Master Thesis

Mountainous snow cover monitoring and snowmelt runoff discharge modeling in the Langtang Khola catchment

Xinliang Li

Department of Physical Geography, Utrecht University

Utrecht, 2010

Supervisor:

Dr. Derek karssenberg

Dr. Walter Immerzeel

C0-supervisor:

Dr. Marc Bierkens

Contents

Figures

Tables

Acknowledgements

My deepest gratitude goes first and foremost to Dr. Derek karssenberg, my supervisor, for his constant encouragement and guidance. He has walked me through all the stages of the writing and studying of this thesis and given me a lot of construction comments to improve the first version. Without his consistent and illuminating instruction, the thesis could not have reached its present version. His enthusiasm and attitude to the academy influence me deeply.

Second, I would like to express my heartfelt gratitude to Dr. Walter Immerzeel, who gave me first version for the model realization and a lot of useful data sets. Anytime when I had questions he would give me answers very soon. I am also greatly indebted to the professor Marc Bierkens, who brought me into the area of stochastic hydrology. Other professors and teachers at the faculty of geoscience: Professor Steven De Jong, Dr. Oliver Schmitz et al., who have instructed and helped me a lot during courses and researches in the past two years.

Last my thanks would go to the Dutch government and Utrecht University to support my study in the Netherlands. Even though I will work back to my county, Dutch culture and societal system will influence my life deeply.

1 Introduction

1.1 Background

Mountainous snow cover exists in many regions on earth and this snow cover can regulate climate cycle (Oerlemans and Fortuin, 1992), global temperature (Rebetez, 1996), water cycle, monsoon intensity of inland areas (Douville and Peings, 2010) and so on. Snow plays a key role in the hydrologic cycle over a large area of mid latitudes regions through its effects on the water storage and surface albedo (Andreadis and Lettenmaier, 2006). Model simulations demonstrate that local snow albedo feedbacks can enhance the North American climate anomalies related to El Nino-Southern Oscillation processes (Cohen and Entekhabi, 1999). Moreover, mountainous snow melt runoff is an important fresh water source of main inland rivers (Immerzeel, et al, 2009). For example, in Asia, the Indus, Ganges, Brahmaputra, Irrawaddy, Salween, Mekong, Yellow and Yangtze rivers are originated from Himalayas and Tibet-Qinghai Plateau. In Europe, the Rhone, Rhine, Danube and Po rivers are originated from Alps Mountains. Water stored in the glaciers represents an important component of the hydrologic balance in many regions of the world. Besides, the snow cover varies with season (Maqueda and Fichefet, 1999), which causes snow and snow melt to have a non-uniform distribution. The existence of snow cover affects many physical, chemical and biological processes, and has an enormous influence on society and economical development (Liang et al, 2008 a, Hall et al, 2002 and Xiao et al, 2004). In many regions the drought and Flood and Waterlogging may happen, causing damages for human beings. Consequently, it is of importance to carry out research of snow cover monitoring and discharge modelling from snow melts.

Monitoring of snow cover and modelling discharge from snowmelt are particularly difficult in mountainous areas because of the large spatial variability of snow characteristics and, often, limited availability of ground based hydrologic data (Parajka and Bloschl, 2008). But since 1966, snow cover data have been available from the National Oceanic and Atmospheric Administration (NOAA). On December 18, 1999, the Terra satellite was launched with five instruments including the Moderate Resolution Imaging Spectroradiometer (MODIS) (Hall et al, 2002). Global snow-cover products are derived from MODIS data and the retrieval methods have been developed for MODIS data themselves (Hall et al, 1995). The new product provides a more convenient and accurate data source for snow cover analysis. Many new researches about snow cover monitoring (Lopez, et al, 2008 and Liang, 2008), hydrologic models (Tekeli, et al, 2005 and Georgievsky, 2009), snow related parameters estimation (Salomonson and Appel, 2004 and Lyapustin et al, 2009) have been explored with the help of the MODIS data source. We can carry out seasonal snow cover monitoring analysis by applying time series MODIS data because the temporal and spatial resolution of snow products are suitable for dynamic analysis.

In hydrological snowmelt models, many parameters are interpolated using spatial algorithm (Minville, et al., 2008), where interpolation is done when many samples on parameters are

available, closely to each other and even a simplified method was used that scalar values are assigned to the regions). Due to the heterogeneous of mountainous ground, the uncertainties of input data would not be always accepted (Parajka and Bloschl, 2008). The uncertainties within hydrological models can be quantified and even narrowed using assimilation method (Kitanidis and Bras, 1980, Kuczera and Mroczkowski, 1988, Kuczera and Parent, 1998 and Weerts and EI Serafy, 2006). Data assimilation is a crucial method to quantify and narrow the uncertainties hence improving model results. The basic principle of data assimilation is to estimate and quantify model errors between different data sources, and update hydrological model states in a way that optimally combines model simulations with other source data (state observation or other model outputs) (Clark, et al, 2008). Data assimilation can obtain more reliable model results because the model errors have been narrowed. Hence, the data assimilation has great potential in hydrology snow melt runoff modeling because different observations (e.g. snow cover remote sensing data, discharge observation, snow water equivalent) can be assimilated into the model.

1.2 Objective and approach

Depth and area of snow cover are key variables in snow melt runoff generation. The snow cover monitoring makes us obtain more accurate and abundant knowledge of snow cover change and its response to the current climate changes. In the research, there are three topics: i) mountainous snow cover monitoring based on MODIS remote sensing data; ii) Snow discharge from snow melt runoff model and observed discharge from snow cover remote sensing data, and iii) assimilating observed discharge into snow melt runoff model.

The objectives and main topics of this thesis are:

1. Describing the seasonal and annual trends of snow cover change in the Langtang catchment using remote sensing techniques and finding the relationship between climate and this trend.

Remote sensing techniques provide a real-time data on snow cover for large areas. There are inter- and between- annual change trends of snow cover. In inter-annual monitoring, the trend of snow cover fluctuates with season in response to seasonal fluctuations in temperature. While in between-annual snow cover monitoring, the trend of annual snow cover is mainly dominated by local climate and environment conditions. The trend knowledge of snow cover is important for snow cover pattern understanding, snow cover forecasting and policy making.

2. Modeling snow melt discharge using a snowmelt runoff model (SRM) and integrating snow cover remote sensing observations into the snowmelt runoff model.

The mountainous snow and glaciers are an important part of water source for many main inland rivers. Modeling snow melt and describing seasonal trend of snow water contribution to rivers are important. A snow melt runoff model (SRM) has been used to model the snowmelt processes. This model is simple to implement and can be easily integrated with remote sensing snow cover data. Remote sensing snow cover provides real-time monitoring data source for the snow cover so if real time snow cover data are integrated into snowmelt runoff model, the model performance can improve a lot. In this part of research, an integration method for remote sensing snow cover and snowmelt runoff model has been developed.

3. Assimilating discharge from remote sensing snow cover data into snowmelt runoff model using ensemble Kalman filter (EnKF) and particle filtering to narrow model uncertainty.

The snowmelt runoff discharge has been modeled with snowmelt runoff model (SRM) and integration snow covers into the model. Data assimilation techniques are applied to narrow the uncertainty of the two model outputs. In the research, the EnKF and the particle filter have been used to assimilate the two modeled outputs.

1.3 Thesis outline

The research case presented in the thesis is in Langtang Khola catchment, Nepal.

The thesis is organized as follows

Chapter 2 is titles as "*Snow cover monitoring based on remote sensing data in Langtang Khola catchment*". This part includes snow cover monitoring analysis with time and its response to the local climate conditions in Lang Khola catchment.

Chapter 3 is titled as "*Discharge modeling from snow melt runoff model and from remote sensing snow cover data*". Snowmelt Runoff Model (SRM) is applied in the research. Remote sensing snow cover data is integrated into SRM to improve model performance.

Chapter 4 is titled as "*Assimilating snow melt runoff results and remote sensing data*". Monte Carlo simulation, ensemble Kalman filter and particle filter are applied in this part. The comparison analysis of assimilation techniques is carried out to determine the method performance and the parameter sensitivity.

Chapter 5 is the conclusion and discussion for the thesis.

2 Snow cover monitoring based on remote sensing data in the Langtang Khola catchment

2.1 Introduction

Mountainous snow is an important solid water source for mainly inland rivers (Schaper et al., 1999) and the change of snow cover is sensitive to the climate change and global water cycles (Orelemans and Fourtuin 1992). Monitoring mountainous snow covers and snow cover changes provides us more knowledge about snow space distribution and their spatial-temporal differences. The real-time and large scale snow cover monitoring is of high importance to know the detailed snow cover patterns.

Remote sensing data is an important monitoring data source for snow cover analysis. Recently, with the development of remote sensing techniques especially MODIS Terra launching since 1999 (Hall, et al, 2002), the snow cover monitoring publications have appeared in different fields. Large scale and real-time snow cover observations provide an important data source for monitoring. Remote sensing is playing an increasingly important role in snow cover monitoring because large scale snow cover monitoring has been possible. Moreover, the temporal and spatial resolution has been improved at a high extent (Liang et al, 2008 b). Snow cover information from remote sensing data has been integrated into hydrologic models, including snow melt runoff model (Georgievsky, 2009), and snow water equivalent (Schaffhauser et al, 2008), soil water extraction and so on.

Using the Terra and Aqua MODIS combined snow cover products, the purpose of this chapter is to examine the temporal variation of snow cover conditions in the Langtang Khola catchment and to find the relationship between the temporal variation of snow cover change and climate conditions. The temporal variation of snow cover change contains two parts: inter- and between annual change trends, which are tightly related with local climate, in particular temperature and climate.

The reminder of this part research is organized as follows. The next section describes the Langtang Khola catchment and MODIS products data containing seven-year (2000-2006) observations. The method section introduces inter and between annual analysis and the relationship between trend and climate condition in this catchment. The final section discusses the results and concludes with remarks on the observed patterns in snow cover.

2.2 Research area and data sets

2.2.1 Research area introduction

The Langtang Khola is situated in the northern part of central Nepal, and is the head water area of the Trisuli River in the Narayani River System. The climate of Langtang is dominated

by the southwest Indian summer monsoon. Monsoonal weather systems move from south to north into the Langtang Himalayas. The valley profile of the Langtang catchment is asymmetric, with the bulk of fan and glacial sedimentation originating on the south-facing slopes that capture the majority of the monsoonal precipitation.

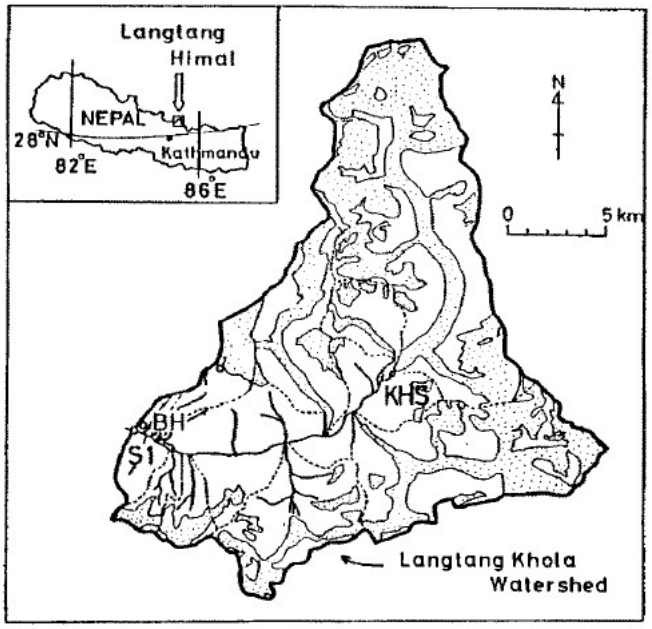


Figure 2.1 Location of study case area-Langtang Khola watershed (Fukushima et al., 1987)

2.2.2 Data sets

The data source used in this research is MODIS snow products from MODIS Terra and MODIS Aqua versions. Bias to Terra is because the snow detection algorithm is based on use of near infrared data at 1.6 μm (Sirguey, et al. 2009). A primary key to snow detection is the characteristic of snow to have high visible reflectance and low reflectance in the near infrared, MODIS band 6. The MODIS data include 315 bands, all of which are snow cover information each eight days from 2000 to 2006. The pixel size is 500×500m.

The climate data is daily precipitation and temperature data. These data is applied to find the relationship between snow cover change trend and local climate change. Due to the scale difference between snow cover data and climate data source, only the same date data are used in the following analysis.

2.3 Methodology

2.3.1 Inter-annual analysis

The inter-annual snow cover change trend reflects the seasonal snow fluctuation. This fluctuation information directly determines snow melt discharge (Georgievsky, 2009). MODIS snow products for The Langtang Khola catchment provide 315 bands covering seven years from 2000 to 2006. By making statistics for the snow cover area, we can get figure 2.2. The figure is to make a statistics of snow cover percentage in the area for each date in the seven years, reflect the overall annual distribution of snow cover in this watershed and obtain the fluctuation trends in different time each year.

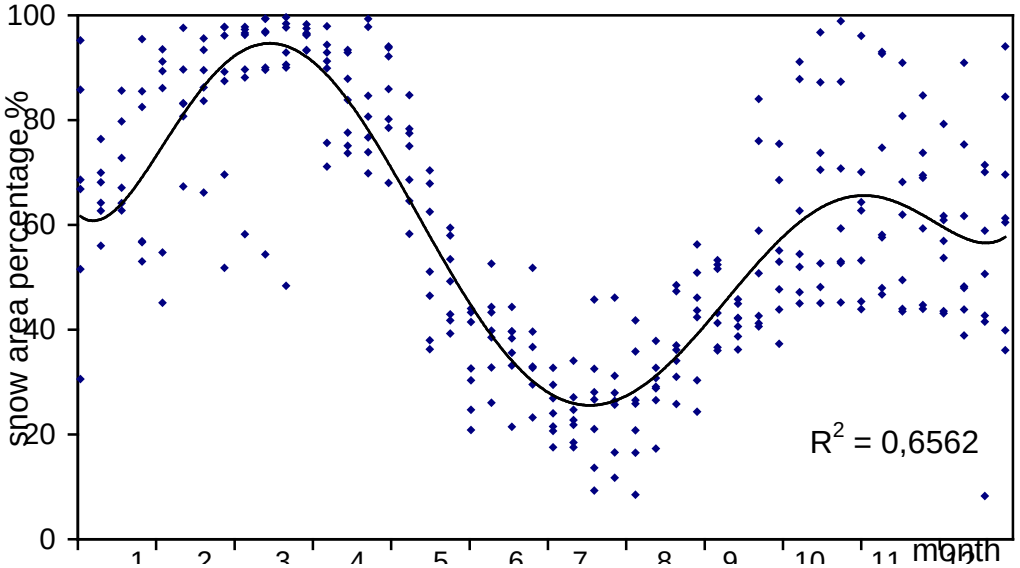


Figure 2. 2 Scatter plot of snow cover percentage in seven years, point data from MODIS snow products and the solid line is the estimation of tendency line, the model is six degree of polynomial function with R2 0.6991

In summer time, the snow cover percentage in the Langtang Khola catchment is in its minimum while the snow cover area percentage in March is the maximum in whole year with almost 94%. However, at the end of March, the snow starts melting resulting in a gradual decrease in snow cover percentage. The snow percentage decrease continues untill the end of July, arriving at the bottom for the whole year with the percentage of 21%. From August onwards, the snow freezing rate is higher than melting rate so the snow covers begins to increase. The snow cover percentage increase extends to winter time.

In winter time, it is colder than in summer. The precipitation is mainly in the form of snow for this catchment. But during annual dry autumn time in October and December, the precipitation is low, so snow cover decreases a little. After this dry period, the snow cover would increase again. Therefore, in the whole annual time period, in terms of snow cover, there are always two peaks (one is in March and another is in December) and a minimum in July. The solid line in figure 2.2 reflects the snow cover change within a year precisely.

2.3.2 Between-annual analysis

When analyzing time series of snow cover data in the Langtang Khola catchment, the figure 2.3 can be obtained. This figure contains annual information of snow cover monitoring. In the seven years (from 2000 to 2006), the snow cover area percentage fluctuates with annual year time. From the estimation of tendency line in figure 2.3, there is annual cycle trend. The snow cover percentage fluctuates over annual time, which is sensitive to the local climate change.

From the figure 2.3, there are always two peaks in winter and one minimum in summer melting time. To describe the changing patterns of snow cover, the ratio between the sum of maximum and minimum snow cover area to the average level is calculated. This ratio reflects the relative spread of the annual snow cover distribution to the annual mean area. Exactly, if the value is larger than 2, the annual snow cover spreads wider. While if the value is lower

than 2, the distribution of annual snow cover is narrower. The standard deviation also reflects the spread of the snow in a year. The statistics is shown in table 2.1. The table shows us that, the minimum snow cover area in 2003 is lowest in all the seven years and the snow cover area maximum value is in 2005.The biggest snow cover area annual difference is in 2005, with the largest deviation. In 2000, the annual snow cover distribution spreads widest in terms of annual mean snow cover area.

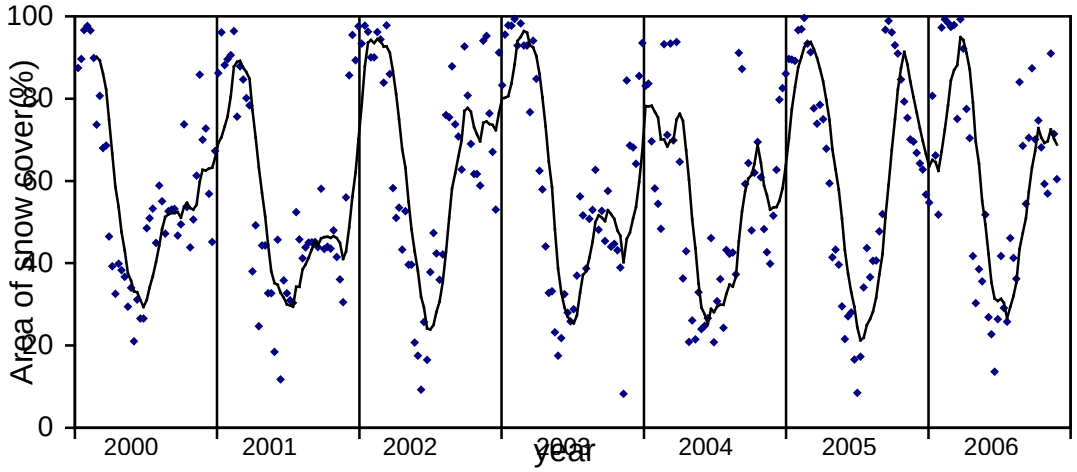


Figure 2.3 Annual snow cover area percentage based on MODIS snow products from 2000 to 2006. Points are snow cover data and solid line is the estimated tendency line using auto regressive integrated moving average model.

In 2003 and 2005, there are second peak points. This is because, in the two years, temperature in autumn time is quite low and, at the same time, the precipitation is high enough to accumulate large snow in autumn.

Table 2.1 Some statistics of annual snow cover. The unit for max and min column is km^2

year	max	min	standard deviation	(max+min)/average
2000	1869	402.75	400.78	2.15
2001	1845.25	224.5	426.47	1.99
2002	1872.25	177.5	499.45	1.65
2003	1902.75	157.75	504.05	1.82
2004	1795.25	397.75	433.93	2.09
2005	1906.75	162.5	511.41	1.68
2006	1901	261	452.30	1.83

2.3.3 Response to climate conditions

Snow covers fluctuate with time in the Langtang Khola catchment. The snow cover trends of inter- and between- annual are related with local climate conditions. The response of snow cover change with time is obviously related to local temperature and precipitation because precipitation directly determines the snow water accumulation and temperature directly determines in what forms (snow or rainfall) the precipitation is. In this part of research, the responses of seasonal snow cover to local temperature and precipitation in the Langtang

Khola catchment are analyzed. The temperature and precipitation data are daily in the seven years. By drawing seasonal snow cover tendency line together with temperature and precipitation respectively, we can obtain figure 2.4 and figure 2.5.

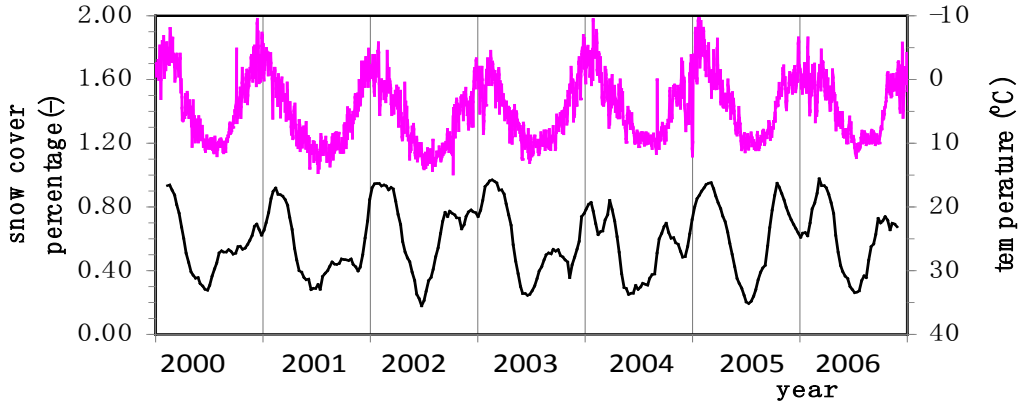


Figure 2.4 The response of snow cover area change to the local temperature. Points are temperature data and the solid line is tendency line of snow cover change.

In terms of the response of seasonal snow cover to temperature, the figure 2.4 shows that when snow covers area increases, the temperature decreases. Generally, when snow cover reaches the minimum value, the temperature always is highest. Sometimes there is a small lag time that when temperature is highest, the minimum snow cover area will be in a short time. During the lag time, the snow is still melting, with a higher rate than that of refreezing. The response to temperature can explain peak and minimum value because the two positions are dominated by local temperature. But for second peak position in each year, precipitation is the main cause.

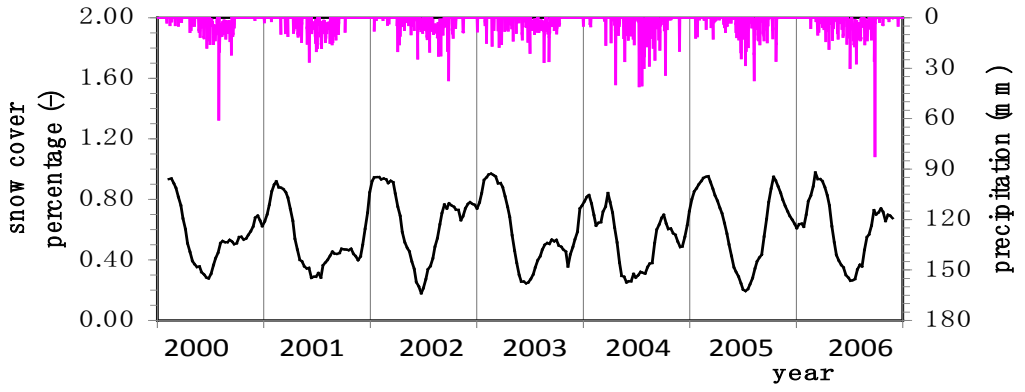


Figure 2.5 The response of snow cover area change to the local precipitation. Bars are precipitation data and the solid line is tendency line of snow cover change.

The figure 2.5 reflects the response of seasonal snow cover to precipitation. Each year, a short period with no precipitation is in end of autumn and winter. When the air temperature is lower than the critical temperature of snow melting, precipitation begins influencing snow accumulation. No or low precipitation results in snow cover decreasing. This is the reason why there is a second peak value in snow cover percentage trend. While if air pemperature is

higher than snowmeting critical temperature (almost the same with snow fall), the precipitation is in the form of rainfall and the rainfall would wash snow cover area resulting in snow storage decreasing. Therefore, seasonal snow cover change is dominated by local temperature and precipitation together.

2.4 Discussion and conclusion

The snow cover between 2000 and 2006 in the Langtang Khola catchment has been monitored successfully based on MODIS snow products. The results showed a seasonal variation in snow cover with two peaks and one minimum in snow cover percentage trends. The maximum snow cover area is about 1800 km^2 (94% in the catchment) in winter time, while in summer time, the bottom is only about 200 km^2 (10% in the catchment). The reason of this seasonal variation of snow is explained by temporal variation in local temperature and precipitation. In summer time, the temperature is above 0 celsius and snow metling occurs. During winter time with quite low temperature, the precipitation is mainly in the form of snow so the snow cover area expands graduately. But in many winter time, there is a drought period without preciptation, which causes the snow cover to reduce. This leads to a second snow cover area percentage peak. After the dry period, snow recession may end, snow cover increases again,which leads to peak area.

This research gives an analysis of snow cover area change with time. Inter- and between-annual analysis methods are two different perspectives for snow cover time monitoring. Some interesting trends would be drawn at last. The response of the change trends to climate conditions (mainly precipitation and temperature) provides explanations for the snow cover percentage trends.

3 Discharge modeling from snow melt runoff model and from remote sensing snow cover data

3.1 Introduction

Mountainous snow cover is known to exert a strong influence on climate (Vavrus, 2007). Snow melting forms a site of marked physical-hydrological processes. Its distribution and temperature have a direct bearing on the hydrological behavior of streams and any change in the snowmelt pattern is considered to be indicative of climate change. Seasonal snow cover is dominant source of fresh water in snow-fed streams (Gupta, et al, 2007). Snow melting during winter and spring seasons on mountainous regions is process of considerable interest to water users in the adjacent arid valleys because they have a direct bearing on the timelines, rate and amount of stream flow in the reminder time of the year for irrigation, power and other purposes (Croft, 1944). The significance of snowmelt modeling is great not only for science development in snow cover but also for water engineering.

Satellite remote sensing observations provide spatial data on snow cover at regular intervals of time and are incorporated in the snowmelt runoff models (SRM) (Martinec et al., 1994, Rango, 1993). With time series of snow cover remote sensing data, modeling discharge can be improved a lot. And the discharge from remote sensing data inverse can be applied to test that from other models with different modeling process. When remote sensing data applied in hydrology model, the modeling performance can be improved (Kite, 1991).

The objectives of this chapter work including: *i) Modeling snow movement and snow melt processes*. This is based on physical process of snow melt. The model applied in this part is snowmelt runoff model (SRM) developed by Martinec et al, (1975). *ii) Integrating remote sensing snow cover into the SRM model*. MODIS snow products provide abundant information for snow cover and snow melt research. Remote sensing data can be used to improve the model performance. *iii) Comparing the two modeling results of SRM and integrated by remote sensing snow cover*. The two models simulate snowfall, snow melt, and snow melt discharge. Simulation results integrated with remote sensing are better than those of SRM model.

The reminder of this chapter is organized as follows. The next section describes snowmelt runoff model, main modeling parameters and analyzing model performance. The third section inverses snowmelt runoff discharge from remote sensing, meteorological and other observed data. In the last section, the conclusion is given compared the two models.

3.2 Snowmelt runoff model

3.2.1 SRM Model description

The snowmelt runoff model applied in this study is Snow Runoff Model (SRM). This model has been presented since 1975 (Martinec, 1975 and Martinec, et al, 1994) and bas been widely applied in different catchments. SRM is a conceptual, deterministic, degree day hydrologic model used to simulate daily runoff resulting from snowmelt and rainfall in mountainous regions (Immerzeel, et al, 2009 and Martinec, et al, 1994). The distributed hydrological processes can be shown in the following figure. Generally, distributed model is constituted by several hydrological processes including precipitation, interception, evaporation, infiltration and runoff production.

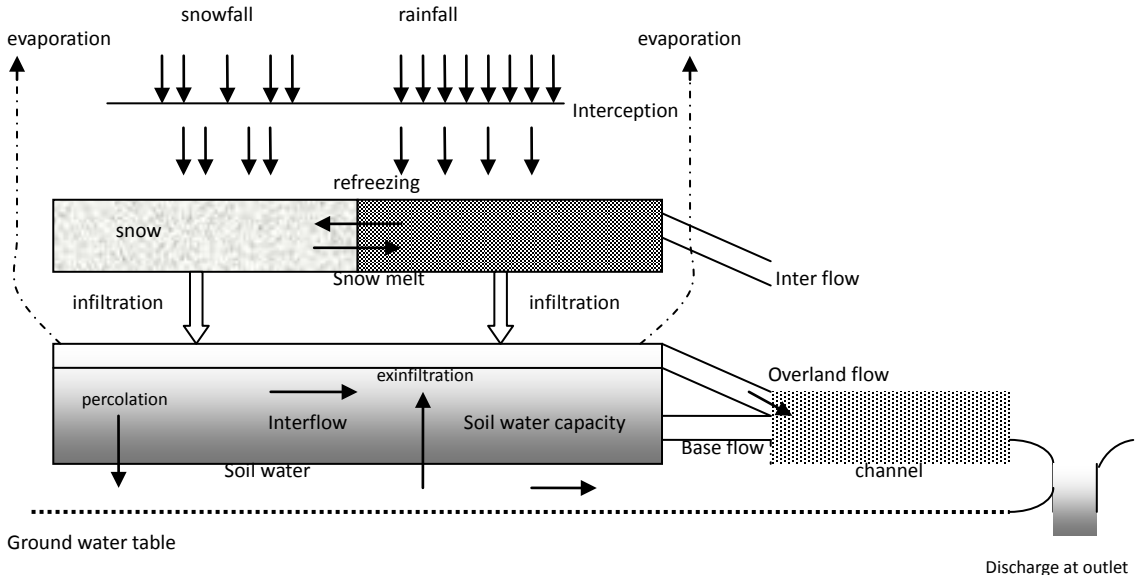


Figure 3. 1 Main hydrological processes in distributed model

SRM requires daily temperature, precipitation and daily snow covered area values as input parameters. SRM model is conceptual, distributed hydrological model to simulate runoff generation especially for snow melt runoff flow because in this model snow melt water is individually modeled.

3.2.2 Main processes

A catchment is divided into a number of grid cells. For each of the cells individually, daily runoff is computed through application of the version of the SRM model. The main processes are shown in figure 3.2.

3.2.2.1 Snow routine

Precipitation enters the model via the snow routine, which is shown in figure 3.3. If the air temperature, T_a, is below a user-defined threshold T_x (≈ 0 °C) precipitation occurs as snowfall, whereas it occurs as rainfall if $T_a \geq T_x$. If precipitation occurs as snowfall, it is added to the snow component. Otherwise it ends up in the rainfall, which represents the liquid water content of the snow pack. Between the two components of the precipitation, interactions take place, either through snow melt (if temperatures are above a threshold T_s). The respective rate of snowmelt is:

$$F_m = C_x\left(T_a - T_s\right); T_a > T \qquad (3.1)$$

where F_m (mm) is the rate of snow melt, and C_x (mm/ °C)is user defined model parameters.

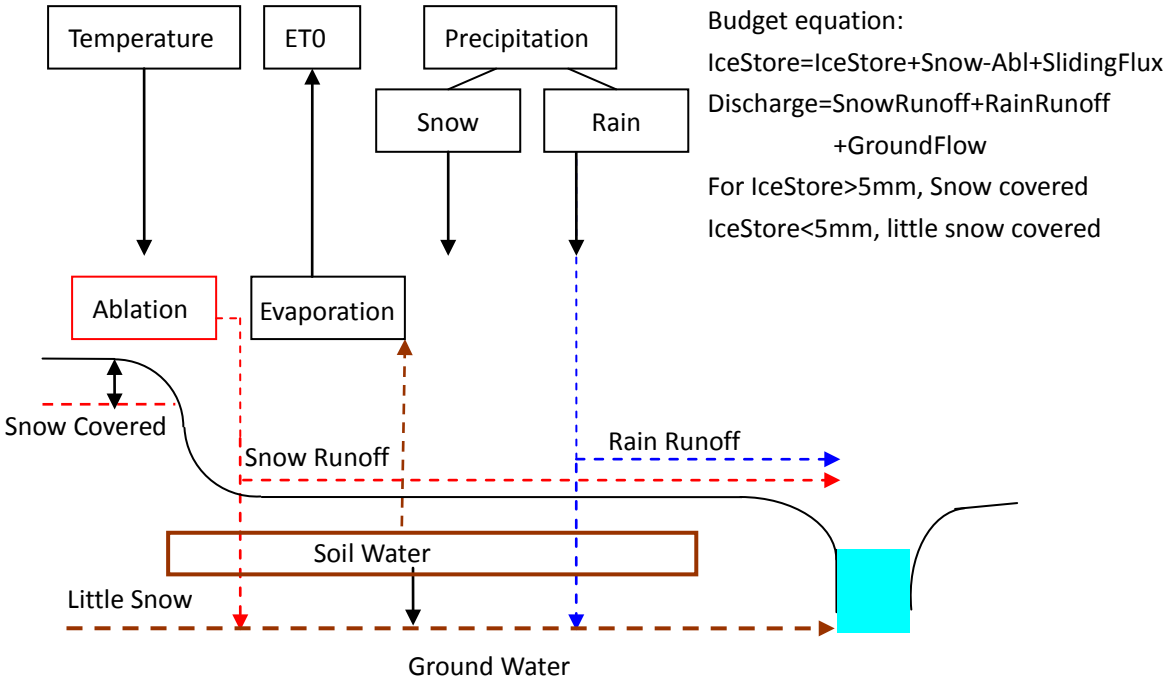


Figure 3. 2 Schematic view of the relevant components of SRM model

The air temperature, T_a, is related to measured daily average temperatures, with additional compensation for vertical gradients (a constant temperature decrease of T_L °C per 1000 m. is assumed to exist). In the RSM model, elevation differences within the catchment are represented through a distribution function which makes the snow module distributed. In the modified version that is applied here, the temperature, T_a, is represented in a fully distributed manner, which means for each grid cell the temperature is related to the grid elevation.

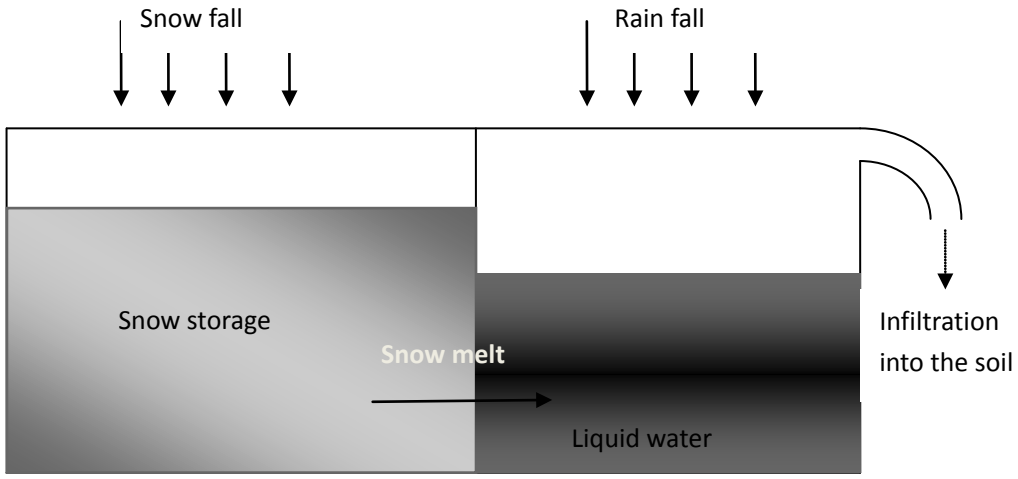


Figure 3. 3 Snow routine of SRM model process

3.2.2.2 Ice sliding

For certain thicknesses of a glacier, shear stress exerts at the rock bed on which ice or glacier spreads thereby causing ice sliding (Weertman, 1957 and 1964). The sliding condition is expressed by τ (N/m^2).

$$\tau = \rho g h \sin(slope) \tag{3.2}$$

where ρ (kg/m^3) is the ice density, g (m/s^2)is gravidity h (m) is ice storage and slope is from DEM in small cell. If τ is above user-defined threshold τ_0, ice sliding may happen and the sliding percentage is defined:

$$\alpha = \frac{\sqrt{\tau - \tau_0}}{\sqrt{v} \cdot R} \tag{3.3}$$

v (-) is bedrock roughness and R (-) is material roughness coefficient.

So there is a part of ice flux transported downstream. The ice storage should be updated with four components shown in figure 3.4 and the equation to update ice storage is:

$$\begin{cases} S_I = Flux_{in} + P_s - M_s \quad Flux_{out} \\ Flux_{out} = \alpha \bullet S_I \end{cases} \tag{3.4}$$

S_I (m) is updated ice storage, which is below maximum ice storage Sm (m). Ice flux (m) is an important component of ice storage, causing the change of DEM update.

$$\Delta DEM = Flux_{in} + P_s - M_s - Flux_{out} \tag{3.5}$$

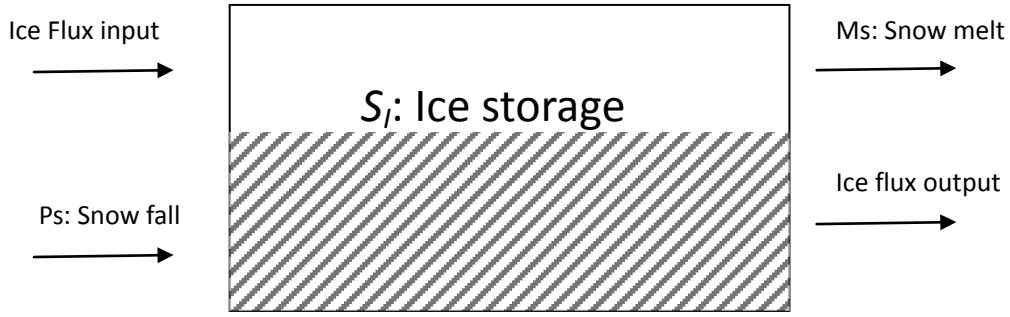


Figure 3. 4 Four components of ice storage flux

3.2.2.3 Hydrological processes

1) Evaporation

Evaporation at a potential rate occurs from that fraction of the basin covered by streams, lakes and riparian vegetation. Evapotranspiration from the remaining part of the catchment is determined by the relative water contents. Evaporation is an important output of water

storage.

$$E = K_C E_p \quad (3.6)$$

E_p (mm) is potential evaporation and K_c (-) is actual coefficient of evaporation.

2) Inter flow

When there is shower into this catchment, inter flow may be generated if rainfall is heavy enough. In this research, SCS runoff is applied to calculate flow Q_I (m/s^3):

$$Q_I = \frac{\sqrt{P_R - 0.2S}}{P_R + 0.8S} \quad \text{when } P_R > 0.2S \quad (3.7)$$

S (m/day) is retention parameter for SCS runoff and P_R (mm) is rain fall. Here, S can be estimated from:

$$S=0.001*25.4*((1000/CN)-10) \quad (3.8)$$

CN (-) is curve number for SCS runoff.

3) Soil saturation

If the surface is not fully covered by ice, water can percolate into soil to saturate soil moisture. From the equation of water budget, soil water can be estimated:

$$S_s = P_R - Q_I - E \quad (3.9)$$

S_{sat} (m) is saturated soil moisture, which is maximum capacity of soil water retention.

4) Snow melt flow

The snow begins melt when the air temperature above T_s (0 ^{o}C) causing snow melt flow. Snow melt flow is a main flow in snow covered areas. The flow can be calculated:

$$Q_M = c_a F_d F_m \quad (3.10)$$

F_m (m) is the rate of snow melt, F_d (-) is degree of day factor for glacier melting and c_a (-) is fraction of total ablation that results in actual runoff.

5) Base flow

Base flow for snow covered areas is mainly from recharge between soil moisture and percolation. When soil is saturated, recharge may be formed:

$$R = P_R + (1 - c_a) F_d F_m - Q_I - E \quad (3.11)$$

R (m) is recharge.

Therefore, base flow can be calculated:

$$Q_B = Q_B' e^{-r_c} + R(1 - e^{-r_c}) \quad (3.12)$$

Q_B' (m/s^3) is previous base flow and r_c (day^{-1}) is recession constant for baseflow.

6) Discharge production

Contribution to the flow component Q_T (m/s^3) is given by snow melt flow, inter flow and base flow.

$$Q_T = Q_I + Q_B + Q_M \quad (3.13)$$

And then, all of the three flows may influx downstream at the outlet of this catchment.

After flowing into outlet, assume Q (m/s^3) is the total discharge at the outlet of this catchment, then:

$$Q = K_x Q' + (1 - K_x) Q_T \quad (3.14)$$

K_x (-) is the flow recession coefficient and Q' (m/s^3) is discharge at previous time setp.

3.2.3 Experiment

The snowmelt runoff model (SRM) contains two parts in this research. First is to simulate the snow movement. This process is to estimate daily snow cover area and snow storage. And another is hydrologic process including precipitation, snow melting, transpiration, recharge, and discharge accumulation. The flow chart showing model process is figure 3.1. The discharge at outlet includes snow runoff flow, rain fall runoff flow and ground flow recharge. The inputs of this model are daily temperature, precipitation and evaporation and other constants shown in figure 3.5.

1) Meteorological data

The meteorological data in the model are daily precipitation and temperature in the Langtang Khola catchment. Meteorological data are main inputs for SRM model. They directly influence snow fall, melting, and snow cover distribution.

The precipitation is estimated by the Tropical Rainfall Measuring Mission (TRMM) at fine spatial scales using a calibration based sequential scheme and data from multiple satellites as well as gauge analysis. Firstly, a number of passive microwave sensors aboard TRMM and other satellites are converted to a precipitation estimate. Secondly, an infrared (IR) estimate is generated using the calibrated microwave estimate. Thirdly, the microwave and IR estimates are combined to provide the best estimate at each grid box at each three hour period. The final step in generating 3B43 is the inclusion of rain gauge data. It is highly advantageous to include rain gauge data in combination data sets. Using the gridded precipitation gauge based product of the Global Precipitation Climatological Centre (Immerzeel et al, 2009). Long term temperature patterns were derived using the most accurate global database currently available:

the climate research unit (CRU) dataset. The latest updated version, referred to as the CRU TS 2.1 dataset in this paper, is used to reconstruct the seasonal temperature patterns in the upper Indus basin from 1901－2002 (Mitchell & Jones, 2005). The CRU TS 2.1 is a set of monthly climate grids which are constructed for nine climate variables and interpolated onto a 0.5° grid and provide best estimates of month-by month variations.

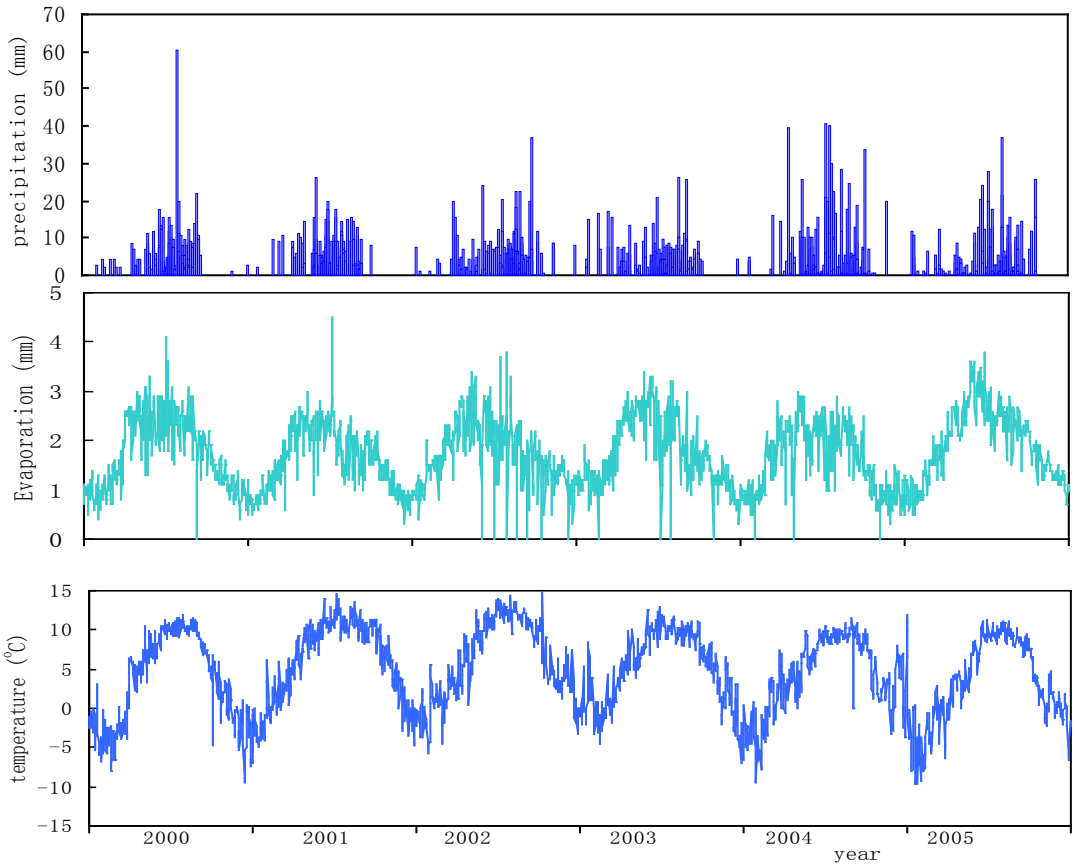


Figure 3. 5 Temperature and precipitation trends in Langtang Khola catchment from 2000 to 2005

The precipitation, evaporation and temperature in the Langtang Khola catchment is cyclic annual from 2000 to 2005. The model results are related tightly with this input database.

2) Model results

The discharge results of RSM model from 2000 to 2005 at outlet of the Langtang Khola catchment is shown in figure 3.3. The discharge fluctuates during these six years, which is tightly related with precipitation. In summer time, with high temperature and abundant precipitation, the discharge at outlet arrives peak point. While, in winter, with low temperature and little precipitation, the discharge is also very low.

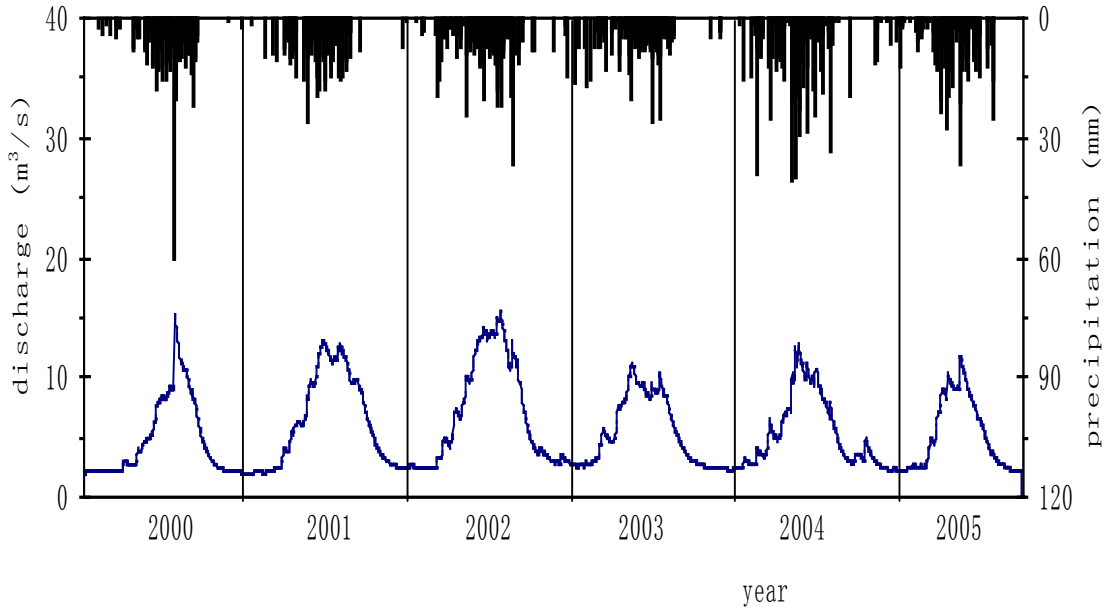


Figure 3. 6 Discharge at outlet based on SRM model in the Langtang Khola catchment from 2000 to 2005

The discharge curve in figure 3.6 is related with temperature. This is mainly because temperature directly influences snow melting hence determining discharge. What is more, when temperature is higher than 0 Celsius, the precipitation is snow fall, which confluxes to the outlet of this catchment. But, when there is time lag between precipitation peak and temperature peak during a year, there would be more discharge peaks, reflecting discharge peaks in 2001 and 2003. In 2000, there is only one discharge peak point in summer time because, after temperature peak, there is a long period with no or little rainfall.

3) Snowmelt runoff discharge

This model simulates the snow melt runoff discharge at outlet in this catchment. The discharge of snow melt runoff and its response to temperature and precipitation are shown in figure 3.7.

Snow melt runoff discharge in this catchment is related with the local temperature. When air temperature is above ice melting temperature snow begins melting and then snowmelt runoff discharge is generated. In summer time, snow melt discharge is larger than that in winter time due to the temperature difference. More or less, the annual peak snowmelt flow discharge appears around temperature peak. During winter period with temperature below zero ^{o}C, the snowmelt runoff discharge is very low even zero in many times.

In the other hand, precipitation is the main incoming snow storage, which is the snow source for melting. Especially when, if the air temperature in this catchment is very high, the rate of snow melt is also very high, the snow storage is the main influence factor. The figure 3.7 reflects that before peak snowmelt runoff discharge, there are heavy snow fall, which is the snow stored for melting. In 2000 and 2005, the precipitations in winter time are very low, so the snow storages for melting in summer are also few, causing snow melt runoff drought in

the year of 2000 and 2005. In 2002, the snow fall in winter is abundant and therefore, the snowmelt runoff discharge is large as well.

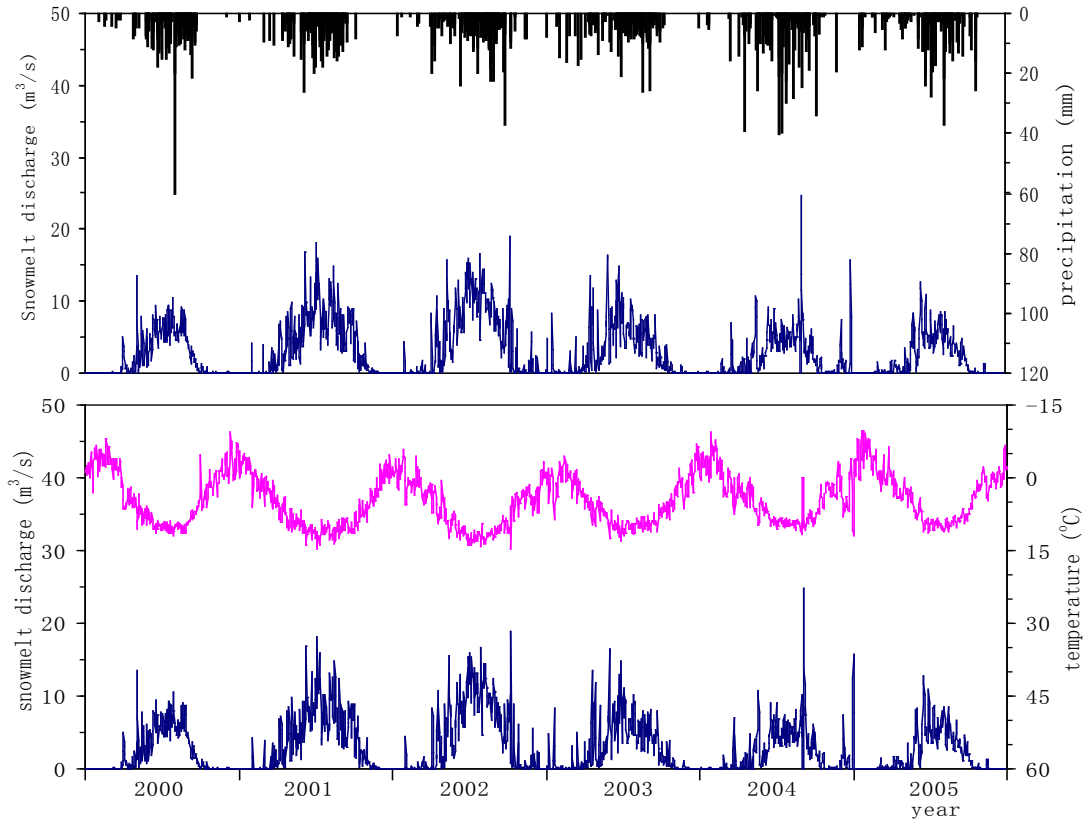


Figure 3. 7 Snow melting runoff discharge at outlet in the Langtang Khola catchment

3.3 Model integrated with remote sensing snow cover data

3.3.1 Integration process

Remote sensing data provides real-time monitoring snow cover for the model. If remote sensing snow products are applied into modeling, the model results can be improved at some extent (Kite, 1991). In this part of research, MODIS snow products are integrated into SRM model to update the real time snow cover information. The snow cover in SRM model is based on snow dynamics simulating using snow fall, melt and snow movement. There is initial snow cover map at the beginning. Then every day snow fall and snow melt may happen in this area and, in the same time, snow cover may move with direction of DEM to update snow cover. After a long run time, the uncertainty of snow cover is very large, which would influence modeling results. If each eight-day remote sensing snow products are applied into SRM model, the uncertainty will be reduced. The time resolution of remote sensing snow cover product is eight days.

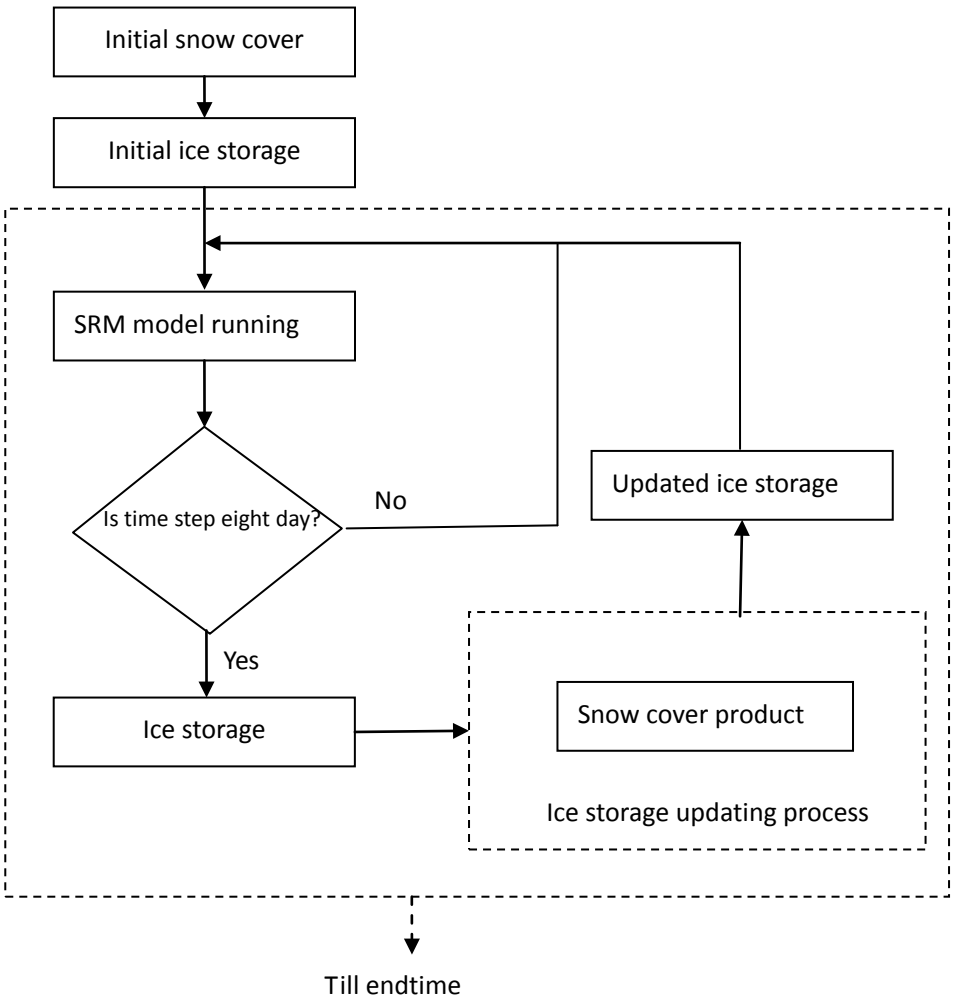


Figure 3. 8 The process of discharge modeling based on remote sensing snow cover product

The model integration processes of SRM model with remote sensing data are shown in figure 3.8. Each time when there is remote sensing snow data, the ice storage will be updated by a new hydrological process “Ice storage updating process”. The output of this process is a new ice storage map derived from remote sensing snow cover map and then the new ice storage map is inserted into the SRM model to simulate snow melt processes. The ice storage updating process is shown in figure 3.9. In the updating process, if ice storage from previous time is not zero (snow covered) and snow cover from remote sensing data is not zero (snow covered), then the new ice storage is copied from previous period. If snow cover from remote sensing data is zero (non-snow covered), then the new ice storage is set zero. But if snow cover from remote sensing is snow covered and ice storage from previous period is zero, then this snow cover is regarded new snow cover because this cell is newly formed but the SRM model does not simulate the snow and minimum ice storage value is set in this case. After this process, new ice storage map is formed and then inserted back to SRM model.

The minimum ice storage in figure 3.9 is threshold (S_t) for ice recognition. In this case, the threshold is 5 mm. The calculating process of the new ice storage is:

$$S_U = \begin{cases} \min(S_t, S_I); \text{ For snow cover} \\ 0; \text{ For non-snow cover} \end{cases} \quad (3.15)$$

S_U is updated snow cover, S_t is the threshold for ice recognition and S_I is ice storage of SRM model.

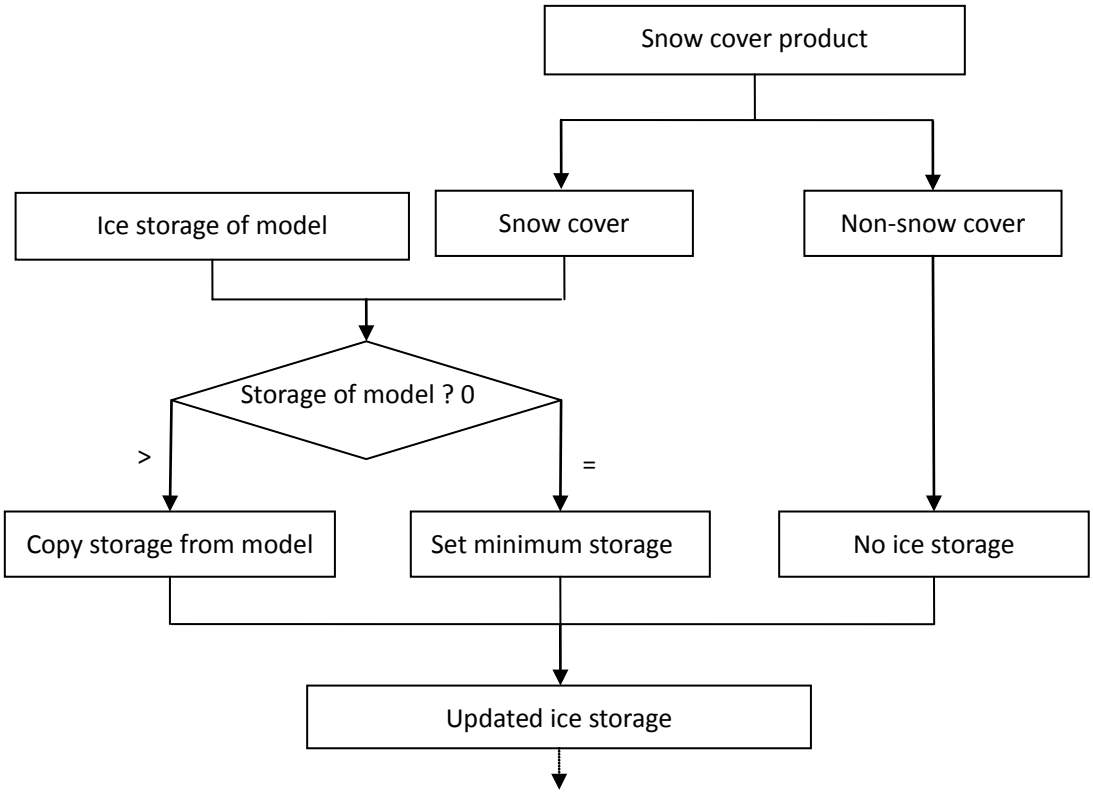


Figure 3. 9 Ice storage updating process in discharge modeling from remote sensing snow cover product

3.3.2 Discharge modeling

In the SRM modeling process, the remote sensing snow cover products are integrated into model to update ice storage every eight days. The outputs of the model are eight-day discharge time series. The results are shown in figure 3.10.

Figure 3.10 reflects the discharges from both models and the difference between them. Both the discharge series shows the discharge seasonal trends. In winter time, with low air temperature the discharges from both models are very low. But with the temperature increase, more snow begins melting resulting in high discharge at outlet. When in summer time, the discharge arrives at high point. The peak discharge of this catchment also appears in summer time, in which snow melt with peak rate. However, there are some differences between the discharge curves of both melting models due to the updated ice storage information difference during model running time. The discharge from SRM model is generally underestimated because in SRM process ice storage is updated not suddenly but gradually, which may not always match with real hydrological environment.

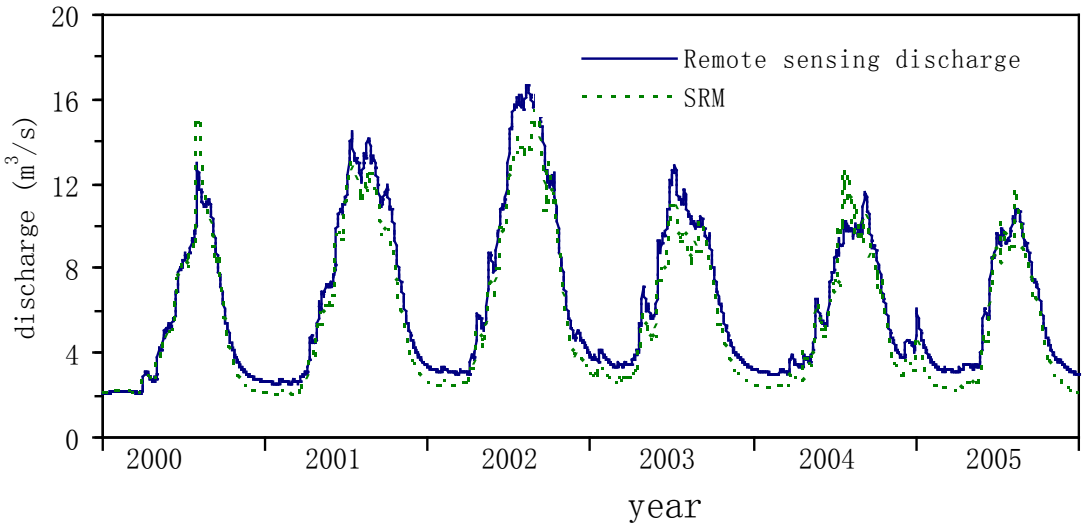


Figure 3. 10 Discharge at outlet of the Langtang Khola catchment based on remote sensing snow product

Precipitation and temperature are two core influencing items for snow melt discharge because the snow cover is mainly from local precipitation and low temperature and snow melting is only determined by temperature and ice storage. Therefore, if more accurate precipitation, temperature and ice storage are applied into snow melt hydrological models, the results can be improved at a large extent. This is also the reason why many observations techniques are applied into mountainous areas to monitor more accurate data sets.

3.4 Evaluation of model performances

To evaluate the model performances, the simulated discharge and observed discharge are plotted in the same figure shown in figure 3.11. The diagonal line in the figure is true relationship between simulation and observation; but actually there are some differences between simulation and observation. The R-square in the figure shows the correlation coefficient between the model simulation and observation. For SRM model performance, the R-square is 0.2457, which is a low correlation coefficient for the simulation. Once remote sensing snow cover data has been integrated into the SRM model, the model performance has been improved, with R-square of 0.5615. The scatter plot figure also shows the points for SRM model performance expand wider than model integrated with remote sensing data.

Average square root error R (m^3/s) of simulated discharge can be used to quantify the model performance. R can be calculated using the formulae:

$$R = \sqrt{\frac{\sum_{i=1}^{N} (Q_S^i - Q_O^i)^2}{N}}$$

where Q^i_s and Q^i_O are simulated discharge and observed discharge respectively.

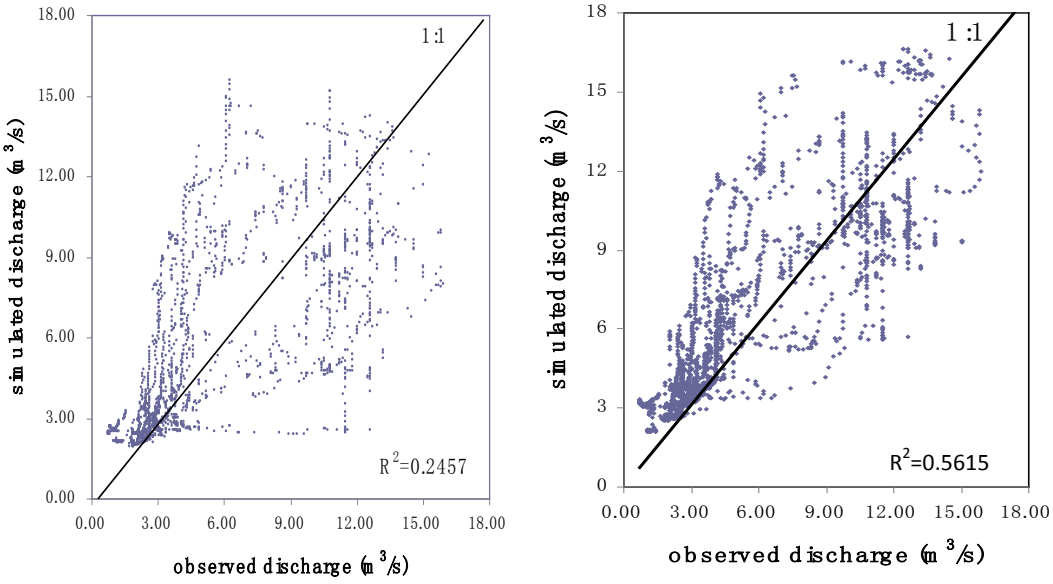
The calculation results are in table 3.1. The table shows that RSM model integrated with

remote sensing data performs better than RSM model alone.

Table 3.2 Model performance results

Model	R
RSM	3.10
Integration	2.42

The accuracy analysis shows that SRM model simulates snow discharge in the Langtang catchment and once remote sensing snow cover data are integrated into the model the performance is better. When remote sensing data is integrated into the model, the model performance is improved. The R of SM alone model and integration with remote sensing data performance are 3.10 and 2.42 respectively.



A: RSMmodel performance

B: RSM integration with Remote sensing performance

Figure 3. 111 scatter plots of simulated and observed discharge

3.5 Discussion and conclusion

Snowmelt Runoff Model (SRM) has been applied into the Langtang Khola catchment to model snowmelt runoff and discharge. SRM model is a conceptual, determinant and distributed hydrological process simulating snow melt, soil moisture, and snowmelt runoff. The total runoff in this model includes three components: inter flow, snow melt flow and base flow, which is simulated separately.

The remote sensing snow cover data have been integrated into SRM model to update ice storage maps for every eight days. The ice storage updates in SRM model are from the estimation from ice flux budget. The results show that the integration technique with remote sensing snow data has been successfully applied, which is important for researchers to integrate real-time snow cover observations into SRM modeling to improve modeling accuracy.

4 Assimilating remotely sensed snow cover into the snowmelt runoff model

4.1 Introduction

Large difference exists between discharges of the SRM model and integrated with remote sensing snow cover data. There are some erroes in remote sensing snow product data because there are gaps in remote sensing data that cannot be identified covered by cloud. The uncertainties of remote sensing snow cover are shown in figure 4.1. On the contrary, in the SRM model, snow data are simulated from snow processes including snow fall, snow melt and snow movement. The whole catchment of precipitation and temperature data is interpolated resulting in model input uncertainties. Therefore, the outputs of the model are with uncertaity. Data assimilation techniques can be applied to reduce the uncertainties of discharge and furthermore improving model accuracy (Sujay, et al, 2008).

The basic idea of data assimilation is to quantify errors in both the hydrological model and observations, and update hydrological model states in a way that optimally combines model states with other data sources (McLaughlin, 2005). Data assimilation is commonly used in ocean (Dunlap, et al, 2010) and meteorological sciences (Tsuyuki, 1996) to improve the predictive power of short-term weather forecasting models and to provide the formal framework for dynamic analysis of oceanographic models (Troch, et al, 2003). Recently, many hydrologists have begun to use this method in the hydrological application because, in hydrological models, many environmental parameters are derived from different data sources (inculding model states, model outputs, observation et al.). Data assimilation can narrow the uncertainties between different source data (McLaughlin, 2005). The assimilation method provides a means of integrating observed data into model predictions. In general terms, data assimilation is a quantitative method for inferring the state of the earth from all available sources of information. A distinctive characteristic of data assimilation is its use of spatially distributed models to merge measurements of different types, accuracies, and resolutions (McLaughlin, 1991).

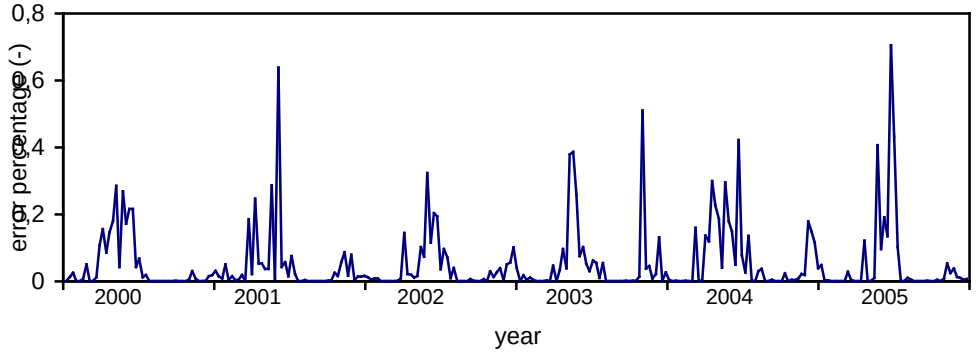


Figure 4. 1 Accuracy errors of snow cover products in Langtang Khola catchment

With remote sensing techniques development, data assimilation became more important in modeling. Hydrologic remote sensing can provide important information about land surface conditions, including evaporation (Chen et al., 2005), surface soil moisture (Pierdicca , et al., 2010), snow water equivalent, snow cover, and land surface temperature(Giraldo, 2009 et al.). Many of these variables would be derived from hydrological model,which are regarded as model states. To quantify the uncertainty of model states, hydrological data assimilation can be applied. In data assimilation, there are many methods to accomplish including techniques such as Kalman filtering, Newtonian nudging and optimal interpolation (Kitanidis, et al, 1980). One of the early hydrologic data assimilation methods is the application of the linear Kalman filter. In the case of nonlinear, hydrological model the data is rendered in state-space form and by assuming that the model state variables are differentiable the Extended Kalman filter (EKF) can be used (Qin, et al. 2008). EKF updates a model state estimate whenever an observation is available, and at the same time takes into account errors in model dynamics and observations (Walker and Houser, 2001). It integrates Monte Carlo sampling from an ensemble of model states to propagate the error information, whereas the Kalman filter and EKF use a dynamic equation to explicitly propagate estimation errors. In the ensemble Kalman filter (EnKF), model error estimates are produced by assuming that the ensemble mean is “truth” and computing the variance of the differences between each ensemble member and the ensemble mean (Clark et al, 2008). Each individual observation is then updated based on the relative error in both the model and observations. If prior probability density is not Gaussian distributed, the posterior is determined not only by the mean and variance of the prior density, but also by the whole density(Weerts and EI Serafy, 2006 and Karssenberg, et al, 2009). Particle filter does not have the assumption that prior probability density functions of model states are Gaussian.

The reminder of this chapter is organized as follows. The next section describes the theories of data assimilation techniques including Monte Carlo simulation, Ensemble Kalman Filter and Particle Filtering. The third section of this chapter is the application of three data assimilation methods into the Langtang Khola catchment to assimilate discharge from remote sensing snow data into the SRM model. The last section is the conclusion.

4.2 Data assimilation theory

4.2.1 Monte Carlo simulation

Due to the uncertainty in inputs and model parameters of the SRM model, the modeling discharge results are also with large uncertainties. The uncertainties in modeling processes can be quantified with stochastic methodology and then the uncertainties can be reduced (Beck, 1987). For different model inputs and parameters, different results are simulated in the modeling. Monte Carlo simulation provides an analysis method for model input and parameters propagation in modeling and then obtains model realizations. From the model realizations, mean and variance can be calculated. Monte Carlo technology allows model input and parameters to be a specific probability distribution. The main purpose of Monte Carlo simulation is to get the model outputs distribution.

In Monte Carlo simulation, there are a large number of realizations to randomize determinant model. Figure 4.2 shows the process of Monte Carlo simulation results. These realizations are

assumed to fit Gaussian distribution, with different mean and variance. At time t_1, the estimated mean and variance are from the realizations and then other parameters can be calculated using this mean and variance. With mean and variance, model uncertainties are quantified and furthermore can be narrowed as much as possible. The model variance is related with the realization density. The variance at t_2 is lower than that at t_1 because at t_2 the spread in the realizations is less than at t_1. So, if the realizations spread too wide, wider than what we can accept, data assimilation techniques can be applied to narrow this distribution such as particle filter.

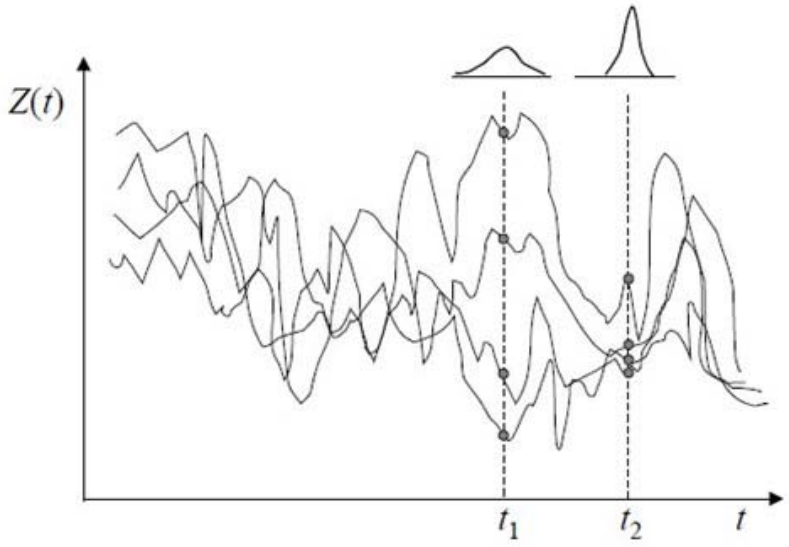


Figure 4. 2 Schematic view of Monte Carlo simulation performance (Bierkens, 2009)

The advantage of Monte Carlo analysis is its general applicability and that it does not impose many assumptions on probability distributions and correlations and that it can be linked to any model code. The key limitation is the large computational demand for intensive models and the huge amount of outputs that are not always straightforward to analyze.

4.2.2 Ensemble Kalman Filter

The Monte Carlo simulation technique quantifies the model uncertainty but never narrows the uncertainty. When there are some observations applied into the stochastic model, the uncertainties of the model can be narrowed to some extent. The figure 4.3 shows how to narrow model uncertainty.

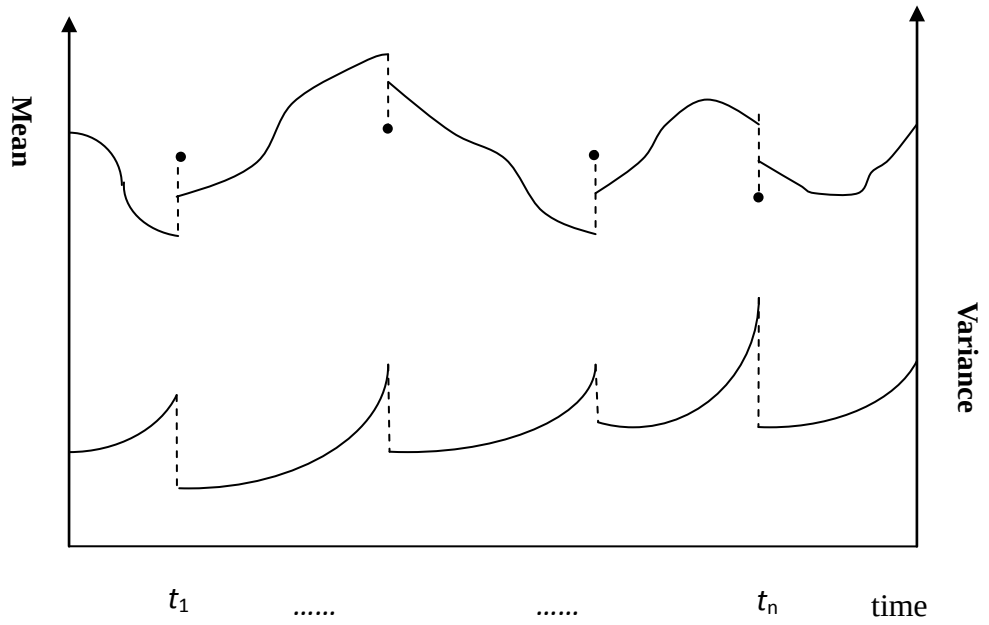


Figure 4. 3 Schematic view of data assimilation performance. Dots are observations line is modeling outputs and corresponding variance.

Once observations are applied into Monte Carlo simulation to narrow model uncertainty, this technique is called data assimilation. The basic idea of data assimilation is to quantify errors in both the hydrological model and observations, and update hydrological model states in a way that optimally combines model background with observations (McLaughlin, 2005). The main method to implement data assimilation is Kalman Filter. The simple Kalman filter is linear Kalman filter (LKF) when the model is linear. But when the model is not linear, some advanced Kalman filter techniques are applied including Extend Kalman Filter and Ensemble Kalman Filter (EnKF). EnKF is based on Monte Carlo simulation to obtain model ensembles. By making statistics of large number of ensembles, many applications can be calculated. The uncertainty of this method is smaller than that of Monte Carlo simulation because observations or other data sources are integrated into the model.

1) Recursive Bayesian filtering

Recursive Bayesian filtering is to estimate probability density function recursively over time using models and observed data. EnKF and particle filter are the optimal solutions for the Bayesian filtering (Moradkhani et al., 2005) and the difference between the two techniques relies on how to obtain the prior probability density function of states.

The true state *X* for Bayesian filtering is assumed to be a Markov process:

$$\mathbf{X}_n = M(X_{n-1}, v_{n-1}) \tag{4.1}$$

M is system model function and *v* is model noise.

And observations *Y* are linked with state *X* with measurement function.

$$Y_n = H(X_n, u_n) \tag{4.2}$$

In this equation, H is measurement function and u is measurement noise. The set of measurements on time step is $D_n = \{Y_n: n=1, \ldots, k\}$.

The core requirement is to structure the conditional probability density function: $p(X_n | D_n)$. in the whole process, the prediction and update are recursively applied to obtain the series of the probability density function. If conditional probability $p(X_n | D_{n-1})$ is available, then $p(X_n | D_n)$ can be updated via Bayesian theory:

$$p(X_n \mid D_n) = \frac{p(Y_n \mid X_n)\, p(X_n \mid D_{n-1})}{p(Y_n \mid D_{n-1})} \tag{4.3}$$

And the probability distribution associated with the predicted state is the sum (integral) of the products of the probability distribution associated with the transition from the (n-1)-th timestep to the n-th and the probability distribution associated with the previous state, over all possible X_{n-1}:

$$p(X_n \mid D_{n-1}) = \int p(X_n \mid X_{n-1}) p(X_{n-1} \mid D_{n-1}) d\ X_{n-1} \tag{4.4}$$

And the denominator of equation 4.3 $p(Y_n|D_{n-1})$ is constant relative to state X:

$$p(Y_n \mid D_{n-1}) = \int p(Y_n \mid X_n) p(X_n \mid D_{n-1}) d\ X_n \tag{4.5}$$

Equation 4.4 is update equation, which is used to obtain the posterior probability density function of state X from previous prior probability density function.

2) EnKF

Ensemble Kalman Filter (EnKF) is used to solve Bayesian filtering. The Ensemble Kalman Filter can be formalized as follows. Let X_b be the (n_{state}, n_{ensem}) background matrix of model states, where n_{state} is the number of state variables and n_{ensem} is the number of ensemble members.

$$\mathbf{X}^b = (X^b{}_1, \cdots, X_{n_{ensem}}{}^b) \tag{4.6}$$

Where X_1, …,Xn_{ensem} are the background vectors of all model states for each ensemble members before the state update. And the mean of ensemble is:

$$\overline{X^b} = \frac{1}{n_{ensem}} \sum_{i=1}^{n_{ensem}} X_i{}^b \tag{4.7}$$

So the anomaly of ensemble is:

$$X'^b = (X^b{}_1 - \overline{X^b}_1, \ldots, X^b{}_{n_{ensem}} - \overline{X^b{}_{n_{ensem}}}) \tag{4.8}$$

From the above equation, the estimate of the model error covariance is directly from the anomalies vectors:

$$P^b = \frac{1}{n_{ensem}} X^{'b} X^{'bT} \quad (4.9)$$

And then, the model states can be updated via the error covariance, the equation is:

$$x_i^a = x_i^b + \mathrm{K}(y_i - Hx_i^b) \quad (4.10)$$

Where, $\mathrm{K} = P^b H^T (HP^b H^T + R)^{-1}$

And each ensemble member is updated separately. And x^a is analysis of model states after the update; y_i is the column vector observations. And H is operator converting the model states to observation space. K is gain in Kalman filter, and R is error covariance matrix of observation.

In EnKF process, the elements of y_i for each ensemble member are sampled as a distribution with mean equal to zero and variance equal to R. To implement the EnKF, prior error distribution is Gaussian distribution. This assumption is not perfect in this assimilation technique because many geographical applications do not always agree with Gaussian distribution, which may cause uncertainty for the filtering results.

4.2.3 Particle Filter

It is clear that EnKF relies on the assumptions that the prior probability density function (PDF) of model states is Gaussian distribution. This assumption is not always nature for hydrology modeling. Particle filter have the advantage that no assumptions for the form of prior PDF of model states are necessary and that full prior density is used in filtering (Weerts and EI Serafy, 2006). Particle filter is a kind of recursive Bayesian filter based on the Monte Carlo simulation and does not need the assumption of the form of prior PDF of model state. The key idea of the particle filter is to represent the posterior PDF by a set of random drawn samples, called particles, with associated weights (Salamon and Feyen, 2009). With a large number of particles this Monte Carlo characterization approximates the posterior PDF. The posterior density can be calculated:

$$p(X_{0:n} | Y_{1:n}) = \sum_{i=1}^{N} W_n^i \delta(X_{0:n} - X_{0:n}^i) \quad (4.11)$$

Where X_n^i is random variable and δ (*x*) is Dirac delta function, the sum of the weights W_n^i is 1. And the weights are defined by importance density:

$$W_n^i \propto \frac{p(X_{0:n}^i | Y_{1:n})}{q(X_{0:n}^i | Y_{1:n})} \quad (4.12)$$

Many researches had weight update process.

$$W_n^i = W_{n-1}^i \frac{p(Y_n \left| X_n^i \right.) p(X_n^i \left| X_{n-1}^i \right.)}{q(X_n^i \left| X_{0:n-1}, Y_{1:n} \right.)} \quad (4.13)$$

The transition prior probability is used as the proposal distribution:

$$q(X_n^i \left| X_{0:n-1}, Y_{1:n} \right.) = p(X_n^i \left| X_{n-1}^i \right.) \quad (4.14)$$

So the weight updating can be simplified into:

$$W_n^i = W_{n-1}^i p(Y_n \left| X_n^i \right.) \quad (4.15)$$

A common problem with the sequential importance sampling filter is the sample impoverishment, where after several iterations only very few particles have non-zero importance weights. This also means that large computational effort is dedicated to updating particles whose contributions to the approximation of the posterior PDF is basically negligible (Salamon and Feyen, 2009). So in many application cases, the number of particles is very large but unfortunately this may cause another big problem of high demand of computer with the increase of particles.

And then the updated weights are applied into Kalman Filter to narrow model uncertainty. In normal Kalman Filter process, the error system is assumed to agree with Gaussian distribution but in many hydrological applications, the assumption is not in this case. So the particle filter may improve the filter performance in data assimilation.

4.3 Application of assimilation in SRM model in the Langtang Khola catchment

The performance of the SRM model is of large uncertainties. The initial condition, model inputs and model itself together result in the model uncertainties. In the processes of precipitation, temperature and evaporation observations, there are measurement errors. The SRM hydrological process is distributed so the environmental heterogeneity of the catchment brings some uncertainties to the model outputs. And the SRM model itself is a simplized hydrological process, which may cause the model error. To obtain more believable model performance, several techniques can be applied to narrow the model uncertainty.

Data assimilation is applied in the Langtang Khola catchment to narrow the SRM model uncertainty. Monte Carlo simulation is the base for other two assimilation methods and is regarded as a simple data assimilation technique. The assimilation processes of SRM model are shown in figure 4.4. The first step is to perturb model parameters to transfer deterministic model process into stochastic one. Then the assimilation techniques can be applied to obtain the realizations for Monte Carlo, ensembles for EnKF and particles for Particle Flitering. The analysis of model results and performance ability is based on the realizations, ensembles and particles.

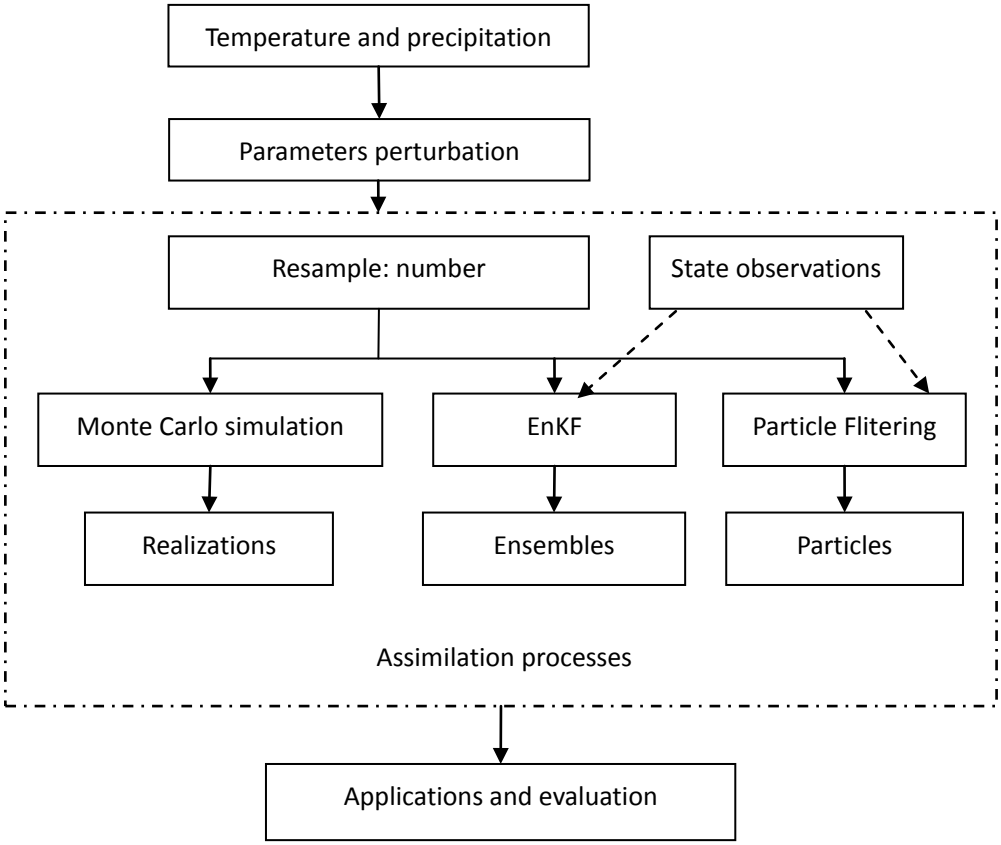


Figure 4. 4 Schematic view of assimilation processes in the SRM hydrological model

4.3.1 Parameters perturbation

Data assimilation uses a stochastic version of hydrological model. The parameter perturbation must be done before assimilation. The input precipitation and temperature are referred to as P_{in} and T_{in}. The true inputs are with a small perturbation:

$$P = P_{in} + \delta P$$
$$T = T_{in} + \delta T \qquad (4.16)$$

The perturbation for δP is realization from a Gaussian distribution with zero mean and a variance of 25mm^2. This value can make the precipitation perturb at a large scale. In most of drought time and rain season, the error of 5 mm did not seem extraordinary. If the variance is below 25mm^2, the model results spread too wide, which is out of the realization.

The perturbation for δT is realization from a Gaussian distribution with zero mean and a variance of 4^{o}C^2. This error is very important for the occurrence of snow fall and snow melt, and directly influencing the snow melt runoff.

The idea for the parameters perturbation is to transfer the determinant model into stochastic one. Then data assimilation techniques can be applied to solve the stochastic hydrological process. In the following part, Monte Carlo simulation, ensemble Kalman Filter and particle filtering are applied in the Langtang Khola catchment to assimilate the SRM model process and state from integrated SRM with remote sensing data.

4.3.2 Assimilation experiments

The data assimilation techniques in this study are implemented using a combination of the PCRaster and the Python scripting language because there is assimilation software framework in PCRaster system (Karssenberg et al., 2010). Table 4.1 reflects the change resample parameter, begin time and end time in assimilation process. To test the sensitivity of sample in assimilation, we can change the nrSamples in Python scripts. In ensemble Kalman filter and particle filter, eight-day discharges from remote sensing snow cover product are regarded as discharge observations at outlet in the Langtang Khola catchment because the real-time snow cover data is integrated in SRM model to update snow storage in modeling process so the results can be improved.

Table 4.1 Sample parameters in Python scripts for data assimilation techniques (classes from Karssenberg et al., 2010)

Data assimilation techniques	Part of Python scripts
Monte Carlo simulation	myModel = SnowModel() dynamicModel = DynamicFramework(myModel, lastTimeStep=2191, firstTimestep=1) mcModel = MonteCarloFramework(dynamicModel, nrSamples=32) mcModel.run()
Ensemble Kalman filter	myModel = SnowModel() dynamicModel = DynamicFramework(myModel, lastTimeStep=2191, firstTimestep=1) mcModel = MonteCarloFramework(dynamicModel, nrSamples=32) kfModel = KalmanFilterFramework(mcModel) filterMoments = [every eight day] kfModel.setFilterTimesteps(filterMoments) kfModel.run()
Particle filter	myModel = SnowModel() dynamicModel = DynamicFramework(myModel, lastTimeStep=2191, firstTimestep=1) mcModel = MonteCarloFramework(dynamicModel, nrSamples=32) pfModel = SequentialImportanceResamplingFramework(mcModel) pfModel.setFilterTimesteps([moments]) pfModel.run()

Figure 4.5 presents the results for the data assimilation using Monte Carlo simulation, ensemble Kalman filter and particle filter (Sequential Importance Resampling). A general improvement in discharge can be shown in comparison to the simulation results without data assimilation techniques. There is large difference between discharge of observation and Monte Carlo simulation because, in Monte Carlo simulation, observations are ignored and this technique is not an assimilation technique but only a stochastic way for determinant hydrological model. Discharges for EnKF and particle filter are close to the observations because, during assimilation process, observations are applied to update model states and then the performance for the assimilation result is improved.

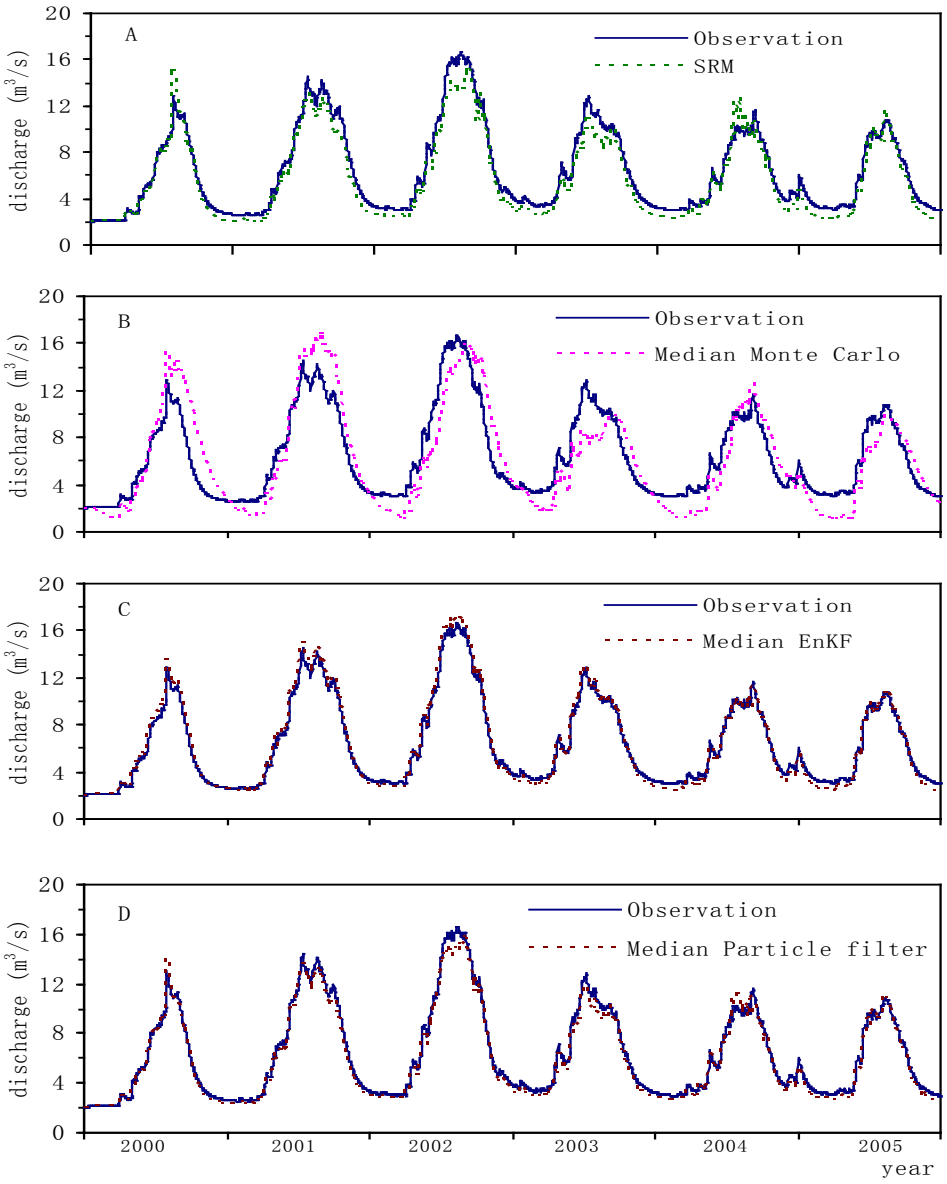


Figure 4. 5 Discharge at outlet in the Langtang Khola catchment for the data assimilation.

To obtain the detailed information of the data assimilation performance, we calculate the deviation between discharge for assimilation techniques and observation. Figure 4.6 presents the discharge deviation between assimilation and observations. The discharge deviation is large in summer time but very small in winter just as that in summer time the discharge is large but small in winter seasons. The errors between median discharge for Monte Carlo simulation and observations are the largest in these techniques. But for ensemble Kalman filter and particle filter, the performances are perfect with small errors between observations and assimilation results. EnKF performance is better than that for particle filter mainly because particle filter relies more on the distribution of resample particles and number of copies. So, in large discharge period, there is a relatively high error for particle filter. EnKF performance has stable small errors in all periods. The performance for EnKF is best in all the data assimilation techniques.

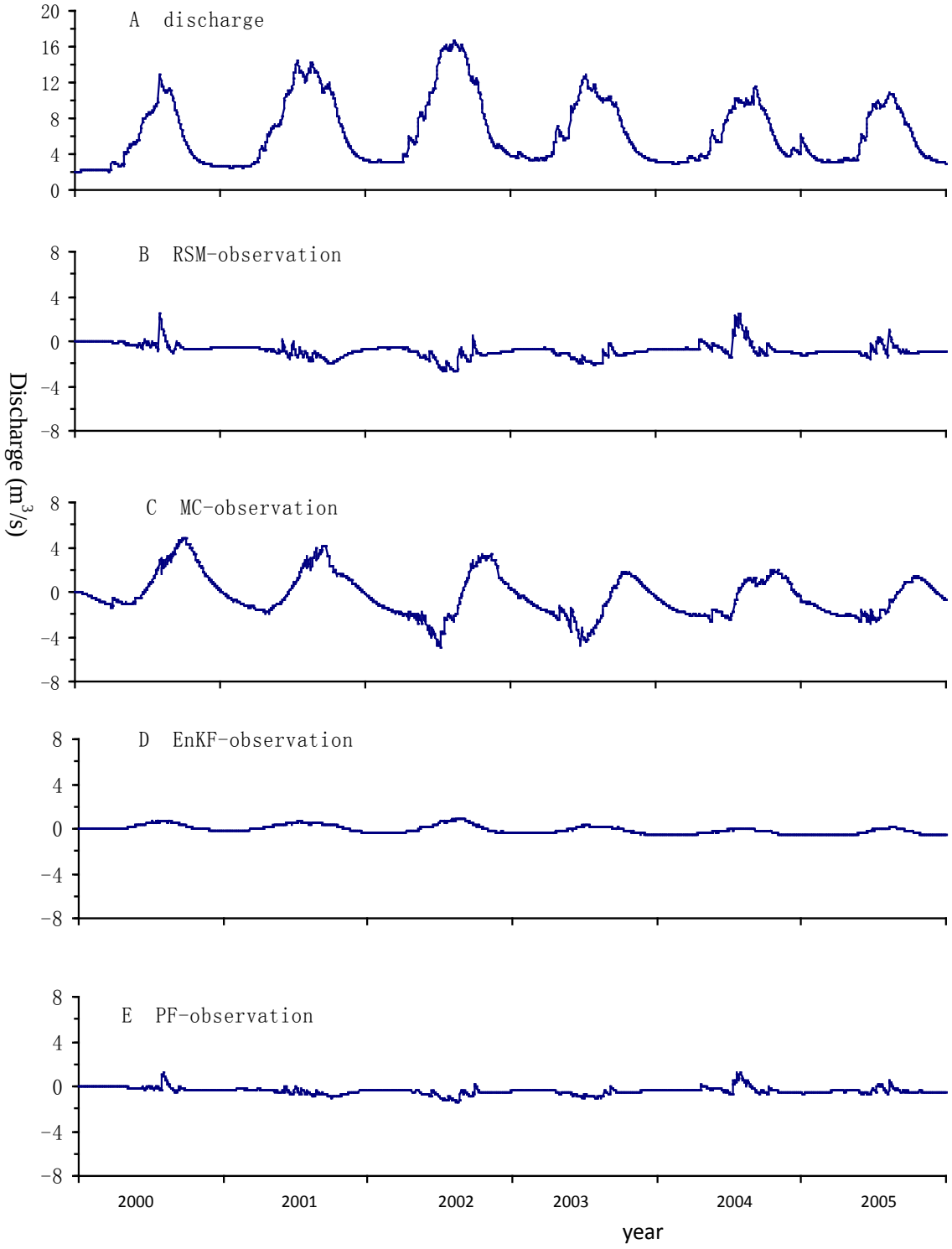


Figure 4. 6 The performance for data assimilation techniques in the Langtang Khola catchment

To evaluate the performance of data assimilation techniques, Root-Square-Mean Error (RSME) is applied in this study. When the number of particles in assimilation is 32, the SRME between observation and median assimilated discharge is shown in table 4.2. The table presents performance for EnKF technique is best in all the assimilation methods.

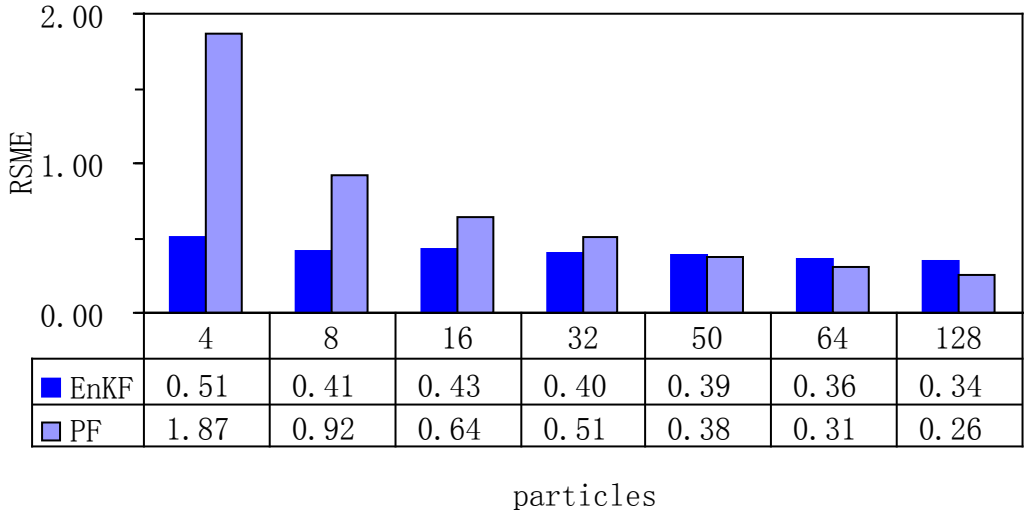
Table 4.2 Root-Square-Mean Error of the observation and assimilated discharge for all assimilation

techniques

Data assimilation	Monte Carlo simulation	EnKF	Particle filter
RSME	1.89	0.37	0.51

Due to the sensitivity of particle filter to the number of resample particles, different numbers of particles are experimented in this research to analyze the effects of particles on the performance for particle filter. The table 4.3 presents that in particle filtering with the increase of particles the performance for the filtering is better and better. But the computational demand is also increasing but more rapidly than that of the model accuracy. Generally, to avoid overlarge computational demand, moderate number of particles is selected in particle filtering. The performance for ensemble Kalman filter is not markedly related with the increase of the number of ensembles almost remains at the same level.

Table 4.3 Sensitivity of the particle number in assimilation to the assimilated performance for EnKF and PF



4.4 Discussion and conclusion

Data assimilation is a crucial method to narrow model uncertainty from different data source. In the study, Monte Carlo simulation, ensemble Kalman filter and particle have been applied into Snowmelt Runoff Model (SRM) and the assimilated results have been analyzed and compared. With the precipitation and temperature perturbation, the RSM model has been transferred into stochastic model. For a low number of ensemble/ particle members, EnKF performed better than particle filter. For an intermediate number of ensemble/ particle member, there is little difference of the performances between the two assimilation techniques. But for large number of the ensemble/ particle members, particle filter outperformed ensemble Kalman filter. While increase in the number of ensembles did not improve the performance in ensemble Kalman filter.

The ensemble Kalman filter and the particle filter have been compared in this research when the two have been applied in SRM model. Data assimilation provides a method for the issues of uncertainty in model and observation. As two main data assimilation methods, ensemble Kalman and particle both performed well in error management of model and observation.

Reference

Andreadis and Lettenmaier 2006, K. M. Andreadis and D. P. Lettenmaier, Assimilating remotely sensed snow observations into a macroscale hydrology model, Advances in Water Resources, 2006, 29, 872–886

Barnett et al., 2005, T.P. Barnett, J.C. Adam and D.P. Lettenmaier, Potential impacts of a warming climate on water availability in snow-dominated regions, Nature, 2005, 438, 303–309

Beck, 1987, M. B. Beck, Water Quality Modeling: A Review of the Analysis of Uncertainty, Water Resour. Res., 1987, 23, 1393–1442.

Bierkens, 2009, M. F. P. Bierkens, 2009, Stochastic hydrology, Utrecht University course, 2009

Chen, et al., 2005. J. M. Chen, X. Chen, W. Ju et al., Distributed hydrological model for mapping evapotranspiration using remote sensing inputs, Journal of Hydrology, 2005, 1-4, 15-39.

Clark et al, 2008, M. P. Clark, D. E. Rupp, R. A. Woods, et al. Hydrological data assimilation with the ensemble Kalman filter: Use of streamflow observations to update states in a distributed hydrological model. Advances in Water Resources, 2008, 31, 1309–1324

Cohen and Entekhabi, 1999, J. Cohen and D. Entekhabi, Eurasian snow cover variability and Northern Hemisphere climate predictability, Geophysical Research Letters, 1999, 26, 345–348.

Croft, 1944, A. R. Croft, Snow Melting and Evaporation, Science, 1944, 100, 169-170.

Douville and Peings, 2010, H. Douville and Y. Peings. Influence of the Eurasian snow cover on the Indian summer monsoon variability in observed climatologies and CMIP3 simulations. Climate dynamics, 2010, 34,643-660

Dunlap et al., 2010, E. Dunlap, Y. Wu and C. Tang, Assimilation of sea surface temperature into CECOM by flux correction, Ocean dynamics, 2010, 60, 403-412

Fukushima et al., 1987, Y. Fukushima, K. Kawashima, M. Suzuki, et al., The hydrological data of the Langtang Valley, Nepal Himalayas, Bulletin of Glacier Research, 1987, 5, 115-120.

Giraldo, 2009 et al.. M. A. Giraldo, D. Bosch, M. Madden, Ground and surface temperature variability for remote sensing of soil moisture in a heterogeneous landscape, Journal of Hydrology, 2009, 1-4, 214-223

Georgievsky 2009, M. Georgievsky, Application of the Snowmelt Runoff model in the Kuban river basin using MODIS satellite images. Environmental research letters, 2009, 4, 045017.

Gupta et al., 2007, R.P. Gupta, A. Ghosh and U.K Haritashya. Empirical relationship between near-IR reflectance of melting seasonal snow and environmental temperature in a Himalayan basin, Remote Sensing of Environment, 2007, 107, 402-413

Hall et al., 1995, D. K. Hall, G. A. Riggs, V. V. Salomonson. Development of methods for mapping global snow cover using Moderate Resolution Imaging Spectroradiometer (MODIS) data. Remote Sensing of Environment, 1995, 54, 127-140

Hall et al., 2002, D.K. Hall, G.A. Riggs and V.V. Salomonson et al., MODIS snow-cover products, Remote Sensing of Environment , 2002, 83, 181–194

Immerzeel et al., 2009, W.W. Immerzeel, P. Droogers, S.M. de Jong, et al, Large-scale

monitoring of snow cover and runoff simulation in Himalayan river basins using remote sensing, Remote Sensing of Environment, 2009, 113, 40–49.
Karssenberg et al., 2010, D. Karssenberg, O. Schmitz, P. Salamon, et al. A software framework for construction of process-based stochastic spatial-temporal models and data assimilation, Environmental Modelling and Software, 2010, 25, 489-502.
Kitanidis and Bras 1980, P. K. Kitanidis, and R. L. Bras, Real-time forecasting with a conceptual rainfall runoff model: 1. Analysis of uncertainty, Water Resour. Res., 1980, 6, 1025–1033.
Kite, 1991, G.W. Kite, Watershed model using satellite data applied to a mountain basin in Canada, Journal of Hydrology, 1991, 128, 157–169
Kuczera and Mroczkowski 1998, G. Kuczera and M. Mroczkowski, Assessment of hydrological parameter uncertainty and the worth of multiresponse data, Water Resour. Res., 1998, 6, 1481– 1489.
Kuczera and Parent 1998, G. Kuczera, and E. Parent, Monte Carlo assessment of parameter uncertainty in conceptual catchment models: The Metropolis algorithm, J. Hydrol., 1998, 211, 69– 85.
Kumar et al., 2008, S.V. Kumar, R.H. Reichle, C.D Peters-Lidard et al., A land surface data assimilation framework using the land information system: Description and applications, Advances in water resources, 2008, 31, 1419-1432
Liang et al., 2008 a, T.G. Liang, X.D. Huang, C.X. Wu, et al, An application of MODIS data to snow cover monitoring in a pastoral area: A case study in Northern Xinjiang, China, Remote sensing of environment, 2008, 112, 1514-1526.
Liang et al., 2008 b, T. Liang, X. Zhang, H. Xie et al, Toward improved daily snow cover mapping with advanced combination of MODIS and AMSR-E measurements, Remote sensing of environment, 2008, 112, 3750-3761
Lopez et al., 2008, P. Lopez, P. Sirguey, and Y. Arnaud, et al. Snow cover monitoring in the Northern Patagonia Icefield using MODIS satellite images (2000-2006). Global and planetary change, 2008, 61, 103-116
Lyapustin et al., 2009, A. Lyapustin, M. Tedesco, Y. Wang, et al, Retrieval of snow grain size over Greenland from MODIS, Remote sensing of environment, 2009, 113, 1976-1987.
Maqueda and Fichefet, 1999, M. Maqueda and T. Fichefet. Modelling the influence of snow accumulation and snow-ice formation on the seasonal cycle of the Antarctic sea-ice cover. Climate dynamics, 1999, 4, 251-268
Martinec, 1975, J. Martinec. Snowmelt runoff model for stream flow forecasts. Nordic Hydrology, 1975, 6, 145-154
Martinec et al., 1994, J. Martinec, A. Rango and R. Roberts In: M. Baumgartner, Editor, Snowmelt runoff model (SRM) user's manual (version 3.0), 1994, Department of Geography, University of Bern, Switzerland .
McLaughlin, 2002, M. McLaughlin, An integrated approach to hydrologic data assimilation: interpolation, smoothing, and filtering, Adv Water Resour., 2002, 25, 1275–1286
McLaughlin et al., 2005, D. McLaughlin, A. O'Neill, J. Derber et al., Opportunities for enhanced collaboration within the data assimilation community, Quart J Roy Meteorol Soc., 2005, 131, 3683–3693.
McLaughlin, 1995, D. McLaughlin, Recent advances in hydrologic data assimilation, US Natl

Rep Int Union Geod Geophys 1991–1994. Rev Geophys, 1995, 33, 977–984.
Minville etal., 2008, M. Minville, F. Brissette and R. Leconte, Uncertainty of the impact of climate change on the hydrology of a nordic watershed, Journal of hydrology, 2008, 1-2: 70-83
Mitchell and Jones 2005, T. D. Mitchell and P. D. Jones. An improved method of constructing a database of monthly climate observations and associated high-resolution grids. International Journal of Climatology, 2005, 25, 693-612.
Moradkhani et al., 2005, H. Moradkhani, S. Sorooshian, H.V. Gupta and P.R. Houser, Dual state-parameter estimation of hydrological models using ensemble Kalman filter, Adv. Water Resour. 2005, 28, 135–147.
Oerlemans and Fortuin, 1992, J. Oerlemans, J. P. F. Fortuin. Sensitivity of glaciers and small ice caps to greenhouse warming. Science, 1992, 258: 115-118.
Pierdicca , et al., 2010. N. Pierdicca, L. Pulvirenti, C. Bignami. Soil moisture estimation over vegetated terrains using multitemporal remote sensing data, Remote Sensing of Environment, 2010, 2, 440-448
Qin et al., 2008, C. Qin, Y. Jia, Z. Su, et al, Integrating Remote Sensing Information Into A Distributed Hydrological Model for Improving Water Budget Predictions in Large-scale Basins through Data Assimilation, Sensors, 2008, 8, 4441
Ramage and Isacks, 2003, J.M. Ramage and B.L. Isacks, Interannual variations of snowmelt and refreeze timing in southeast Alaskan icefields, USA, Journal of Glaciology, 2003, 49, 102–116.
Rango, 1993, A. Rango, Snow hydrology processes and remote sensing, Hydrological Processes, 1993, 7, 121–138.
Rango, 1997, A. Rango, The response of areal snow cover to climate change in a snowmelt-runoff model, Annals of Glaciology, 1997, 25, 232–236
Rebetez, 1996, M. Rebetez. Seasonal relationship between temperature, precipitation and snow cover in a mountainous region, Theoretical and Applied Climatology, 1996, 54, 99-106
Salamon and Feyen, 2009, P. Salamon and L. Feyen. Assessing parameter, precipitation, and predictive uncertainty in a distributed hydrological model using sequential data assimilation with the particle filter. Journal of Hydrology, 2009, 3, 428-442
Salomonson and Appel, 2004, V.V. Salomonson and I. Appel, Estimating fractional snow cover from MODIS using the normalized difference snow index. Remote sensing of environment, 2004, 89, 351-360.
Schaffhauser et al., 2008, A. Schaffhauser, M. Adams, R. Fromm, et al, Remote sensing based retrieval of snow cover properties, Cold regions science and technology, 2008, 54, 164-175.
Schaper et al., 1999, J. Schaper, J. Martinec and K. Seidel, Distributed mapping of snow and glaciers for improved runoff modeling, Hydrologic Processes, 1999, 13, 2023–2031
Sirguey, et al. 2009, P. Sirguey, R. Mathieu and Y. Arnaud, Subpixel monitoring of the seasonal snow cover with MODIS at 250 m spatial resolution in the Southern Alps of New Zealand: Methodology and accuracy assessment, Remote sensing of environment, 2009, 1, 160-181.
Stanzel et al., 2008, Ph Stanzel, U. Haberl and H. P. Nachtnebel, Modelling snow accumulation and snow melt in a continuous hydrological model for real-time flood forecasting, IOP conference series. Earth and environmental science, 2008, 4, 012-016
Tekeli et al., 2005, A.E. Tekeli, Z. Akyurek, A. Arda Sorman, et al, Using MODIS snow cover maps in modeling snowmelt runoff process in the eastern part of Turkey. Remote sensing of

environment, 2005, 97, 216-230.
Troch et al, 2003, P. A. Troch, C. Paniconi and D. McLaughlin. Catchment-scale hydrological modeling and data assimilation. Advances in Water Resources, 2003, 26, 131–135
Tsuyuki, 1996, T. Tsuyuki, Variational data assimilation in the tropics using precipitation data part I: Column model, Meteorology and atmospheric physics, 1996, 60, 87-104
Vavrus, 2007, S. Vavrus, The role of terrestrial snow cover in the climate system, Climate Dynamics, 2007, 29, 73-88
Walker and Houser 2001, J. P. Walker and P. R. Houser. A methodology for initializing soil moisture in a global climate model: assimilation of near-surface soil moisture observations. J Geophys Res, 2001, 106, 61–74.
Wang and Xie, 2009, X. Wang and H. Xie. New methods for studying the spatiotemporal variation of snow cover based on combination products of MODIS Terra and Aqua, Journal of Hydrology, 2009, 371, 192-200.
Weertman, 1957, J. Weertman, On the sliding of glaciers, Journal of Glaciology, 1957, 21, 33-38.
Weertman, 1964, J. Weertman, The theory of glacier sliding. Journal of Glaciology, 1964, 5, 287-303
Weerts and El Serafy, 2006, A. H., Weerts and G. Y. H. El Serafy, Particle filtering and ensemble Kalman filtering for state updating with hydrological conceptual rainfall-runoff models, Water Resour. Res., 2006, 42, W09403.
Xiao et al., 2004, X. Xiao, Q. Zhang and S. Boles et al., Mapping snow cover in the Pan-Arctic zone using multi-year (1998–2001) images from optical VEGETATION sensor, International Journal of Remote Sensing, 2004, 25, 5731–5744.

Summary

Mountainous snow is essential substance in the earth surface because it is important for inland fresh water provision, global and local climate regulation and water cycles. The research is to monitor mountainous snow cover, simulate snow melt runoff using snowmelt runoff model (SRM) and assimilate state from remote sensing data into snow melt runoff model in the Langtang catchment.

The first topic is to monitor snow cover in the Langtang catchment using MODIS snow product. It gives an analysis of snow cover area change with time. Inter- and between- annual analysis methods are two different perspectives for snow cover time monitoring. Some interesting trends would be drawn in this section. The results showed a seasonal variation in snow cover with two peaks and one minimum in snow cover percentage trends. The maximum snow cover area is about 1800 km^2 (94% in the catchment) in winter time, while in summer time, the bottom is only about 200 km^2 (10% in the catchment). The reason of this seasonal variation of snow is explained by temporal variation in local temperature and precipitation. In summer time, the temperature is above 0 celsius and snow metling occurs. During winter time with quite low temperature, the precipitation is mainly in the form of snow so the snow cover area expands graduately. But in many winter time, there is a drought period without preciptation, which causes the snow cover to reduce. This leads to a second snow cover area percentage peak. After the dry period, snow recession may end, snow cover increases again,which leads to peak area.

The second topic focused on the snowmetl runoff modeling. Snowmelt Runoff Model (SRM) has been applied into the Langtang Khola catchment to model snowmelt runoff and discharge. It simulates snow melt, snowmelt runoff and discharge. Remote sensing data provide important snow cover information for the snow storage. The remote sensing snow cover data have been integrated into SRM model to update ice storage maps for every eight days. The ice storage updates in SRM model are from the estimation from ice flux budget. Integration remote sensing data with RSM model can improve model performance.

The final component is to quantify model uncertainties and narrow model errors using data assimilation techniques. Ice storage is an important state for SRM model. Icestorage maps are simulated from remote sensing snow cover data. In data assimilation processes, ice storage maps from remote sensing are used to update model state. Monte Carlo simulation, ensemble Kalman filter and particle have been applied into the Snowmelt Runoff Model (SRM) to quantify model errors. For a low number of ensemble/ particle members, EnKF performed better than particle filter. For an intermediate number of ensemble/ particle member, there is little difference of the performances between the two assimilation techniques. But for large number of the ensemble/ particle members, particle filter outperformed ensemble Kalman filter. While increase in the number of ensembles did not improve the performance in ensemble Kalman filter.

The study has covered a wide range for snow research. Snow cover is the basis for snowmelt runoff modeling and snowmelt runoff model is the basis for assimilation. The remote sensing snow cover data are integrated and assimilated into snowmelt runoff model.

Appendix

Code script for integration process:

```
# RSMinte.py
from PCRaster import *
from PCRaster.Framework import *

class dataModel(DynamicModel):
    def __init__(self):
        DynamicModel.__init__(self)
        setclone("clone90.map")

    def initial(self):
        self.snow=scalar(0)
        self.mask=self.readmap("mask90")
        # dem correction
        self.tempLapse=scalar(-5.341300E-03)
        self.dem=self.readmap("demfil90")
        stationH=scalar(3920)
        self.tempCor=ifthenelse(self.mask,(self.dem-stationH)*self.tempLapse,0)
        self.Ldd=self.readmap("ldd90")
        # day degree factor
        self.DDF=scalar(4)
        # outlet
        self.Outlet=self.readmap("outlet")

        IniSoil=0.050;                  # Initial soil moisture storage (m)
        self.MaxSoil=0.100;             # Maximum soil moisture storage (m)
        self.SoilS=IniSoil;               # Initial soil moisture storage (m)
        self.kx=.9490430;               # x flow recession coefficient
        self.QtotAold=2;                # initial discharge (m3/s)
        gf=.5000000;                      # fraction of total ablation that results in
actual runoff
        self.SnowF=scalar(0)
        self.AblF=scalar(0)
        self.ETc=0.0918
        self.rc=0.001
        self.kc=0.8
        self.Qini=0.0005
        self.Qb=self.Qini

        self.MaxIceIni=200
```

```
        self.SinSlope=sin(atan(slope(self.dem)))
        self.gmask=self.readmap("gmask90")
        self.SnowC=self.gmask

Ice=cover(ifthenelse(self.gmask,ifthenelse(self.SinSlope==0,200,13.343968/self.SinSlope),0))
        self.IceIni=min(200,ifthenelse(self.mask,Ice,0))
        self.IceStore=self.IceIni
        self.CN=75.000
        self.S=0.001*25.4*((1000/self.CN)-10)
        self.nr=8
        self.mr=0

        #self.IceStore=self.IceIni

        # initialise timeoutput
        self.runoffTSS=TimeoutputTimeseries("runoff2.txt",self,"outlet.map",)
#noHeader=False

    def dynamic(self):

        self.mr=self.mr+1

        if self.mr==self.nr:
            self.SnowC=self.readmap("D:\biyelunwen\rs_snow_map\ms")
            self.IceStore=ifthenelse(boolean(self.SnowC),max(0.005,self.IceStore),0)

        self.report(self.IceStore,"D:\biyelunwen\states\IceS")      #output icestorage state

        # prec
        Prec=timeinputscalar("clim_00_07.tss",1)
        Temp=timeinputscalar("clim_00_07.tss",2)
        #Prec=self.readmap("Prec")
        # temp
        #Temp=self.readmap("Temp")



        Snow=ifthenelse(self.mask,cover(ifthenelse(Temp<0, Prec/1000,0)),0)
        # ablation
        Abl=cover(min(self.IceStore,ifthenelse(Temp>0, Temp*self.DDF/1000,0)))
        rain=Prec/1000-Snow


        self.IceStore=self.IceStore+Snow-Abl
```

```
        self.SnowC=ifthenelse(self.IceStore>0,scalar(1),scalar(0))

        # hydrological process
        ET0=timeinputscalar("clim_00_07.tss",3)
        ET0=(ET0+self.tempCor*self.ETc)/1000


        potentialS=self.SoilS+rain-self.kc*ET0
        potentialSG=self.SoilS
        self.SoilS=ifthenelse(self.IceStore>0.005,potentialSG,min(self.MaxSoil,potentialS))


SurQ=ifthenelse(self.IceStore<0.05,ifthenelse(rain>0.2*self.S,(sqr(rain-0.2*self.S))/(rain+0.8*
self.S),0),0)

        Abl=0.50000*Abl
        Recharge=ifthenelse(self.SoilS==self.MaxSoil,rain-self.kc*ET0+Abl*0.5-SurQ,0)
        self.Qb=self.Qb*exp(-1*self.rc)+Recharge*(1-exp(-1*self.rc))

        Qtot=Abl+SurQ+self.Qb

        Qtot=(Qtot*90*90)/(24*3600)
        QtotA=accuflux(self.Ldd,Qtot)
        QtotA=(1-self.kx)*QtotA+self.kx*self.QtotAold
        self.QtotAold=QtotA
        discharge=self.QtotAold


        self.report(discharge,"srunoff.txt")
        self.runoffTSS.sample(discharge)




myModel=dataModel()
dynModel=DynamicFramework(myModel, lastTimeStep=2191, firstTimestep=1)

dynModel.run()
```

Code script for assimilation process:
Monte Carlo simulation

```
from PCRaster import *
from PCRaster.Framework import *

class SRModel(DynamicModel, MonteCarloModel):
    def __init__(self):
        DynamicModel.__init__(self)
        MonteCarloModel.__init__(self)
        setclone("clone.map")

    def premcloop(self):

        dem = self.readmap("demfil90")
        self.ldd = lddcreate(self.dem, 1e31, 1e31, 1e31, 1e31)
        elevationMeteoStation = scalar(3920)
        self.elevationAboveMeteoStation = self.dem - elevationMeteoStation
        self.DDF = 4
        self.SinSlope=sin(atan(slope(self.dem)))
        self.ETc=0.0918
        self.kc=0.8
        self.rc=0.001
        self.kx=0.9490403
        self.gf=0.50000
        self.mask=self.readmap("mask90")
        self.gmask=self.readmap("gmask90")
        self.Outlet=self.readmap("outlet")

    def initial(self):
        IniSoil=0.050
        self.MaxSoil=0.100
        self.SoilS=IniSoil
        self.Qini=0.0005
        self.Qb=self.Qini
        self.QtotAold=2
        self.Qtot=0
        self.rho=916.7
        self.tau0=120000
        self.g=9.81
        self.R=5.059000E+08
        self.v=0.1
        self.CN=75.000
        self.S=0.001*25.4*((1000/self.CN)-10)
```

```
Ice=cover(ifthenelse(self.gmask,ifthenelse(self.SinSlope==0,200,13.343968/self.SinSlope),0))
        self.IceIni=min(200,ifthenelse(self.mask,Ice,0))
        self.IceStore=self.IceIni
        self.nr=8
        self.SnowC=scalar(self.gmask)

    def dynamic(self):
        temperatureObserved = self.readDeterministic("Temp")
        precipitationObserved = self.readDeterministic("Prec")
        precipitation = max(0, precipitationObserved * (mapnormal() * 5 + 1.0))
        temperature = self.temperatureObserved*(mapnormal()*0.2+1)
        Snow=ifthenelse(self.mask,cover(ifthenelse(temperature<0,precipitation/1000,0)),0)
        Abl=ifthenelse(self.mask,cover(ifthenelse(temperature>0,
temperature*self.DDF/1000,0)),0)
        Abl=min(self.IceStore,Abl)

        tau=self.rho*self.g*self.IceStore*self.SinSlope
        us=ifthenelse(tau>self.tau0,sqr((tau-self.tau0)/(sqr(self.v)*self.R)),0)
        Flux=min(self.IceStore,us*(self.IceStore/celllength())*24*3600)
        ASPECT=aspect(self.dem)
        SINASPECT=sin(ASPECT)
        COSASPECT= cos(ASPECT)
        FRACTO_X = SINASPECT / (abs(SINASPECT)+abs(COSASPECT))
        FRACTO_Y = COSASPECT / (abs(SINASPECT)+abs(COSASPECT))
        DZ_BLOCK_XLEFT=    max(0,shift0(-FRACTO_X*Flux,0, 1))
        DZ_BLOCK_XRIGHT= max(0,shift0( FRACTO_X*Flux,0,-1))
        DZ_BLOCK_YUP=       max(0,shift0( FRACTO_Y*Flux, 1,0))
        DZ_BLOCK_YDOWN=    max(0,shift0(-FRACTO_Y*Flux,-1,0))
        FluxDS                                                                                  =
ifthenelse(self.mask,DZ_BLOCK_XLEFT+DZ_BLOCK_XRIGHT+DZ_BLOCK_YDOWN+DZ_BLOCK_
YUP,0)
        self.IceStore=ifthenelse(self.mask,max(0,self.IceStore+Snow+FluxDS-Flux-Abl),0)
        self.IceStore=min(200,self.IceStore)
        rain=(precipitation/1000)-Snow
        ET0=ET0=timeinputscalar("clim_00_07.tss",3)
        ET0=(ET0+self.temperatureCorrection*self.ETc)/1000

SurQ=ifthenelse(self.IceStore<0.05,ifthenelse(rain>0.2*self.S,(sqr(rain-0.2*self.S))/(rain+0.8*
self.S),0),0)

        pontentialS=max(0,rain+self.SoilS-self.kc*ET0-SurQ)
        potentialSG=max(0,self.SoilS)
        self.SoilS=ifthenelse(self.IceStore>0.05,potentialSG,min(self.MaxSoil,pontentialS))
```

```
        Abl=self.gf*Abl
        Recharge=ifthenelse(self.SoilS==self.MaxSoil,rain-self.kc*ET0+Abl*(1-self.gf)-SurQ,0)
        Recharge=max(0,Recharge)
        self.Qb=self.Qb*exp(-1*self.rc)+Recharge*(1-exp(-1*self.rc))
        self.Qtot=Abl+self.Qb+SurQ
        self.Qtot=(self.Qtot*90*90)/(24*3600)
        #self.report(self.Qtot,"Wtot")
        QtotA=accuflux(self.ldd,self.Qtot)
        QtotA=(1-self.kx)*QtotA+self.kx*self.QtotAold
        QtotA=max(0,QtotA)
        self.QtotAold=QtotA
        discharge=self.QtotAold
        self.report(discharge,"q")
        #self.runoffTSS.sample(discharge)

def postmcloop(self):
        names = ["q"]
        mcaveragevariance(names, self.sampleNumbers(), self.timeSteps())
        percentiles = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
        mcpercentiles(names, percentiles, self.sampleNumbers(), self.timeSteps())

myModel = SRModel()
dynamicModel = DynamicFramework(myModel, lastTimeStep=2191, firstTimestep=1)
mcModel = MonteCarloFramework(dynamicModel, nrSamples=128)
mcModel.run()
```

Particle Filter:

```
from PCRaster import *
from PCRaster.Framework import *

class SRModel(DynamicModel, MonteCarloModel, ParticleFilterModel):
    def __init__(self):
        DynamicModel.__init__(self)
        MonteCarloModel.__init__(self)
        ParticleFilterModel.__init__(self)
        setclone("clone90.map")

    def premcloop(self):
        dem = self.readmap("demfil90")
        self.ldd = lddcreate(dem, 1e31, 1e31, 1e31, 1e31)
```

```
        elevationMeteoStation = scalar(3920)
        self.elevationAboveMeteoStation = dem - elevationMeteoStation
        self.DDF = 4
        self.SinSlope=sin(atan(slope(self.dem)))
        self.ETc=0.0918
        self.kc=0.8
        self.rc=0.001
        self.kx=0.9490403
        self.gf=0.50000
        self.mask=self.readmap("mask90")
        self.gmask=self.readmap("gmask90")
        self.Outlet=self.readmap("outlet")

    def initial(self):

        IniSoil=0.050
        self.MaxSoil=0.100
        self.SoilS=IniSoil
        self.Qini=0.0005
        self.Qb=self.Qini
        self.QtotAold=2
        self.Qtot=0
        self.rho=916.7
        self.tau0=120000
        self.g=9.81
        self.R=5.059000E+08
        self.v=0.1
        self.CN=75.000
        self.S=0.001*25.4*((1000/self.CN)-10)

Ice=cover(ifthenelse(self.gmask,ifthenelse(self.SinSlope==0,200,13.343968/self.SinSlope),0))
        self.IceIni=min(200,ifthenelse(self.mask,Ice,0))
        self.IceStore=self.IceIni
        self.nr=8
        self.SnowC=scalar(self.gmask)

        self.temperatureLapseRate = 0.005 + (mapnormal() * 0.001)
        self.temperatureCorrection = self.elevationAboveMeteoStation\
                * self.temperatureLapseRate

    def dynamic(self):

        self.temperatureObserved = self.readDeterministic("Temp")
        self.precipitationObserved = self.readDeterministic("Prec")
```

```
precipitation = max(0, self.precipitationObserved * (mapnormal() * 0.2 + 1.0))
temperature = self.temperatureObserved*(mapnormal()*0.2+1)
Snow=ifthenelse(self.mask,cover(ifthenelse(temperature<0,precipitation/1000,0)),0)
Abl=ifthenelse(self.mask,cover(ifthenelse(temperature>0,
temperature*self.DDF/1000,0)),0)
Abl=min(self.IceStore,Abl)
tau=self.rho*self.g*self.IceStore*self.SinSlope
us=ifthenelse(tau>self.tau0,sqr((tau-self.tau0)/(sqr(self.v)*self.R)),0)
Flux=min(self.IceStore,us*(self.IceStore/celllength())*24*3600)
ASPECT=aspect(self.dem)
SINASPECT=sin(ASPECT)
COSASPECT= cos(ASPECT)
FRACTO_X = SINASPECT / (abs(SINASPECT)+abs(COSASPECT))
FRACTO_Y = COSASPECT / (abs(SINASPECT)+abs(COSASPECT))
DZ_BLOCK_XLEFT=    max(0,shift0(-FRACTO_X*Flux,0, 1))
DZ_BLOCK_XRIGHT= max(0,shift0( FRACTO_X*Flux,0,-1))
DZ_BLOCK_YUP=       max(0,shift0( FRACTO_Y*Flux, 1,0))
DZ_BLOCK_YDOWN=    max(0,shift0(-FRACTO_Y*Flux,-1,0))
FluxDS                                                                                    =
ifthenelse(self.mask,DZ_BLOCK_XLEFT+DZ_BLOCK_XRIGHT+DZ_BLOCK_YDOWN+DZ_BLOCK_
YUP,0)
self.IceStore=ifthenelse(self.mask,max(0,self.IceStore+Snow+FluxDS-Flux-Abl),0)
self.IceStore=min(200,self.IceStore)
self.report(self.IceStore,"IceS")
rain=(precipitation/1000)-Snow
ET0=ET0=timeinputscalar("clim_00_07.tss",3)
ET0=(ET0+self.temperatureCorrection*self.ETc)/1000

SurQ=ifthenelse(self.IceStore<0.05,ifthenelse(rain>0.2*self.S,(sqr(rain-0.2*self.S))/(rain+0.8*
self.S),0),0)
pontentialS=max(0,rain+self.SoilS-self.kc*ET0-SurQ)
potentialSG=max(0,self.SoilS)
self.SoilS=ifthenelse(self.IceStore>0.05,potentialSG,min(self.MaxSoil,pontentialS))
Abl=self.gf*Abl
Recharge=ifthenelse(self.SoilS==self.MaxSoil,rain-self.kc*ET0+Abl*(1-self.gf)-SurQ,0)
Recharge=max(0,Recharge)
self.Qb=self.Qb*exp(-1*self.rc)+Recharge*(1-exp(-1*self.rc))
self.Qtot=Abl+self.Qb+SurQ
self.Qtot=(self.Qtot*90*90)/(24*3600)
QtotA=accuflux(self.ldd,self.Qtot)
QtotA=(1-self.kx)*QtotA+self.kx*self.QtotAold
QtotA=max(0,QtotA)
self.QtotAold=QtotA
self.discharge=self.QtotAold
```

```
        self.report(self.discharge,"q")

    def postmcloop(self):
        names = ["q"]
        mcaveragevariance(names, self.sampleNumbers(), self.timeSteps())
        percentiles = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
        mcpercentiles(names, percentiles, self.sampleNumbers(), self.timeSteps())

    def updateWeight(self):
        modelledMap = self.readmap("IceS")
        observedMap = self.readDeterministic("IceR")
        observedStdDevMap = ifthenelse(observedMap > 0, observedMap* 0.4, 0.01)
        sum = maptotal(((observedMap - modelledMap) ** 2) / (-2.0* (observedStdDevMap **
2)))
        weight = exp(sum)
        weightFloatingPoint, valid = cellvalue(weight, 1, 1)
        return weightFloatingPoint

    def suspend(self):

        self.reportState(self.IceStore, "IceS")
        self.reportState(self.Qtot, "RunO")

    def resume(self):

        self.IceStore = self.readState("IceS")

SnowmeltModel = SRModel()
dynamicModel = DynamicFramework(SnowmeltModel, lastTimeStep=2191, firstTimestep=1)
MonteCModel = MonteCarloFramework(dynamicModel, nrSamples=64)
ParticleFModel = SequentialImportanceResamplingFramework(MonteCModel)
pfModel.setFilterTimesteps([8,16,24,32,40,48,56,64,72,80,88,96,104,112,120,128,136,144,1
52,160,168,176,184,192,200,208,216,224,232,240,248,256,264,272,280,288,296,304,312,32
0,328,336,344,352,360,368,376,384,392,400,408,416,424,432,440,448,456,464,472,480,488
,496,504,512,520,528,536,544,552,560,568,576,584,592,600,608,616,624,632,640,648,656,
664,672,680,688,696,704,712,720,728,736,744,752,760,768,776,784,792,800,808,816,824,8
32,840,848,856,864,872,880,888,896,904,912,920,928,936,944,952,960,968,976,984,992,10
00,1008,1016,1024,1032,1040,1048,1056,1064,1072,1080,1088,1096,1104,1112,1120,1128,
1136,1144,1152,1160,1168,1176,1184,1192,1200,1208,1216,1224,1232,1240,1248,1256,12
64,1272,1280,1288,1296,1304,1312,1320,1328,1336,1344,1352,1360,1368,1376,1384,1392,
1400,1408,1416,1424,1432,1440,1448,1456,1464,1472,1480,1488,1496,1504,1512,1520,15
28,1536,1544,1552,1560,1568,1576,1584,1592,1600,1608,1616,1624,1632,1640,1648,1656,
1664,1672,1680,1688,1696,1704,1712,1720,1728,1736,1744,1752,1760,1768,1776,1784,17
92,1800,1808,1816,1824,1832,1840,1848,1856,1864,1872,1880,1888,1896,1904,1912,1920,
```

```
1928,1936,1944,1952,1960,1968,1976,1984,1992,2000,2008,2016,2024,2032,2040,2048,20
56,2064,2072,2080,2088,2096,2104,2112,2120,2128,2136,2144,2152,2160,2168,2176,2184]
)
ParticleFModel.run()
```