes sense
!   to be running the model in local time, and UTC (or GMT) time
!   is required.  UTC_flag = 0.0 means the the model will be run
!   in local time.  UTC_flag = -1.0 means the longitude data file
!   is a grads binary file with the name grid_lon.gdat.  UTC_flag
!   = 1 means the longitude data file is an ascii text file with
!   the same format that ascii topo and veg files would have and
!   with a name of grid_lon.asc.  Note that the .gdat and .asc
!   files are y-reversed from each other (see ascii topo/veg
!   notes).  It is assumed that the file, if provided, is located
!   in an 'extra_met/' directory off of the main model directory.
!   Note that if UTC_flag is non-zero, lat_solar_flag should
!   probably also be non-zero.
      UTC_flag = 0.0

! Define which models that are going to run.  A value of 1.0 means
!   that you want to run the model (0.0 means don't run it).
!   Currently the model is only configured to run the following
```

```
!   combinations (micromet), (micromet, enbal), (micromet, enbal,
!   snowpack), (micromet, snowtran), or (micromet, enbal, snowpack,
!   snowtran).
      run_micromet = 1.0
      run_enbal = 1.0
      run_snowpack = 1.0
      run_snowtran = 0.0

! Define whether you are doing a standard model simulation, or a
!   precipitation and/or melt correction simulation to force the
!   model towards observed swe distributions.  For a standard
!   simulation, use irun_corr_factor = 0, for a precipitation/melt
!   factor correction run, use irun_corr_factor = 1.  Note that
!   irun_corr_factor = 1 requires swe data and some extra input
!   files.  See the dataassim_user.f subroutine for details.  If
!   you are using this feature, you will also need to edit the
!   outputs_user.f file.
      irun_corr_factor = 1
!     irun_corr_factor = 1

! Define whether want to implement the history restart option.
!   This creates periodic output files that allow you to restart
!   the simulation from somewhere other than the beginning.  All
!   of the required data arrays are saved to a grads file that is
!   read back in when the simulation is restarted.  The new
!   simulation starts at the history restart iteration, with the
!   initial conditions defined by the saved data arrays.  The
!   new simulation starts in the micromet file where the original
!   simulation ended.  This is also true of the output files,
!   if the run has been set up to write continuously to a given
!   output file.  This option also assumes that you have not made
!   any changes to the model setup; it does no error checking.
!   ihrestart_flag = -2 turns this option off, ihrestart_flag = -1
!   does the restart saves, and ihrestart_flag >= 0 runs a history
!   restart where the number defines the restart iteration (e.g.,
!   ihrestart = 720 will set iteration 720 as the intitial
!   condition for the start of the new simulation.  For the case
!   where ihrestart_flag = 0, and i_dataassim_loop <= -1, the
!   run will start at the begining of the second data assimilation
!   loop.
      ihrestart_flag = -2
!     ihrestart_flag = -1
!     ihrestart_flag = 3444
!     ihrestart_flag = 0

! For the case where you are running a history restart, define
!   whether you want to restart the run during a standard run or
!   the first loop of a data assimilation model run
!   (i_dataassim_loop = 1) or during the second loop of a data
!   assimilation run (i_dataassim_loop < 0).  The value of this
!   negative number is used to define how many obervation dates
!   you have in your data assimilation run (e.g., -3 would mean
!   you have 3 observation dates and nobs_dates = 3 in
!   dataassim_user.f).  Also note that to correctly run the history
!   restart function for the case of a data assimilation run, you
!   need to first set i_corr_factor = 0 and save a history restart
!   at the end of that run before going on to the second half of
!   the assimilation run.
      i_dataassim_loop = -1
!     i_dataassim_loop = -1

! If ihrestart_flag = -1, then define how often the restart
!   files are generated.  This is in iteration units (e.g., every
!   30 days for the model running at hourly time steps would be
!   ihrestart_inc = 720).  The history restart files will be
!   placed in an 'hrestart/' directory off of the main model
!   directory.  In addition to this time increment, if the
!   ihrestart_flag = -1, the last model iteration is always saved
!   (so if you just want to save the last iteration, just set
!   ihrestart_inc to a number bigger than your maximum iteration).
!   Note that if you want to do history restarts with data
!   assimilation this increment must be evenly divisible into
!   the maximum interation; in other words, the simulation must
!   at least do a write at the end of the first data assimilation
!   loop.
      ihrestart_inc = 10000

! The code is set up to write out an individual data file for each
!   sub-model (see the sub-model sections below).  In addition, you
!   can use a user-defined subroutine to output data in the format
```

```
!   of your choice.  Define whether you want the data written out
!   to this file (print_user = 1.0, else 0.0).  The name and
!   location of the user-defined output file(s) will be defined
!   within the subroutine.  If you are using this feature, you
!   will need to edit the outputs_user.f file.
      print_user = 1.0

! For printing to the standard output files, define the iteration
!   increment for each write to the files (e.g., iprint_inc = 1
!   gives every time step, iprint_inc = 24 gives a write at the end
!   of each day when using hourly time steps, etc.).
!     iprint_inc = 24
      iprint_inc = 1

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!  MICROMET MODEL SETUP
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! Define which fields you want to process/distribute. 1 = do it,
!   0 = don't do it.  As an example, if you are running SnowTran-3D
!   with no melting, you don't need solar radiation.
      i_tair_flag = 1
      i_rh_flag = 1
      i_wind_flag = 1
      i_solar_flag = 1
      i_longwave_flag = 1
      i_prec_flag = 1

! Force the model to determine a value for every grid cell (=1), or
!   just define values for some specified "radius of influence" (=0).
      ifill = 1

! Let the model determine an appropriate "radius of influence" (=0),
! or define the "radius of influence" you want the model to use (=1).
! 1=use obs interval below, 0=use model generated interval.
      iobsint = 0

! The "radius of influence" or "observation interval" you want the
!   model to use for the interpolation.  In units of deltax, deltay.
      dn = 1.0

! The barnes station interpolation can be done two different ways:
!   First, barnes_oi does the interpolation by processing all of
!     the available station data for each model grid cell.
!   Second, barnes_oi_ij does the interpolation by processing only
!     the "n_stns_used" number of stations for each model grid cell.
!   For small domains, with relatively few met stations (100's),
!   the first way is best.  For large domains (like the
!   United States, Globe, Pan-Arctic, North America, Greenland)
!   and many met stations (like 1000's), the second approach is the
!   most efficient.  But, the second approach carries the following
!   restrictions: 1) there can be no missing data for the fields of
!   interest; 2) there can be no missing stations (all stations
!   must exist throughout the simulation period); and 3) the
!   station met file must list the stations in the same order for
!   all time steps.  This second method works well for regridding
!   atmospheric analyses/model datasets.  Also, the MicroMet
!   preprocessor can be used to fill in missing data segments.
!   Use barnes_lg_domain = 0.0 for the first method, and
!   barnes_lg_domain = 1.0 for the second method.
      barnes_lg_domain = 0.0

! For the case where barnes_lg_domain = 1.0, define the number
!   of nearest stations to be used in the interpolation (5 or
!   less).  If barnes_lg_domain = 0.0, n_stns_used is not used, but
!   n_stns_used still needs to have some value.
      n_stns_used = 5

! The curvature is used as part of the wind model.  Define a length
!   scale that the curvature calculation will be performed on.  This
!   has units of meters, and should be approximately one-half the
!   wavelength of the topographic features within the domain.
      curve_len_scale = 600.0

! The curvature and wind_slope values range between -0.5 and +0.5.
!   Valid slopewt and curvewt values are between 0 and 1, with
!   values of 0.5 giving approximately equal weight to slope and
!   curvature.  I suggest that slopewt and curvewt be set such
```

```
!   that slopewt + curvewt = 1.0.  This will limit the total
!   wind weight to between 0.5 and 1.5 (but this is not required).
      slopewt = 0.58
      curvewt = 0.42
!     slopewt = 0.25
!     curvewt = 0.75

! Avoid problems of zero (low) winds (for example, turbulence
!   theory, log wind profile, etc., says that we must have some
!   wind.  Thus, some equations blow up when the wind speed gets
!   very small).  This number defines the value that any wind speed
!   below this gets set to.
      windspd_min = 1.0

! Define whether the model is to use the default monthly lapse
!   rates (= 0) or user supplied monthly lapse rates (= 1).  To use
!   user supplied lapse rates, you have to edit the user lapse rate
!   data array in micromet_code.f (subroutine get_lapse_rates).
!   Note that this implementation currently only defines
!   average-monthly lapse rates.
      lapse_rate_user_flag = 1

! Define whether the precipitation adjustment factor, with units of
!   km^-1 (kind of a precipitation lapse rate, used to adjust the
!   precipitation for locations above and below the precipitation
!   observing station(s)), is to use the default monthly lapse
!   rates (= 0) or user supplied monthly lapse rates (= 1).  To use
!   user supplied lapse rates, you have to edit the user lapse rate
!   data array in micromet_code.f (subroutine get_lapse_rates).
      iprecip_lapse_rate_user_flag = 1

! Define whether you want the model to calculate, use, and output
!   sub-forest-canopy windspeed, incoming solar and longwave
!   radiation (= 1.0), or above canopy values (= 0.0).
      calc_subcanopy_met = 1.0

! Define the canopy gap fraction (0-1).  This parameter accounts
!   for solar radiation reaching the snow surface below the canopy,
!   beyond that defined by the canopy transmissivity calculation.
!   In effect, it allows additional solar radiation to penetrate
!   the canopy (e.g., through gaps in the forest), thus inceasing
!   melt rates, etc.  A gap_frac = 0.0 produces the sub-canopy
!   solar radiation using the default transmissivity calculation,
!   a gap_frac = 1.0 (all gaps) produces sub_canopy radiation equal
!   to the top-of-canopy radiation.  If you want the snow in the
!   forest to melt faster, increasing this value will do it.
      gap_frac = 0.20

! To handle the case, for example, of a met station located in
!   an inversion layer recording high relative humidity, and the
!   model producing anomolously high cloud-cover fractions, the
!   cloud_frac_factor can be used to decrease the simulated cloud
!   fraction.  This number is multiplied by the calculated cloud
!   fraction.  For example, a cloud_frac_factor = 1.0 produces the
!   simulated cloud fraction, a cloud_frac_factor = 0.5 produces
!   the half the simulated cloud fraction, and a cloud_frac_factor
!   = 0.0 forces the simulated cloud fraction to be zero.
      cloud_frac_factor = 1.0

! Define whether the simulation will assimilate shortwave radiation
!   observations (no = 0.0, yes = 1.0).  If yes, the model assumes
!   there is a shortwave radation station data file, called
!   shortwave.dat, in an 'extra_met/' directory off of the main
!   model directory.  See the micromet code for file format details.
      use_shortwave_obs = 0.0

! Define whether the simulation will assimilate longwave radiation
!   observations (no = 0.0, yes = 1.0).  If yes, the model assumes
!   there is a longwave radation station data file, called
!   longwave.dat, in an 'extra_met/' directory off of the main model
!   directory.  See the micromet code for file format details.
      use_longwave_obs = 0.0

! Define whether the simulation will assimilate surface pressure
!   observations (no = 0.0, yes = 1.0).  If yes, the model assumes
!   there is a surface pressure station data file, called
!   sfc_pressure.dat, in an 'extra_met/' directory off of the main
!   model directory.  See the micromet code for file format details.
      use_sfc_pressure_obs = 0.0
```

```
! The code is set up to write out an individual data file for each
!   sub-model.  Define whether you want the data written out
!   (print_micromet = 1.0, else 0.0), and the name of that output
!   file.
      print_micromet = 0.0
      micromet_output_fname = outputs/micromet.gdat

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!  SNOWTRAN-3D MODEL SETUP
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!SnowTran-3D not used, parameters excluded

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!  ENBAL-2D MODEL SETUP
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! Identify whether the 2-D surface energy balance calculation for
!   this simulation will include a non-zero conduction term
!   (icond_flag = 0 = no conduction, 1 = conduction).  Note that
!   the icond_flag = 1 has not been fully implemented yet.
      icond_flag = 0

! Define the albedo for a melting snowcover under the forest
!   canopy.  This allows the user to control, to some degree, the
!   melt rates in the forests.  Note that adjusting the gap-frac
!   parameter is also an effective way to do this.  The non-melting
!   snow albedo is set to 0.8 in the code.
      albedo_snow_forest = 0.45

! Define the albedo for a melting snowcover in non-forested areas.
!   This allows the user to control, to some degree, the melt
!   rates in non-forested areas.  The non-melting snow albedo is
!   set to 0.8 in the code.
      albedo_snow_clearing = 0.60

! Define the albedo for a glacier surface (dry and melting).
      albedo_glacier = 0.40

! The code is set up to write out an individual data file for each
!   sub-model.  Define whether you want the data written out
!   (print_enbal = 1.0, else 0.0), and the name of that output
!   file.
      print_enbal = 0.0
      enbal_output_fname = outputs/enbal.gdat

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!  SNOWPACK MODEL SETUP
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

! Define whether static-surface (non-blowing snow) sublimation will
!   be included in the model calculations (sfc_sublim_flag = 1.0).
!   To turn this off, set sfc_sublim_flag = 0.0.  I am waiting for
!   the flux-tower data Matthew and I are collecting in Alaska, to
!   validate this part of the model simulations.  If the
!   sfc_sublim_flag is turned on, the latent heat flux (Qe)
!   calculated in ENBAL is used to add/remove snow from the
!   snowpack.
      sfc_sublim_flag = 1.0

! The code is set up to write out an individual data file for each
!   sub-model.  Define whether you want the data written out
!   (print_snowpack = 1.0, else 0.0), and the name of that output
!   file.
      print_snowpack = 0.0
      snowpack_output_fname = outputs/snowpack.gdat
```