

Utah State University

DigitalCommons@USU

All Graduate Theses and Dissertations

Graduate Studies

5-2019

Advancing Streamflow Forecasts Through the Application of a Physically Based Energy Balance Snowmelt Model With Data Assimilation and Cyberinfrastructure Resources

Tseganeh Zekiewos Gichamo
Utah State University

Follow this and additional works at: <https://digitalcommons.usu.edu/etd>



Part of the [Civil and Environmental Engineering Commons](#)

Recommended Citation

Gichamo, Tseganeh Zekiewos, "Advancing Streamflow Forecasts Through the Application of a Physically Based Energy Balance Snowmelt Model With Data Assimilation and Cyberinfrastructure Resources" (2019). *All Graduate Theses and Dissertations*. 7463.

<https://digitalcommons.usu.edu/etd/7463>

This Dissertation is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Theses and Dissertations by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.



ADVANCING STREAMFLOW FORECASTS THROUGH THE APPLICATION OF A
PHYSICALLY BASED ENERGY BALANCE SNOWMELT MODEL WITH DATA
ASSIMILATION AND CYBERINFRASTRUCTURE RESOURCES

by

Tseganeh Zekiewos Gichamo

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Civil and Environmental Engineering

Approved:

David G. Tarboton, Sc.D.
Major Professor

Jeffery S. Horsburgh, Ph.D.
Committee Member

David E. Rosenberg, Ph.D.
Committee Member

Daniel W. Watson, Ph.D.
Committee Member

Gilberto E. Urroz, Ph.D.
Committee Member

Richard S. Inouye, Ph.D.
Vice Provost for Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2019

Copyright © Tseganeh Z Gichamo 2019

All Rights Reserved

ABSTRACT

Advancing Streamflow Forecasts through the Application of a
Physically based Energy Balance Snowmelt Model with Data
Assimilation and Cyberinfrastructure Resources

by

Tseganeh Z Gichamo, Doctor of Philosophy

Utah State University, 2019

Major Professor: Dr. David G. Tarboton
Department: Civil and Environmental Engineering

In many parts of the world, snow is a significant component of water resources. Currently many operational streamflow forecasting systems use temperature index snowmelt models that may have limited predictive capability for weather and land cover conditions different from the ones for which the models were calibrated. This study advances streamflow forecasting through the use of physically based models, assimilation of observed data, data services Cyberinfrastructure, and High Performance Computing (HPC). First, the Utah Energy Balance (UEB) snowmelt model was integrated into the Research Distributed Hydrologic Model (RDHM) framework. In this framework, approaches for assimilation of observed snow water equivalent (SWE) data into the UEB model and streamflow observations to update the soil moisture and stream channel states in RDHM were developed. The integrated UEB+RDHM models with the data assimilation were then evaluated for ensemble streamflow forecasting. In addition, a set

of web-based, hydrological data services called HydroDS was developed that provides access to hydrologic data and server side data processing tools. Finally, to enhance the ability of the models to be executed in HPC systems, two parallel versions of the UEB model were implemented using the Message Passing Interface (MPI) and the Compute Unified Device Architecture (CUDA) code on Graphics Processing Units (GPUs).

Results showed that the spatially distributed snow data assimilation approach improves the modeled SWE over the watershed grids, especially during snow accumulation period. The ensemble streamflow forecasts were also improved through the snow data assimilation, while the assimilation of streamflow observations did not add any improvement over that achieved by the SWE assimilation. Evaluation of HydroDS demonstrated the ability of the data services to reduce the time and effort spent by hydrologic modelers accessing and processing model inputs. Data processing workflows using HydroDS also enhance reproducibility and preserve provenance. Evaluation of the parallel UEB model with MPI implementation showed that although the computation kernel scales well with increased parallelization, the efficiency of the parallel code as a whole degrades due to poor scalability of input/output operations. Results from the CUDA GPU implementation demonstrated that obtaining performance comparable to that of other parallel processing methods with CUDA GPUs does not necessarily require major re-work of the existing UEB MPI code.

PUBLIC ABSTRACT

Advancing Streamflow Forecasts through the Application of a Physically based Energy Balance Snowmelt Model with Data Assimilation and Cyberinfrastructure Resources

Tseganeh Z Gichamo

The Colorado Basin River Forecast Center (CBRFC) provides forecasts of streamflow for purposes such as flood warning and water supply. Much of the water in these basins comes from spring snowmelt, and the forecasters at CBRFC currently employ a suite of models that include a temperature-index snowmelt model. While the temperature-index snowmelt model works well for weather and land cover conditions that do not deviate from those historically observed, the changing climate and alterations in land use necessitate the use of models that do not depend on calibrations based on past data. This dissertation reports work done to overcome these limitations through using a snowmelt model based on physically invariant principles that depends less on calibration and can directly accommodate weather and land use changes. The first part of the work developed an ability to update the conditions represented in the model based on observations, a process referred to as data assimilation, and evaluated resulting improvements to the snowmelt driven streamflow forecasts. The second part of the research was the development of web services that enable automated and efficient access to and processing of input data to the hydrological models as well as parallel processing methods that speed up model executions. These tasks enable the more detailed models and data assimilation methods to be more efficiently used for streamflow forecasts.

DEDICATION

To my wonderful parents, Tirunesh W. Ansebo and Zekiewos G. Anamo.

ACKNOWLEDGMENTS

This work is dedicated to my parents who sacrificed a lot so that my siblings and I could get the best quality education available from early age. I thank them for their love, patience, and being excellent role models of dedication and hard work. I would like to thank my sisters, brothers, and cousins for the love and support and belief in me.

I am very grateful to my advisor, Dr. David G. Tarboton, for the great opportunity to do this study under his mentorship. I would like to thank him for the generous financial support, continuous availability to provide guidance and feedback, and encouragement when things got tough. I would also like to thank my committee members, Drs. Jeffery S. Horsburgh, David E. Rosenberg, Daniel W. Watson, Gilberto E. Urroz, and Fred Ogden for their time and effort in providing valuable guidance throughout the process. I appreciate my former professors at IHE Delft, Drs. Ioana I. Popescu and Solomon D. Seyoum, and Dr. Richard Cuenca at Oregon State University for their kind help. I would also like to acknowledge Dr. Pabitra Dash, Software Architect at Tarboton research group, Dr. Gerald N. Day and his group at RTI International, and staff at the CBRFC for the opportunities to collaborate.

Fellow graduate students at CEE and friends at USU and in Logan: thank you for your friendship and making the life of graduate studentship a bit less stressful.

This work was supported by the National Science Foundation under collaborative grants EPS 1135482 and 1135483, and NASA award NNH15CM90C. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation and

NASA.

Compute, storage, support, and other resources from the Division of Research Computing in the Office of Research and Graduate Studies at Utah State University, Advanced Research Computing Center at the University of Wyoming, and Center for High Performance Computing at the University of Utah are gratefully acknowledged.

Tseganeh Z Gichamo

CONTENTS

	Page
ABSTRACT.....	iii
PUBLIC ABSTRACT	v
DEDICATION.....	vi
ACKNOWLEDGMENTS	vii
LIST OF TABLES.....	xii
LIST OF FIGURES	xiii
CHAPTER	
1. INTRODUCTION	1
1.1 Background.....	1
1.2 Research Questions	5
1.3 Objectives.....	6
1.4 Context	7
1.5 Summary and Organization	8
References	10
2. ENSEMBLE STREAMFLOW FORECASTING USING AN ENERGY BALANCE SNOWMELT MODEL COUPLED TO A DISTRIBUTED HYDROLOGIC MODEL WITH ASSIMILATION OF SNOW AND STREAMFLOW OBSERVATIONS.....	14
Abstract	14
2.1 Introduction	15
2.2 Data and Methods.....	19
2.2.1 Streamflow Forecast Scheme: Integrated UEB+RDHM Models.....	19
2.2.2 Study Watershed and Data	22
2.2.3 The Data Assimilation Methods: EnKF and PF	23
2.2.4 Implementation of the Ensemble Kalman Filter (EnKF) in UEB	26
2.2.5 Implementation of the Particle Filter (PF) in RDHM (SAC-SMA + rutpix7).....	36
2.2.6 Evaluation and Performance Metrics	39
2.3 Results and Discussion.....	42
2.3.1 Assimilation of SWE in UEB.....	42
2.3.2 Streamflow forecast with assimilation of SWE and Q in RDHM.....	45
2.4 Summary and Conclusions	48

References	51
3. DATA SERVICES IN SUPPORT OF PHYSICALLY BASED, DISTRIBUTED HYDROLOGICAL MODELS	74
Abstract	74
3.1 Introduction	76
3.2 Background.....	78
3.2.1 Geospatial Data Analyses for Hydrologic Models.....	78
3.2.2 Web based Data and Modelling Services.....	81
3.3 HydroDS: A set of Web-based Hydrologic Data Processing Services ...	83
3.3.1 Required Functionality	83
3.3.2 Design and Architecture	85
3.4 Case Study: Evaluation of HydroDS with Input Data Preparation for the Utah Energy Balance Snowmelt Model.....	89
3.4.1 Motivation	89
3.4.2 Study Watersheds and Input Data Pre-processing.....	90
3.5 Results and Discussion.....	91
3.6 Summary and Conclusions	96
References	98
4. UEB PARALLEL: DISTRIBUTED SNOW ACCUMULATION AND MELT MODELING USING PARALLEL COMPUTING	111
Abstract	111
4.1 Introduction	112
4.2 Factors to Consider in Parallel Programming.....	115
4.3 Methods	119
4.3.1 Utah Energy Balance (UEB) Model.....	119
4.3.2 UEB Parallel.....	121
4.3.3 Test Case Study: The Logan River Watershed.....	123
4.3.4 Description of Computing Resources.....	124
4.3.5 Performance Metrics	125
4.4 Results and Discussion	127
4.5 Summary and Conclusions	131
References	133
5. SUMMARY, CONCLUSIONS AND RECOMMENDATIONS.....	141
5.1 Summary and Conclusions	141
5.2 Recommendations	146
APPENDICES	149
Appendix: UEB Land Cover Variables Lookup Table based on NLCD	150

CURRICULUM VITAE.....151

LIST OF TABLES

Table	Page
2.1 SNOTEL stations with SWE data used in the study	56
2.2 Parameters of forcing perturbations for UEB ensemble generation.....	56
2.3 Steps for Particle Filter in SAC-SMA + rutpix7	57
3.1 UEB input processing time for the Logan River Watershed for the WY 2009	104
3.2 UEB input data pre-processing time using HydroDS for the WY 2009	105
4.1 Run times (seconds) and speed-up for different computing resources.....	137

LIST OF FIGURES

Figure	Page
2.1 Ensemble Streamflow Forecasting Scheme with UEB as RDHM component.....	58
2.2 Temporal organization of Ensemble Streamflow Forecast Procedure.....	58
2.3 Study watershed: Green River above Warren Bridge.	59
2.4 Illustration of the observation operator H for a hypothetical watershed with 59 grid cells (filled grid cells with different colors representing sub-watersheds) and 2 grids with observations. The observation operator is a 2 by 59 matrix where all but the two elements corresponding to the grid cells with observations have zero value.	60
2.5 Illustration of a hypothetical watershed with three observation stations handled separately from the 59 model grid cells (filled grid cells with different colors represent sub-watersheds). In this case, there is no need for special observation operator; the state matrix in observation space consists of the ensemble of model states for the three observation points.	61
2.6 Correlation between UEB simulated SWE at two SNOTEL sites.	62
2.7 Summary of Ensemble Kalman Filter steps in UEB.....	62
2.8 Snow water equivalent (SWE) at Loomis Park SNOTEL Station in Green River Watershed. UEB simulated SWE without data assimilation, ensembles with the EnKF assimilation of SWE and their mean, and observed SWE at this SNOTEL station are shown. At the dates of each data assimilation the spread of ensemble members contract towards the observations.....	63
2.9 Snow water equivalent (SWE) at Loomis Park SNOTEL Station in Green River Watershed. The labels for the error statistics, at the top-right, refer to RMSE with no data assimilation (RMSE_NODA), assimilation of observations from all four stations (RMSE_DAall), and assimilation excluding data at this site (RMSE_DAEclStn). Similar terminology applies to NSE and NIC. The data assimilation outputs are from the EnKF assimilation of SWE and the plots shown are the ensemble mean for each case.	64
2.10 Same as Figure 2.9 but for the Gunsight Pass SNOTEL station.....	65
2.11 Same as Figure 2.9 but for the New Fork Lake SNOTEL station.	65

Figure	Page
2.12 Same as Figure 2.9 but for the Kendall RS SNOTEL station.	66
2.13 Same as Figure 2.12 but the data assimilation extends only up to April 1, 2009.	67
2.14 Ensemble streamflow forecast at Warren Bridge near Daniel with no data assimilation (NO-DA) for the water year 2009.	68
2.15 Ensemble streamflow forecast at Warren Bridge near Daniel with SNOTEL SWE data assimilated every two weeks using EnKF (EnKF-SWE) for the water year 2009.	69
2.16 Ensemble streamflow forecast at Warren Bridge near Daniel with SNOTEL SWE data assimilated every two weeks using EnKF and assimilation of daily streamflow using PF with biweekly resampling (EnKF-SWE_PF-Q) for the water year 2009.	70
2.17 Scatter plots and error statistics of daily streamflow for the ensemble mean of April - September 2009 forecast discharge versus observed discharge at Warren Bridge near Daniel for the three scenarios No-DA, EnKF-SWE, and EnKF-SWE_PF-Q. The green line is the 1:1 line.	71
2.18 April – July volume error of ensemble streamflow forecast at Warren Bridge near Daniel for the three scenarios No-DA, EnKF-SWE, and EnKF-SWE_PF-Q for the water year 2009.	72
2.19 April – July volume error of ensemble streamflow forecast at Warren Bridge near Daniel for the three scenarios No-DA, EnKF-SWE, and EnKF-SWE_PF-Q for the water years 2005 - 2009.	73
3.1 Workflow for the Utah Energy Balance snowmelt model (UEB) input preparation.	106
3.2 High-level architecture of HydroDS. HydroDS comprises HydroDS Services and HydroDS Python Client Library.	107
3.3 A desktop, Google Map-based GUI program for accessing USGS DEM and watershed delineation through HydroDS. Google Map drawing tools are used to specify the bounding box around the watershed of interest and Google Map marker is used to select an approximate watershed outlet location.	108
3.4 Map showing study watersheds. These watersheds were selected as typical streamflow forecast entities in the CBRFC.	109

Figure	Page
3.5 Flow chart of the Utah Energy Balance snowmelt model (UEB) input data preparation steps.	110
4.1 Flow chart for the parallel version of the UEB model (UEB Parallel) with MPI.	137
4.2 Flow chart for the parallel version of UEB model (UEB Parallel) with GPU.	138
4.3 Study area: Location map of the Logan River watershed and its elevation (DEM).	138
4.4 Total simulation time (a), speed-up (b), efficiency (c), and ratio of IO time to total time (d) vs the number of processes for the test in Linux Cluster with MPI.	139
4.5 Speed-up (a) and efficiency (b) vs the number of processes for the test in Linux Cluster with MPI and reading multiple arrays at a time.	139
4.6 Speed-up and ideal speed-up from Amdahl's law revised to account for IO operation.	140

CHAPTER 1

INTRODUCTION

1.1 Background

Accumulated snow and glaciers form about 69% of the global fresh water storage (USGS, 2015). Hence, for many river systems, it is important to adequately model snow accumulation and melt in order to be able to forecast river flow quantity and timing for purposes of water supply, irrigation, energy production, flood control, maintenance of ecosystems, etc. The complexity of snow models, at least in a research context, vary from temperature index models to physically based, multi-dimensional models solving the energy and mass balance equations of the snowpack as well as simulating snow transport mechanisms causing spatial distribution of snow. Examples include wind driven drift and blowing of snow (Déry and Yau, 2002; Pomeroy and Essery, 1999) and effect of forest canopy on spatial pattern of snow accumulation and ablation (Mahat and Tarboton, 2014; Mahat et al., 2013). The effect of topography on wind and temperature results in heterogeneous snow distribution (Geddes et al., 2005; Mott et al., 2010). Other processes include rain and melt infiltration into the snowpack and refreezing as it advances into the snowpack (You et al., 2014); drainage of excess water through snowpack layer (Wever et al., 2014); and snow metamorphism that involves grain size change and accompanied changes in density, heat conductivity, and hydraulic conductivity (Lehning et al., 2002).

Currently, the National Weather Service (NWS) in the U.S. uses the SNOW-17 model (Anderson, 2006) as part of its hydrologic modeling suite for issuing operational streamflow forecasts such as river flood warnings or seasonal forecasts of inflow to

reservoirs from spring and summer snowmelt. SNOW-17 uses air temperature as an index for energy exchanges at the snow surface. It only requires two inputs: precipitation and temperature. The application of SNOW-17 to streamflow forecasting depends on the assumption that the calibrated relationship between temperature and snowmelt or freezing holds over the domain for which the model was calibrated. In addition, it relies on the fact that temperature is easy to measure and reliably forecast into a few days in the future in operational settings.

While temperature-index snowmelt models work quite well if they are adequately calibrated, they prove insufficient when the weather conditions significantly deviate from those for which the models were calibrated (Anderson, 2006). In addition, climate change is threatening to alter the hydrological dynamics in snow-dominated watersheds, as the proportion of precipitation falling as snow is expected to decrease and warming temperatures cause early onset of spring snowmelt, resulting in reduced natural storage of water in the form of snow. These, in combination with other factors such as altered evapotranspiration and dry soil moisture conditions, will affect the availability and quantity of summer and fall streamflow (Barnett et al., 2005; Barnett et al., 2008; Dozier, 2011). Early melt combined with rain-on-snow events can also exacerbate the risk of flooding. In addition, these changes may be spatially variable, e.g., a decrease in accumulated snow depth in one part of the basin may be accompanied by an increase in snow depth in another (Miller et al., 2011). Such changes in the hydrological dynamics of a watershed necessitate re-calibration of conceptual models, while weakening the statistical basis for using historical data for model calibration, and accentuate the need for using distributed models instead of lumped ones.

One of the prime motivators of current hydrological research is the need to understand and quantify the possible impacts on water resources of changes in climate, land cover, land use, population and urbanization (Biederman et al., 2015; Broxton et al., 2015; Fowler et al., 2007). An equally motivating factor is the desire to predict events such as floods or droughts anywhere globally and to develop mitigation strategies (Winsemius et al., 2013). In these situations, hydrologic models are required to represent multiple processes of the hydrologic cycle (Levine and Salvucci, 1999; Maxwell et al., 2014; Paniconi and Putti, 2015), and have sufficiently high resolution to represent the local process variability that affects aggregate basin scale (Kollet et al., 2010; Ogden et al., 2015; Qu and Duffy, 2007; Shi et al., 2013; Vivoni et al., 2011; Wood et al., 2011). Therefore, both in operational streamflow forecast settings and for the study of impact of changes in hydrologic dynamics, there is a need to use distributed, physically based hydrologic models. Using physically based energy balance snow models for operational streamflow forecasting has been a long-term goal of the NWS (Franz et al., 2008).

The rationale for distributed, physically based models is that they have state variables that represent the actual-physical processes present, and that these state variables are in principle measurable and comparable to observations from multiple sources. The models are thus amenable to improvement, either through use of these observations in calibration, or through updating the states based on observations, a method referred to as data assimilation. On the other hand, conceptual model state variables are not physically measurable and thus cannot benefit from more detailed direct observations. However, physically based models are more data demanding, and obtaining the extensive set of input data they require is a critical challenge. Leonard and

Duffy (2013) refer to the set of inputs needed to drive physically based distributed hydrologic models as “Essential Terrestrial Variables (ETV).” Obtaining ETV’s in a format organized for use in distributed models is a significant bottleneck in distributed hydrologic modelling. The ability (or lack thereof) to configure and populate the data structures of distributed models with data could enhance or hinder their use. Therefore, to realize the promise of distributed modeling through better representation of the physical environment, models need to be configured to automatically ingest the environmental information that is available (Lahoz and De Lannoy, 2013; Leonard and Duffy, 2013; Wood et al., 2011; Yu et al., 2013). Another challenge associated with the use of distributed, physically based models is that their computational burden is high. Therefore, models have to be designed to utilize high performance computation (HPC) technologies that are becoming increasingly available (Kollet et al., 2010; Wilkins-Diehr et al., 2008; Wood et al., 2011) but can still be challenging to use.

The use of physically based models entails added uncertainty arising from the additional input data required by the models and from model parametrizations of the various processes (Beven et al., 2015; Semanova and Beven, 2015). Increased model complexity introduces model structure and parameterization uncertainty in that some processes may not be sufficiently understood to be modelled in detail. There might also be sub-grid processes that are not properly accounted for. In addition to model structure and parametrization, two important sources of uncertainty in streamflow forecast are uncertainty in model states such as snow storage and soil moisture and uncertainty in the weather forcing data.

In operational streamflow forecasts, assimilation of data is an important factor

that can help reduce some of the uncertainty in model states and improve model predictions in real time (Clark et al., 2008; Pathiraja et al., 2016). Data assimilation is a procedure whereby observations are used to adjust and update model states so that forecasts based on the updated model states are improved. The model and observed values are combined based on the relative magnitudes of their uncertainties (Lahoz et al., 2010), represented by statistical measures such as their variances. The model state after data assimilation is expected to have reduced uncertainty. This way, process model and observations complement each other.

1.2 Research Questions

Given the challenges listed above the goal of this research was to investigate ways to enhance the application of a physically based, distributed, energy balance snowmelt model for streamflow forecasts. The following questions were addressed:

- Given the hydro-meteorological data currently available in operational settings, and given that there are only few SNOTEL stations in a forecast watershed, how much does the data from one station improve the simulated SWE at a remote location in a watershed?
- Can SWE data from the sparse SNOTEL stations in a watershed be used to update SWE for the whole watershed in a distributed model to arrive at better basin states in a way that improves streamflow forecast?
- Can assimilation of streamflow measured at the basin outlet into the integrated UEB+RDHM model result in an improved streamflow forecast beyond that which is achieved by assimilation of only snow observations?

- Do web-based hydrological data services facilitate the preparation of inputs to distributed hydrologic models compared to similar desktop-based tools, with respect to metrics such as data storage, processing speed, reproducibility and reliability? If they do help facilitate such data preparation, what are the factors that contribute to such improvements? What can be learned from the experience of designing and implementing such data services?
- How much do the implementations of parallel programming with the Message Passing Interface (MPI) and Graphics Processing Units (GPU) improve the computational performance of UEB, quantified in terms of the metrics speed-up and efficiency? What are the factors that affect these performance metrics?

1.3 Objectives

To address these questions the objectives of this dissertation were:

1. Developing a procedure to use observations of snow water equivalent to update model states and reduce streamflow forecast uncertainty.
2. Integration of the energy balance snowmelt model UEB in a distributed hydrologic model RDHM and evaluation of the integrated hydrologic model for streamflow forecasting, by accounting for the effect on the forecast streamflow of assimilated SNOTEL SWE and USGS gage discharge at basin outlet.
3. Design, implementation, and evaluation of web-based data services that enhance access to hydrologic data and geospatial data processing tools that enable preparation of model inputs in the format required by distributed hydrologic models.
4. Implementation and evaluation of parallel processing methods that take

advantage of high performance computation (HPC) resources in UEB, and consequently facilitate the evaluation and adoption of the physically based UEB model in operational settings such as streamflow forecast where computational time can be critical.

1.4 Context

This research was part of the CI-WATER and NASA-ROSES projects. One of the objectives of the CI-WATER project was enhancing data- and compute- intensive water resource modeling by developing a set of tools for input pre-processing, web services for data access, and gateway functionality to HPC centers. The tools are required to automatically generate modeling scenarios for a user-selected domain, thereby reducing the time and effort spent setting up and running simulation models. The objective of the NASA-ROSES project was to advance streamflow forecasts in the snow-dominated Colorado River basin. This included integration of a distributed, physically based, energy balance snowmelt model into a prototype of the National Weather Service (NWS) operational hydrologic modeling system and evaluating whether these enhancements result in improvements in the model. It also included development of the capability to assimilate observations into the model, by updating model state variables with these observations, and evaluating whether this data assimilation produces improvements in water supply forecasts.

I used the Utah Energy Balance snowmelt model (UEB) (Tarboton et al., 1995) to carry out these studies. The UEB model is a parsimonious, physically based, point energy and mass balance model with a single ground snowpack layer and a vegetation component that accounts for major snow processes in forested watersheds without undue

burden of over-parameterization of multi-layer models (Luce and Tarboton, 2010; Mahat and Tarboton, 2012, 2014; Mahat et al., 2013). It is driven by temperature, precipitation, radiation, humidity, and wind speed. Spatial variability of snow is accounted for either by using depletion curves (Luce and Tarboton, 2004) or by using a gridded approach (Sen Gupta et al., 2015). Its gridded configuration also promises to be amenable to parallel processing as model computations in each grid cell are independent, with interactions only required for aggregation of watershed or sub-watershed outputs.

I integrated the UEB model into the NWS's Research Distributed Hydrologic Model (RDHM) (Koren et al., 2004; NOAA's National Weather Service, 2008). RDHM is a modeling framework for gridded hydrologic simulation of watershed processes that provides a platform for coupling different components such as snowmelt, surface and subsurface runoff, stream routing. The UEB integrated RDHM model was run in 'reforecast mode,' i.e., run a historical simulation as if a forecast is being generated. The forecast output, ensemble streamflow at the basin outlet, was then be evaluated for cases with and without data assimilation.

1.5 Summary and Organization

The work described in this dissertation was organized into three research tasks, and the outcomes of each are presented in chapter formatted for publication as a separate paper.

Chapter 2 presents evaluation of the potential improvement to simulated snow water equivalent from assimilation of observed data. The Ensemble Kalman Filter (EnKF) was used to assimilate SNOTEL SWE data in UEB and model outputs were

compared to simulations without data assimilation. It was shown that due to the covariance between two points in a watershed at different modeling grid cells, data from sparse SNOTEL sites in a watershed can be used to update the SWE in the whole watershed. Further, ensemble streamflow forecast by a hydrologic model employing the UEB model was evaluated. For this, the UEB snowmelt model was coupled to a distributed hydrologic modeling framework RDHM with soil moisture accounting (SAC-SMA) and river routing (rutpix7) capabilities. In addition to SWE, observations of streamflow at a watershed outlet were assimilated using the Particle Filter (PF) method to update the SAC-SMA and rutpix7 states. Results showed that improved streamflow forecasts were obtained with assimilation of snow data, as compared to that with no data assimilation. On the other hand, there was little or no additional forecast improvement provided by the streamflow assimilation with PF beyond that achieved through snow data assimilation

The next two chapters, 3 and 4, involve the development and evaluation of Cyberinfrastructure resources to address issues that often present an impediment to the application of more detailed physically based models in a hydrologic modeling and forecasting environment. Chapter 3 discusses the design, implementation, and evaluation of a set of web-based, hydrological data processing services called HydroDS. HydroDS provides a number of data processing functionalities and ability to handle geospatial data in three widely used data formats: GeoTiff raster, shapefile, and multi-dimensional NetCDF. The data services are comprised of functions that can be used independently for a specific task or can be chained together in a workflow that integrates a number of related tasks involving geospatial data analyses and web-based data access. A Python

client library facilitates the scripting and execution of these workflows from a desktop computer, providing access to data processing tools from an accessible and relatively easy to use programming environment that is platform independent. HydroDS helps enhance reproducibility in hydrologic modeling, helps preserve the provenance of the hydrologic data processing steps, reduces time and effort by researchers accessing, organizing, and processing model input data, and contributes towards the goal of providing hydrologic ‘software as a service.’

Chapter 4 introduces parallel versions of the UEB snowmelt model where two parallel processing approaches were implemented and evaluated. The first approach was based on the Message Passing Interface (MPI) while the second one uses Graphics Processing Units (GPU). Evaluations showed that parallelizing the Input and Output (IO) operations in the model, in addition to computational operations, was a critical factor that affected the efficiency of the parallel code. In addition, the CUDA GPU implementation on one GPU node achieved slightly better performance compared to 64 processors with the MPI implementation without requiring major re-working of the existing UEB code.

References

- Anderson, E., 2006. Snow accumulation and ablation model–SNOW-17, NWSRFS Users Manual Documentation, Office of Hydrologic Development, NOAA’s National Weather Service.
- Andersson, E., Thépaut, J.-N., 2010. Assimilation of operational data, In: Lahoz, W., Khattatov, B., Ménard, R. (Eds.), *Data Assimilation: making sense of observations*. Springer, pp. 283-299.
- Barnett, T.P., Adam, J.C., Lettenmaier, D.P., 2005. Potential impacts of a warming climate on water availability in snow-dominated regions. *Nature* 438(7066) 303-309.
- Barnett, T.P., Pierce, D.W., Hidalgo, H.G., Bonfils, C., Santer, B.D., Das, T., Bala, G., Wood, A.W., Nozawa, T., Mirin, A.A., 2008. Human-induced changes in the hydrology of the western United States. *Science* 319(5866) 1080-1083.

- Beven, K., Cloke, H., Pappenberger, F., Lamb, R., Hunter, N., 2015. Hyperresolution information and hyperresolution ignorance in modelling the hydrology of the land surface. *Science China Earth Sciences* 58(1) 25-35.
- Biederman, J.A., Somor, A.J., Harpold, A.A., Gutmann, E.D., Breshears, D.D., Troch, P.A., Gochis, D.J., Scott, R.L., Meddens, A.J., Brooks, P.D., 2015. Recent tree die - off has little effect on streamflow in contrast to expected increases from historical studies. *Water Resources Research* 51(12) 9775-9789.
- Broxton, P., Harpold, A., Biederman, J., Troch, P.A., Molotch, N., Brooks, P.D., 2015. Quantifying the effects of vegetation structure on snow accumulation and ablation in mixed - conifer forests. *Ecohydrology* 8(6) 1073-1094.
- Clark, M.P., Rupp, D.E., Woods, R.A., Zheng, X., Ibbitt, R.P., Slater, A.G., Schmidt, J., Uddstrom, M.J., 2008. Hydrological data assimilation with the ensemble Kalman filter: Use of streamflow observations to update states in a distributed hydrological model. *Advances in Water Resources* 31(10) 1309-1324.
- Déry, S.J., Yau, M., 2002. Large - scale mass balance effects of blowing snow and surface sublimation. *Journal of Geophysical Research: Atmospheres* (1984 - 2012) 107(D23) ACL 8-1-ACL 8-17.
- Dozier, J., 2011. Mountain hydrology, snow color, and the fourth paradigm. *Eos, Transactions American Geophysical Union* 92(43) 373-374.
- Fowler, H., Blenkinsop, S., Tebaldi, C., 2007. Linking climate change modelling to impacts studies: recent advances in downscaling techniques for hydrological modelling. *International Journal of Climatology* 27(12) 1547-1578.
- Franz, K.J., Hogue, T.S., Sorooshian, S., 2008. Operational snow modeling: Addressing the challenges of an energy balance model for National Weather Service forecasts. *Journal of Hydrology* 360(1) 48-66.
- Geddes, C.A., Brown, D.G., Fagre, D.B., 2005. Topography and vegetation as predictors of snow water equivalent across the alpine treeline ecotone at Lee Ridge, Glacier National Park, Montana, USA. *Arctic, Antarctic, and Alpine Research* 37(2) 197-205.
- Kollet, S.J., Maxwell, R.M., Woodward, C.S., Smith, S., Vanderborght, J., Vereecken, H., Simmer, C., 2010. Proof of concept of regional scale hydrologic simulations at hydrologic resolution utilizing massively parallel computer resources. *Water Resources Research* 46(4).
- Koren, V., Reed, S., Smith, M., Zhang, Z., Seo, D.-J., 2004. Hydrology laboratory research modeling system (HL-RMS) of the US national weather service. *Journal of Hydrology* 291(3) 297-318.
- Lahoz, W., Khattatov, B., Menard, R., 2010. *Data assimilation: making sense of observations*. Springer Science & Business Media.
- Lahoz, W.A., De Lannoy, G.J., 2013. Closing the gaps in our knowledge of the hydrological cycle over land: conceptual problems. *Surveys in Geophysics* 1-38.
- Lehning, M., Bartelt, P., Brown, B., Fierz, C., Satyawali, P., 2002. A physical SNOWPACK model for the Swiss avalanche warning: Part II. Snow microstructure. *Cold Regions Science and Technology* 35(3) 147-167.
- Leonard, L., Duffy, C.J., 2013. Essential Terrestrial Variable data workflows for

- distributed water resources modeling. *Environmental Modelling & Software* 50 85-96.
- Levine, J.B., Salvucci, G.D., 1999. Equilibrium analysis of groundwater–vadose zone interactions and the resulting spatial distribution of hydrologic fluxes across a Canadian Prairie. *Water Resources Research* 35(5) 1369-1383.
- Luce, C.H., Tarboton, D.G., 2004. The application of depletion curves for parameterization of subgrid variability of snow. *Hydrological Processes* 18(8) 1409-1422.
- Luce, C.H., Tarboton, D.G., 2010. Evaluation of alternative formulae for calculation of surface temperature in snowmelt models using frequency analysis of temperature observations. *Hydrology and Earth System Sciences* 14(3) 535-543.
- Mahat, V., Tarboton, D.G., 2012. Canopy radiation transmission for an energy balance snowmelt model. *Water Resources Research* 48(1) W01534.
- Mahat, V., Tarboton, D.G., 2014. Representation of canopy snow interception, unloading and melt in a parsimonious snowmelt model. *Hydrological Processes* 28(26) 6320-6336.
- Mahat, V., Tarboton, D.G., Molotch, N.P., 2013. Testing above- and below-canopy representations of turbulent fluxes in an energy balance snowmelt model. *Water Resources Research* 49(2) 1107-1122.
- Maxwell, R.M., Putti, M., Meyerhoff, S., Delfs, J.O., Ferguson, I.M., Ivanov, V., Kim, J., Kolditz, O., Kollet, S.J., Kumar, M., 2014. Surface - subsurface model intercomparison: A first set of benchmark results to diagnose integrated hydrology and feedbacks. *Water Resources Research* 50(2) 1531-1549.
- Miller, W.P., Piechota, T.C., Gangopadhyay, S., Pruitt, T., 2011. Development of streamflow projections under changing climate conditions over Colorado River basin headwaters. *Hydrology and Earth System Sciences* 15(7) 2145.
- Mott, R., Schirmer, M., Bavay, M., Grünewald, T., Lehning, M., 2010. Understanding snow-transport processes shaping the mountain snow-cover. *The Cryosphere* 4(4) 545-559.
- NOAA's National Weather Service, H.D.M.R., 2008. The HL Research Distributed Hydrologic Model (HL-RDHM) Developer's Manual v. 2.0.
- Ogden, F.L., Lai, W., Steinke, R.C., 2015. ADHydro: QUASI-3D HIGH PERFORMANCE HYDROLOGICAL MODEL, SEDHYD 2015, 3rd Joint Federal Interagency Conference (10th Federal Interagency Sedimentation Conference and 5th Federal Interagency Hydrologic Modeling Conference): Reno, Nevada.
- Paniconi, C., Putti, M., 2015. Physically based modeling in catchment hydrology at 50: Survey and outlook. *Water Resources Research* 51(9) 7090-7129.
- Pathiraja, S., Marshall, L., Sharma, A., Moradkhani, H., 2016. Hydrologic modeling in dynamic catchments: A data assimilation approach. *Water Resources Research* 52(5) 3350-3372.
- Pomeroy, J., Essery, R., 1999. Turbulent fluxes during blowing snow: field tests of model sublimation predictions. *Hydrological Processes* 13(18) 2963-2975.
- Qu, Y., Duffy, C.J., 2007. A semidiscrete finite volume formulation for multiprocess

- watershed simulation. *Water Resources Research* 43(8).
- Semenova, O., Beven, K., 2015. Barriers to progress in distributed hydrological modelling. *Hydrological Processes* 29(8) 2074-2078.
- Sen Gupta, A., Tarboton, D., Hummel, P., Brown, M., Habib, S., 2015. Integration of an energy balance snowmelt model into an open source modeling framework. *Environmental Modelling & Software* 68 205-218.
- Shi, Y., Davis, K.J., Duffy, C.J., Yu, X., 2013. Development of a Coupled Land Surface Hydrologic Model and Evaluation at a Critical Zone Observatory. *Journal of Hydrometeorology* 14(5).
- Tarboton, D.G., Chowdhury, T.G., Jackson, T.H., 1995. A Spatially Distributed Energy Balance Snowmelt Model, In: K. A. Tonnessen, M.W.W.a.M.T. (Ed.), *Biogeochemistry of Seasonally Snow-Covered Catchments (Proceedings of a Boulder Symposium)*. IAHS, pp. 141-155.
- USGS, 2015. *Ice, Snow, and Glaciers: The Water Cycle*. .
- Vivoni, E.R., Mascaro, G., Mniszewski, S., Fasel, P., Springer, E.P., Ivanov, V.Y., Bras, R.L., 2011. Real-world hydrologic assessment of a fully-distributed hydrological model in a parallel computing environment. *Journal of Hydrology* 409(1) 483-496.
- Wever, N., Fierz, C., Mitterer, C., Hirashima, H., Lehning, M., 2014. Solving Richards Equation for snow improves snowpack meltwater runoff estimations in detailed multi-layer snowpack model. *The Cryosphere* 8(1) 257-274.
- Wilkins-Diehr, N., Gannon, D., Klimeck, G., Oster, S., Pamidighantam, S., 2008. TeraGrid science gateways and their impact on science. *Computer* 41(11) 32-41.
- Winsemius, H., Van Beek, L., Jongman, B., Ward, P., Bouwman, A., 2013. A framework for global river flood risk assessments. *Hydrology and Earth System Sciences* 17(5) 1871-1892.
- Wood, E.F., Roundy, J.K., Troy, T.J., van Beek, L.P.H., Bierkens, M.F.P., Blyth, E., de Roo, A., Döll, P., Ek, M., Famiglietti, J., Gochis, D., van de Giesen, N., Houser, P., Jaffé, P.R., Kollet, S., Lehner, B., Lettenmaier, D.P., Peters-Lidard, C., Sivapalan, M., Sheffield, J., Wade, A., Whitehead, P., 2011. Hyperresolution global land surface modeling: Meeting a grand challenge for monitoring Earth's terrestrial water. *Water Resources Research* 47(5) W05301.
- You, J., Tarboton, D., Luce, C., 2014. Modeling the snow surface temperature with a one-layer energy balance snowmelt model. *Hydrology and Earth System Sciences* 18(12) 5061-5076.
- Yu, X., Bhatt, G., Duffy, C., Shi, Y., 2013. Parameterization for distributed watershed modeling using national data and evolutionary algorithm. *Computers & Geosciences* 58 80-90.

CHAPTER 2

ENSEMBLE STREAMFLOW FORECASTING USING AN ENERGY BALANCE
SNOWMELT MODEL COUPLED TO A DISTRIBUTED HYDROLOGIC MODEL
WITH ASSIMILATION OF SNOW AND STREAMFLOW OBSERVATIONS¹**Abstract**

In many parts of the world, the quantity and timing of streamflow depend on processes of snow accumulation and melt. However, detailed snowmelt modeling is often hampered by limited input data availability and uncertainty arising from inadequate model structure and parametrization. This is particularly true in operational streamflow forecasting where modelers do not have the benefit of post-processed data. Data assimilation that updates model states based on observations provides a way to reduce uncertainty and improve model forecasts of streamflow. In this study, we evaluated coupled snowmelt, soil moisture accounting, and river routing models for streamflow forecasting in a snow dominated headwater watershed. We integrated the Utah Energy Balance (UEB) snowmelt model into the Research Distributed Hydrologic Model (RDHM), which has implementations for the Sacramento Soil Moisture Accounting (SAC-SMA) and rutpix7 river routing models. We implemented two different data assimilation methods for different parts of the model: the Ensemble Kalman Filter (EnKF) for assimilation of snow water equivalent (SWE) observations from Snow Telemetry (SNOTEL) stations in UEB and the Particle Filter (PF) for assimilation of streamflow to update the SAC-SMA and rutpix7 states. Using leave one out validation, it

¹ Authors: Tseganeh Z Gichamo, David G. Tarboton.

was shown that the modeled SWE at an observation location whose observations were excluded from the data assimilation was improved through assimilation of data from other stations, especially during the snow accumulation period. This suggests that assimilation of sparse observations of SWE has the potential to improve the distributed modeling of SWE over watershed grid cells where SWE is not observed. In addition, the distributed snow data assimilation resulted in improved ensemble streamflow forecasts. For example, the Nash-Sutcliffe Efficiency (NSE) for the ensemble mean of forecasted streamflow improved from 0.52 with no data assimilation to 0.83 with snow data assimilation, and the April – July volume error was reduced by 94 %. On the other hand, the assimilation of streamflow observations using the Particle Filter (PF) method did not provide additional forecast improvement over that achieved by the SWE assimilation.

Keywords—Ensemble streamflow forecast; Utah Energy Balance (UEB) snowmelt model; Research Distributed Hydrologic Model (RDHM); data assimilation; Ensemble Kalman Filter (EnKF); Particle Filter (PF);

2.1 Introduction

In many parts of the world, as it is in the intermountain US, snow is a significant component of water resources. Hence, it is important to adequately model snow accumulation and melt in order to be able to forecast the quantity and timing of streamflow for purposes of water supply, energy production, flood control, maintaining the ecosystem, etc. In terms of complexity, snow models vary from lumped conceptual models to physically based, multi-dimensional models solving the energy and mass balance of the snowpack and spatial redistribution of snow such as due to wind

(Anderson, 1976; Bartelt and Lehning, 2002; Déry and Yau, 2002; Geddes et al., 2005; Jin et al., 1999; Jordan, 1991; Liston and Sturm, 1998; Mahat and Tarboton, 2014; Mahat et al., 2013; Mott et al., 2010; Pomeroy and Essery, 1999; Tarboton et al., 1995; You et al., 2014; Lehning et al., 2002; Wever et al., 2014).

Currently, the U.S. National Weather Service (NWS) uses the SNOW-17 model (Anderson, 2006) for streamflow forecasts. SNOW-17 uses temperature as an index for energy exchanges at the snow surface, and it only requires two inputs: precipitation and temperature. The application of SNOW-17 to streamflow forecasting depends on the assumption that the calibrated relationship between temperature and snowmelt holds over the domain for which the model was calibrated. In addition, it relies on the fact that temperature is easy to measure and accurately forecast for a few days into the future in operational settings. While temperature-index snowmelt models work quite well if they are adequately calibrated, they prove insufficient when the weather conditions significantly deviate from those for which the models were calibrated (Anderson, 2006). In addition, conceptual models that rely on calibration of parameters based on historical data may be limited when there is a shift in hydrologic regime due to changes in climate, land cover, land use, urbanization etc. (Biederman et al., 2015; Broxton et al., 2015; Fowler et al., 2007).

Using physically based, energy balance snow models for operational streamflow forecasting offers the opportunity to overcome some of these shortcomings and has been a goal of the U.S. NWS (Franz et al., 2008). While physically based models reduce the uncertainty due to limited process representations and the overdependence on calibration in conceptual models, their application on the other hand entails added uncertainty arising

from the additional input data requirements and parameterization uncertainty introduced by increased model complexity (Beven *et al.*, 2015; Semanova and Beven, 2015).

In operational streamflow forecasts, assimilation of observed data is used to reduce forecast uncertainty by conditioning the forecasts on best possible model states at the time of the forecast (Clark *et al.*, 2008; Franz *et al.*, 2003; Pathiraja *et al.*, 2016). In data assimilation, observations are used to adjust and update model states balancing observation and model uncertainty represented by a statistical measure such as error variance. In snowmelt driven streamflow forecasts, the updated model states that the forecasts are conditioned on include snowpack, soil moisture, and stream channel states.

Prior studies with respect to assimilation of observations in hydrologic models have focused on a specific process and data related to it. Examples of these include assimilation of soil moisture related data (Reichle *et al.*, 2002), assimilation of data related to snow (Clark *et al.*, 2006; Slater and Clark, 2006; Su *et al.*, 2008), assimilation of streamflow observations (Abaza *et al.*, 2014; Clark *et al.*, 2008). However, in a model that integrates multiple processes, there is an opportunity to assimilate multiple data types into the different model components with the potential to obtain improved outputs such as streamflow at basin outlet. The different observation types used in such multi-data assimilation complement each other where the limitation of one observation may be compensated by another observation type (Franz *et al.*, 2014). It is also possible to apply different assimilation methods customized to different components of the integrated modeling system. For example, assimilation of streamflow using the Ensemble Kalman Filter (EnKF) or its variants may not be very efficient to update storage states such as snow water equivalent (SWE) or soil moisture. This is likely due to the non-linearity of

the relationship between the internal (watershed) model states and the streamflow at the outlet (Clark *et al.*, 2008) that makes it difficult to compute the cross-covariances between states and outputs on which the EnKF relies (Rakovec *et al.*, 2012).

Point observations such as those from Snow Telemetry (SNOTEL) (<https://www.wcc.nrcs.usda.gov/snow/>) stations provide an opportunity to, through assimilation, update model SWE states to improve forecast initial conditions. However SNOTEL observations are sparse, and an approach is needed to propagate information from these sparse sites to each model grid cell. Slater and Clark (2006) interpolated normalized standard deviates of SWE to each grid cell, and then inferred grid cell SWE from an ensemble of historic model simulations at that grid cell. This interpolated SWE was treated as an observation and used in an EnKF to separately assimilate SWE at each grid cell (Slater and Clark, 2006). As an alternative to this treatment of an interpolation as an observation there is the potential to use the EnKF to directly propagate information from observation sites to unobserved model grid cells through the spatial correlation of SWE. This can be done, for example, by augmenting the model grid SWE state vector with SWE states at observation locations and then having the EnKF observation function act on observation location elements of the augmented state vector.

In order to contribute towards the goal of adopting more physically based models in operational streamflow forecasting, and to evaluate the use of an energy balance snowmelt model as part of the modeling suite used in the NWS forecasting system we integrated the Utah Energy Balance (UEB) snowmelt model into the Research Distributed Hydrologic Model (RDHM), which has implementations for the Sacramento Soil Moisture Accounting (SAC-SMA) and rutpix7 river routing models. In addition, to

improve the ability to assimilate snow and streamflow observations to their respective model components and evaluate the potential improvement to forecast streamflow, we implemented two different data assimilation methods. The EnKF was used to assimilate point SWE from SNOTEL in UEB and streamflow data were assimilated using the Particle Filter (PF) method to update SAC-SMA and rutpix7 model states. The PF was selected for streamflow assimilation because, as stated earlier, the EnKF may struggle to accurately represent the cross-covariances between soil moisture and streamflow at the outlet.

This paper is organized as follows. In Section 2.2, descriptions of the streamflow forecast scheme, the data assimilation methods, the study watersheds and model data, and the performance metrics are provided. Results and discussion are given in Section 2.3, followed by summary and conclusions in Section 2.4.

2.2 Data and Methods

2.2.1 *Streamflow Forecast Scheme: Integrated UEB+RDHM Models*

Figure 2.1 shows the ensemble streamflow forecasting scheme. Its major components are the Utah Energy Balance (UEB) snowmelt model and its integration into the Research Distributed Hydrologic Model (RDHM), the generation of ensemble forcing inputs, the snow and streamflow data assimilation, and ensemble streamflow forecasting. The UEB model represents the major physical processes critical for snow accumulation and ablation (Luce and Tarboton, 2010; Mahat and Tarboton, 2012, 2014; Mahat et al., 2013; Tarboton et al., 1995; You et al., 2014). As a single layer model, UEB is parsimonious, avoiding some of the complexity and input needed for more detailed multi-

layered snowmelt models. This makes it a promising candidate for operational streamflow forecasting where computational time can be critical, and where there is interest in incrementally evaluating the improvements possible through adding better physical process representations.

RDHM is a modeling framework for gridded hydrologic simulation of watershed processes that comprises a platform for coupling different components and facilitating the transfer of data among different modeling components during run-time (Koren et al., 2004; NOAA's National Weather Service, 2008). Inside the integrated UEB+RDHM framework, the UEB model was run first. The rain + melt (Rmelt) output from UEB provides input to the Sacramento Soil Moisture Accounting (SAC-SMA) model (Burnash and Singh, 1995; Burnash et al., 1973), which simulates runoff that is subsequently routed to the basin outlet with the kinematic wave routing model rutpix7 (NOAA's National Weather Service, 2008) to generate streamflow.

The ensemble streamflow forecasting procedure is illustrated in Figure 2.2. A model run starts at the beginning of the water year, October 1, and proceeds up to the 'forecast time' using observed data for that year as input. The forecast time is the point when the streamflow forecasts are issued. The streamflow forecast is then run for a duration labeled 'forecast period', using input forcing data from all available historic years. This results in the ensemble of possible future traces depicted in Figure 2.2. In this study, the forecast period was from April 1 to September 30. The forecast period is comprised of 'lead time' and a 'forecast window'. The forecast window is the time span for which the forecast is sought after, while lead time is a period between the issuing of forecast and the start of the forecast window. Lead time provides water managers time to

prepare to deal with the forecasted event. The forecast time for this study was chosen to be April 1, and the forecast period and forecast window in this study were the same, between April 1 and September 30 (thus the lead time was 0). The choice of April 1 for forecast time was based on the fact that for most watersheds in the western U.S., the snowmelt season starts around this date. In addition, for the purposes of reservoir operations and other water allocation and management practices, the April – July streamflow volume is one of the primary variables of interest at the Colorado Basin River Forecast Center (CBRFC).

Figure 2.2 depicts a single simulation trace based on observed forcing data up to the forecast time as an illustration of the general idea. However in this study we ran the model simulation between the start of simulation (October 1) and forecast time (April 1) using multiple traces that account for input uncertainty with assimilation of snow water equivalent and streamflow used to update model states, balancing and trading off uncertainty due to the inputs and observations. The objective of this was to arrive at a set of states that represent the best possible estimate of initial state, and its uncertainty, and to use this to initialize the forecast. The Ensemble Kalman Filter (EnKF) was used for assimilation of snow data, while the Particle Filter was used to assimilate streamflow observations at the watershed outlet. Perturbed observations for the forecast year were used to represent input forcing uncertainty and input to the EnKF. The implementations of the data assimilation methods are described in Section 2.2.3 below. The ensemble simulation for the forecast period uses multiple realizations of future weather conditions from numerical weather prediction (NWP) models to account for uncertainties in forecast weather forcing (Cloke and Pappenberger, 2009). Alternatively, long-term observations

or re-analysis of historical weather forcing are taken to represent samples of the likely distribution of the future weather conditions (Day, 1985; Franz et al., 2003; Wood and Lettenmaier, 2008). The latter approach was used in this study.

2.2.2 *Study Watershed and Data*

The study watershed used to evaluate the work in this chapter is the Green River watershed above Warren Bridge (Figure 2.3). Observed snow water equivalent (SWE) data from four SNOTEL stations shown in Figure 2.3 and summarized in Table 2.1 were assimilated in UEB. We evaluated the SWE assimilation at the observation points for the water year 2009 (October 1, 2008 – September 30, 2009). The water year 2009 was selected because of the availability of gridded weather forcing data from the Colorado Basin River Forecast Center (CBRFC) for that year, at the start of this study. For streamflow forecast evaluation, we used water years 2005 to 2009.

We used a 30 m Digital Elevation Model (DEM) from the USGS to extract terrain variables slope and aspect. Canopy variables were generated based on the 2011 National Land Cover Database (NLCD) (Homer et al., 2015). We used precipitation and air temperature data from CBRFC. These are gridded datasets with a resolution of 800 m produced using interpolation based on the PRISM (Parameter-elevation Relationships on Independent Slopes Model) dataset (Daly et al., 2008; Daly et al., 1994). Humidity and wind speed data were obtained from the North American Land Data Assimilation System (NLDAS) forcing datasets (Mitchell et al., 2004). Solar and longwave radiation were parameterized based on air temperature and humidity in UEB (Tarboton et al., 1995; You et al., 2014). The NLDAS datasets were downscaled, with elevation adjustment, following a downscaling methodology described by Sen Gupta and Tarboton (2016). For

downscaling, the DEM from NLDAS represents the elevation of the NLDAS datasets while the elevation from the USGS DEM at the center of the grid cell was the target elevation to which the forcing data were downscaled.

The simulations were carried out on a grid with cells of size 0.25 Hydrologic Rainfall Analysis Project (HRAP) units (Reed and Maidment, 1999). HRAP is a coordinate system used by the NWS for gridded modeling, and 0.25 HRAP units corresponds to about 1190 m. RDHM takes as an input a connectivity file that defines the model domain(s), resolution, and outlet location(s) for a given watershed (or multiple watersheds). It is generated from digital elevation model (DEM) data and specifies grid cell connectivity. During runtime, RDHM resamples the forcing, terrain, and land cover data inputs at the centers of the grid cells in the connectivity file. Given this configuration and given that the terrain and land cover data were sampled from the 30 m resolution DEM, the UEB model at each 0.25 HRAP (~1190 m) grid cell represents a point with 30 m support scale (or footprint) with 1190 m spacing. This scheme is premised on the assumption that these points at 1190 m spacing aggregated over the watershed represent the distribution of snow and snowmelt input over the watershed.

2.2.3 *The Data Assimilation Methods: EnKF and PF*

In data assimilation, a model is run forward in time, up to a point when observations are available. Then the observations are used to adjust and update model states to reduce their uncertainty. These improved model states are then used to further propagate the model forward in time, to the point of another observation update, or for a forecast. During the assimilation, modeled and observed information is combined based on the relative magnitudes of their uncertainties (Lahoz et al., 2010). The resulting

combined information from model and observation is expected to reduce the uncertainty of the predicted variable, e.g., by reducing the variance, compared to that of either the model or the observations by themselves. In addition, simulation model and state observations complement each other in that the observations often present a more accurate value of the state but at sparse temporal and spatial sampling intervals, while the simulation model represents spatially and temporally continuous system dynamics.

One of the most popular data assimilation methods is the Kalman Filter, in which model predicted and observed state values are linearly combined. The difference between observed and model predicted values, referred to as the residual, is multiplied by a weighting factor and is added to the model predicted value to obtain the updated or assimilated value (Reichle, 2008). In the Kalman Filter, the weighting factor, called the Kalman Gain, is computed as the ratio of the variance of the model prediction error to the sum of variances of the observation and model prediction errors (Brown and Hwang, 2012).

The Kalman filter provides the optimal estimate in the least mean squared error sense, dependent on the assumption that the underlying distributions of both model prediction and observations are Gaussian and the system being modeled is linear (Drécourt, 2003). Methods that rely on linearization of process model equations such as the Extended Kalman Filter (EKF) and the variance methods such as the 4D VAR have been applied extensively over the years (Nichols, 2010). The requirement for linearization is a big challenge in terms of computational burden and the strong nonlinearities some environmental models exhibit. Alternative methods that do not require linearization of the model equations include those that derive from Monte Carlo

sampling methods, such as the Particle Filter, and variants of the KF that also employ sampling methods such as the Unscented Kalman Filter and the Ensemble Kalman Filter (Liu et al., 2012).

The Ensemble Kalman Filter (EnKF) is a form of Kalman Filter that has been widely used in earth sciences and hydrology (Clark et al., 2008; Drécourt, 2003; Kumar et al., 2008; Liu et al., 2012). The EnKF relies on execution of Monte Carlo simulations to generate multiple realizations of model states that are assumed to be equally likely. The covariance computed based on the ensemble members represents the model error covariance from which the Kalman Gain is computed (Evensen, 2003). The filter can be used for non-linear processes and, while optimal filtering requires the distribution underlying the ensembles to be Gaussian, it has been shown that the EnKF provides sufficiently satisfactory, although sub-optimal, performance for systems that deviate from Gaussian distribution or in systems where the underlying distribution is unknown (Reichle et al., 2002; Zhou et al., 2006).

The Particle Filter (PF) is a data assimilation method that is similar to the EnKF in that it also relies on Monte Carlo simulation and generation of multiple realization of possible model states. However, in PF the particles (each particle representing a vector of state variables in state space) are not necessarily assumed to be equally likely. Rather, the probability distribution of the model states are represented by a set of particles (similar to ensemble members in EnKF) and associated weights (Brown and Hwang, 2012; Grewal and Andrews, 2015; Labbe, 2015). In addition, in PF the update step does not involve computation of Kalman Gain and adjustment of ensemble members. Instead, the particles' weights are updated based on their (particles') distance from the observations.

Particles closer to the observations are given higher weights and vice-versa. Then the particle weights are used to select those that are close to the observations and generate more particles with similar characteristics while discarding particles that are too far away from observations.

2.2.4 *Implementation of the Ensemble Kalman Filter (EnKF) in UEB*

Hydrological and land surface processes are said to be “damped” (Reichle, 2008). This is to say that, unlike atmospheric models, which represent chaotic processes in which small errors in the initial conditions of states may be amplified as the process evolves forward, the primary sources of uncertainty in hydrological models are atmospheric forcing inputs, model dynamics, and parameters (Reichle et al., 2002). The effect of errors in initial conditions diminishes over time. Hence, in this study, the ensemble realizations were generated by driving the model with ensemble forcing data obtained from perturbation of input forcing by adding randomly sampled small errors. This approach is similar to that taken by other snow data assimilation studies (e.g., Slater and Clark, 2006). The errors for the forcing perturbation were sampled from a multivariate normal distribution generated across the grid cells in a watershed.

The UEB model state considered for assimilation is the snow water equivalent (SWE) at each grid cell. The data used for assimilation in this study consisted of observed SWE at SNOTEL stations. Other UEB model states include the snowpack energy content and dimensionless age of snow surface (also known as snow surface condition). We considered only SWE for assimilation because updating the other state variables would require covariance/correlation between the observed SWE and the other states. For example, updating bulk snowpack energy based on observed SWE is only possible if the

covariance between these two variables can be obtained. However, there is no clear covariance/correlation between the two, e.g., depending on net energy inputs, snowpack energy content may increase or decrease while the snowpack water equivalent (SWE) remains the same. Therefore, the state variables other than SWE were allowed to evolve in the ensembles but were not updated.

Four SNOTEL stations in the study watershed and in the area surrounding it were used in this study. This sparsity is typical for watersheds such as this. One approach used to update the SWE over the whole watershed based on the few observation points is interpolation of SWE from the sparse SNOTEL stations across the model domain before assimilating it with the model simulated SWE at each grid cell. Instead of the direct interpolation of SWE, Slater and Clark (2006) interpolated the normalized standard deviates of SWE (also known as z scores) before back-computing the assimilated SWE.

In this study, we used an approach that depends on the assumption that the SWE between two grid cells in the study watersheds are correlated, and their covariance can be captured from the ensemble of simulated SWE by the energy balance snow model. This assumption arises from the following consideration: Spatial distribution of watershed snowpack depends on spatial variability of factors that include watershed topography, land cover, and weather forcing such as precipitation, temperature, humidity, wind, and radiation (Clark et al., 2011; Luce, 2000). If a distributed, energy balance snow model accounts for these variabilities, then one can expect the spatial variability of the snowpack states to be captured by such a model. It follows, then, that the ensemble of model simulations at a grid cell, if adequately sampled, represents the distribution of the model states in that grid cell. And, statistical measures such as the mean and standard

deviation can be computed from such a distribution. Moreover, the distributions of any two grid cells can be used to compute the covariance between the states at the two grid cells. If one of the grid cells happens to have observations, the covariance so computed can be used together with the variance of the observation to apply the adjustment/update to the grid cell that does not have observed data because in Kalman Filter the Gain is proportional to the covariance of simulated distributions and the variance of observation.

The steps that were followed for implementation of the EnKF in UEB are presented below. In the equations that follow, all the variables with upper case, e.g., X , Y ...are matrices, and those with lower case, such as u , represent vectors. The subscripts refer to the time step while the superscripts differentiate between the prediction, also known as the background state, e.g., x_k^b , prior to update, and state after update, also called analysis, e.g., x_k^a (Lahoz et al., 2010). Thus, given a model domain of N locations comprising N_g centers of model grid cells and m measurement sites ($N=N_g+m$), the state vector at time step k , x_k^a is represented as,

$$x_k^a = \begin{pmatrix} W_1 \\ W_2 \\ \dots \\ W_N \end{pmatrix} \quad (1)$$

where W is the UEB model snow water equivalent, and W_i refers to snow water equivalent at location i . The m measurement sites do not necessarily coincide with grid cell centers and have their own parameters determined from the DEM, land cover, and their own downscaled inputs and are thus separate grid cells where the model is run. Following standard data assimilation literature notation, observations of W at the m measurement grid cells are represented by the vector z_o .

$$z_o = \begin{pmatrix} z_1 \\ z_2 \\ \dots \\ z_m \end{pmatrix} \quad (2)$$

The general procedure followed at a given time step involves running the UEB model at all N grid cells in the watershed in ensemble mode and computing the model error covariances from the ensemble of states at each grid cell. This is followed by computing the Kalman gain from the covariances and the variances of the states. The Kalman gain is then used with observations to adjust/update the simulated SWE. Steps 1, 2, and 3 below represent generation of ensemble UEB states and are applicable at all model times. Thus, the model evolves the ensemble of UEB states through time, regardless of whether there are observed data for assimilation or not. Steps 4 through 11 however are executed only at model steps with observed data. Note also that the procedure below is applicable for the time period before the forecast time as there is no data assimilation during the forecast. Beyond the forecast time, each year of historical weather forcing data is used to generate model states for a forecast ensemble member.

1. Perturb meteorological forcing such as precipitation and temperature.

$$U_k = u_k * \theta_u, \theta_u = N(1, S_u) \text{ for all forcings except temperature} \quad (3)$$

$$U_k = u_k + \theta_u, \theta_u = N(0, S_u) \text{ for temperature} \quad (4)$$

Where u_k is an input vector of forcing at all grid cells (size N). U_k (N x E) is the generated ensemble forcing. E is the ensemble size, and therefore, E forcing inputs were generated at each grid cell by perturbing the input forcing according to the covariance structure between grid cells represented by S_u . There are six UEB input forcing variables: precipitation (P), air temperature (Ta), wind speed (V), relative humidity (RH), solar

radiation (Qsi), and atmospheric/longwave radiation (Qli). The multiplier factors (or additive factor for temperature) θ_u were sampled from a multivariate Gaussian distribution with mean 1 (0 for temperature) and covariance S_u . For all the forcing variables except temperature, if this simulation results in a multiplier less than 0 (which happens with very small probability) the multiplier is set to 0 to preclude negative values that are possible with a Gaussian distribution. The covariance matrix S_u , is computed based on the spatial correlation between the forcing variables in different grid cells and the standard deviation of each forcing. We assumed correlation between grid cells that is exponentially decaying with distance.

In addition to the spatial correlation that is an exponential function of distance between model grids, the correlations among the errors in precipitation, solar radiation, and longwave radiation were accounted for based on values from previous studies (Kumar et al., 2014; Kumar et al., 2009; Xue et al., 2018). For temperature, wind speed, and relative humidity, the forcing errors were assumed to be uncorrelated, because we have no information on their relationships with the other variables.

Following Clark *et al.* (2008) and Evensen (2003), the temporal correlation in forcing errors was modeled as:

$$S_{t+1} = \rho S_t + \sqrt{1 - \rho^2} w_t \quad (5)$$

where w is a spatially correlated random variable sampled from a standard normal distribution (zero mean and variance 1) and ρ is the temporal persistence parameter quantified as:

$$\rho = 1 - \frac{\Delta t}{\tau} \quad (6)$$

where Δt is the model time step and the decorrelation time parameter τ quantifies the rate at which forcing error correlation decays with time.

Parameters determining these correlations and perturbations are the standard deviation for the forcing variables, the spatial correlation length, and decorrelation time. These parameters were selected similar to values from the literature and with some trial and error adjustments so that the ensemble mean of model simulations with perturbed forcing and without data assimilation is unbiased. The parameters used in this study for the generation of ensemble forcing are shown in Table 2.2.

Often, when running the Ensemble Kalman Filter, the initial states are also perturbed to get a distribution of initial states. In this study, all the simulations start at the beginning of the water year when there is no snow on the ground. Therefore, all ensemble members have the same initial state values of zero SWE.

2. Run ensemble of simulations using the perturbed forcing.

$$X_{k+1}^b = M_{k,k+1}(X_k^a, U_k) \quad (7)$$

where X_{k+1}^b ($N \times E$) is a matrix of UEB predicted/background states. $M_{k,k+1}$ represents the model run evolving the states from time step k to time step $k+1$. In other words, at each grid cell the model is run E times (with perturbed forcing) for time step k and produces an ensemble of state vectors of size E at the next time step $k+1$. All the remaining steps hereafter refer to the time step $k+1$; hence, the time subscript is dropped, and hereafter the matrix subscripts refer to matrix elements. The matrix for the ensemble of background states is thus

$$X^b = (x_1^b, x_2^b, \dots, x_E^b) \quad (8a)$$

where E = ensemble size and the ensemble matrix is comprised of state vectors x_i^b giving snow water equivalent at each grid cell for each ensemble member:

$$x_i^b = \begin{pmatrix} W_{1i}^b \\ W_{2i}^b \\ \vdots \\ W_{Ni}^b \end{pmatrix} \quad (8b)$$

3. Compute ensemble mean \bar{x}^b

$$\bar{x}^b = \frac{1}{E} \sum_{i=1}^E x_i^b \quad (9)$$

4. Compute ensemble anomaly X' : subtract the ensemble mean vector from each column in the ensemble states matrix.

$$x'_i = x_i^b - \bar{x}^b \quad (10)$$

and form the anomalies matrix, X' , comprised of anomaly column vectors

$$X' = (x'_1, x'_2, \dots, x'_E) \quad (11)$$

5. Compute model error covariance P_{xx} using the ensemble anomaly.

$$P_{xx} \approx \frac{1}{E-1} X' (X')^T \quad (12)$$

The model error covariance is a function of individual variances and covariances.

For N grid cells, the matrix is symmetric and takes the following form:

$$P_{xx} = \begin{pmatrix} \text{var}(W_1) & \text{covar}(W_1, W_2) & \dots & \text{covar}(W_1, W_N) \\ \text{covar}(W_2, W_1) & \text{var}(W_2) & \dots & \text{covar}(W_2, W_N) \\ \dots & \dots & \dots & \dots \\ \text{covar}(W_N, W_1) & \text{covar}(W_N, W_2) & \dots & \text{var}(W_N) \end{pmatrix} \quad (13)$$

6. Map states into observation space.

$$X_h^b = f_h(X^b) \quad (14)$$

where X_h^b is the vector of observations that would be produced if the model state was X^b

and f_h is the ‘‘observation function’’ that describes how the quantity being measured

relates to model state variables. The observation function is required because often the measured variable is not the same as a model state. For example, a model may have soil moisture or snow water equivalent as state variables and predict streamflow at a point in time as a function of these, without streamflow being a state variable. The observation function links the model predicted state to the observed variable; the model state is said to be projected onto the observation space (Labbe, 2015). For nonlinear observation functions, a “forward model” that computes the observed quantities from model predicted states is required (Clark et al., 2008). For linear observation functions, a matrix called an observation operator H (of dimension $m \times N$) would be used to project model states to observation space.

$$X_h^b = HX^b \quad (15)$$

When the quantity observed is the same as a model state variable, then the matrix H will have 1s for grid cells that have corresponding observations and 0s for those that do not have an observation. This is the case for example when SNOTEL snow water equivalent observations are assimilated into a model that has snow water equivalent as a state variable. This is illustrated in Figure 2.4 for a hypothetical watershed where there are two SNOTEL stations with observations, resulting in an observation operator H as a matrix with zeros everywhere except for the elements corresponding to the two grid cells containing the SNOTEL stations.

In this study, we separately simulated the observation points with high-resolution input data. This was in part because the terrain variables used for the UEB simulation at the 1190 m grid cell (obtained by sampling) may be different from those for the exact

location of the SNOTEL station even if the station is within the grid cell. In addition, simulating observation points separately allows incorporation of observation points that fall outside the watershed boundary but are close enough to the watershed to provide data for assimilation (Figure 2.5). The observation operator thus selects from the matrix X^b the subset of locations that are observation points. The resulting matrix X_h^b consists of the ensemble of model states for all m observation points.

$$X_h^b = \begin{pmatrix} W_{11}^b & W_{12}^b & \dots & W_{1E}^b \\ W_{21}^b & W_{22}^b & \dots & W_{2E}^b \\ \dots & \dots & \dots & \dots \\ W_{m1}^b & W_{m2}^b & \dots & W_{mE}^b \end{pmatrix} \quad (16)$$

7. Compute the cross covariance between model states and states in observation space, from their associated ensemble anomalies.

$$P_{xz} = \frac{1}{E-1} X' (X_h^b)^T \quad (17)$$

The ensemble anomaly in observation space X_h^b is computed in a similar manner as X' (ensemble anomaly of model states). When the matrix H is used this reduces to:

$$P_{xz} = \frac{1}{E-1} X' (X_h^b)^T = P_{xx} H^T \quad (18)$$

8. Compute the innovation: the difference between observation and model state in observation space.

$$d = Z_o - X_h^b \quad (19)$$

The matrix Z_o of dimensions (m x E) is obtained from the observation vector z_o by perturbing the observations with random errors sampled from a normal distribution with zero mean and observation error covariance R. The observation error covariance matrix R (m x m) provides an estimate of uncertainty in measurements. Such perturbation

of observations arises from the need to treat the observations as random variables and ensures that the update ensembles have sufficient variance (Burgers et al., 1998).

$$Z_o = z_o + \varepsilon_R, \varepsilon_R \sim N(0, R) \quad (20)$$

The R matrix is comprised of individual observation error variances. In this study, we assumed that there is no correlation between errors at the measurement locations, i.e., the random error in measurement in one SNOTEL station is independent of the random errors at other stations. Hence, the off-diagonal elements of the matrix R are zero.

9. Compute the innovation covariance P_{zz} , which is the sum of observation error covariance R and error covariance of model states in observation space X_h^b . The innovation covariance accounts for uncertainty in observation and uncertainty in process model—projected onto the observation space.

$$P_{zz} = \frac{1}{E-1} X_h' (X_h')^T + R \quad (21)$$

When the matrix H is used

$$P_{zz} = \frac{1}{E-1} X_h' (X_h')^T + R = H P_{xx} H^T + R \quad (22)$$

10. Compute the Kalman Gain K from covariance matrices P_{xz} and P_{zz} .

$$K = P_{xz} P_{zz}^{-1} \quad (23)$$

The Kalman Gain assigns proportional weights to the model predictions and observations based on their respective uncertainties (represented in terms of their error covariances). A simple way to think about this is if the state and observation vectors were one dimensional, the covariance matrices would represent covariance and variance and the gain would be the ratio of these.

11. Update the states.

$$X^a = X^b + K d \quad (24)$$

Except for step 1, generation of the spatially and temporally correlated forcing perturbations, each model grid cell executes all the steps independently of other grid cells. Effectively each model grid cell and the vector of observations become a “model” to which the EnKF approach is applied. Therefore, the dimension of the EnKF model at each grid cell is one plus the number of observation points.

Steps 10 and 11 demonstrate that the Kalman Gain and the resulting updates depend on the relative magnitude of the error covariances of the model and observations. It follows from this that a successful assimilation is only possible when there is a correlation between the model and observed variables. This is why we are not updating the state variables other than snow water equivalent (W) during assimilation, as we do not know if there is clear covariance/correlation between the observations of W and the other states. On the other hand, the same reasoning was used to justify assimilation of W measurements at an observation grid cell to simulated W at a different grid cell if there is a covariance/correlation between the simulated water equivalents at the two grid cells. Figure 2.6 shows a scatter plot of the UEB simulated SWE at two SNOTEL stations about 15 km apart demonstrating the strong correlation between the SWE at the two points during the snow accumulation season and motivating this assimilation approach to obtain the spatially distributed SWE.

The EnKF implementation for the assimilation of point SWE observations in UEB is summarized in Figure 2.7.

2.2.5 *Implementation of the Particle Filter (PF) in RDHM (SAC-SMA + rutpix7)*

The Particle Filter (PF) was implemented in RDHM to extend the data

assimilation beyond just snow (UEB) by using observed streamflow to adjust SAC-SMA and rutpix7 states. In PF, the vector of state variables representing a point in state space is referred to as a particle. Then as the model evolves and state variables change, the particles move. As stated earlier, the PF represents the probability distribution of the model states by a set of particles and associated weights. In this study, multiple realization of the SAC-SMA + rutpix7 models (X) and their associated weights (w) constitute the particles.

$$X = \{x_i\}_1^N; w = \{w_i\}_1^N \quad (25)$$

The steps for the implementation of the PF are summarized in Table 2.3. At the start of the simulation, the initial particles are generated by perturbing the initial SAC-SMA states with small errors sampled from normal distribution with zero mean. These initial particles are assumed to have uniform weights.

$$\{w_i\}_1^N = \frac{1}{N} \text{ at time step } t_k = 0 \quad (26)$$

After the SAC-SMA and rutpix7 models are run for all the particles with inputs of rain plus melt (Rmelt) output from UEB producing streamflow outputs, the weight of each particle is updated based on the likelihood function:

$$w_i \leftarrow w_i * \exp\left(\frac{-1}{2}(z - h(x_i))R^{-1}(z - h(x_i))^T\right) \quad (27)$$

Where z is observation, in this case discharge at the watershed outlet, with error covariance R . h is the observation function, in this case the RDHM model, projecting the model states onto observation space.

The weights are normalized so they sum to one.

$$w_i \leftarrow w_i / \sum_{i=1}^N w_i \quad (28)$$

The weight update assigns higher weights to those particles closer to the observation and vice versa. One consequence of this is that some particles may end up getting extremely small weights leading to a condition called “Particle degeneracy.” To mitigate this, particle resampling is employed to discard particles with negligible weights and multiply those with higher weights. The particle resampling uses the cumulative distribution of the particles, where the normalized particle weights represent the probability density (Arulampalam et al., 2002; Brown and Hwang, 2012). Resampling introduces another problem called “Particle impoverishment” where the diversity of particles is lost in that the resampled particles’ set may consist of one or very few model realizations (Liu and Gupta, 2007). Perturbation of samples or ‘sample roughening’ is introduced to generate particles that increase sample diversity (Crassidis and Junkins, 2011). In this study, sample roughening was performed with randomly generated small errors added to the SAC-SMA states.

To summarize, the Particle Filter proceeds by evolving the model states in time and updating the associated weights based on observations followed by resampling and (optional) perturbation of the particles.

The purpose of the data assimilation prior to streamflow forecasting is to arrive at the best possible set of model states at forecast time (April 1) so that the uncertainty in the forecast streamflow is reduced. The EnKF SWE assimilation attempts to get the best snow state, while the purpose of the streamflow assimilation was to select the best possible SAC-SMA and rutipix7 states. One challenge in assimilation of streamflow to

adjust soil moisture is that the observed discharge at a given time step is the result of integrated effect of the watershed processes at longer time span. Hence, it is often difficult to merely assimilate outlet discharge at a time step and expect ‘best’ trajectories of states. To deal with this, the streamflow assimilation needs to account for possible lag between soil moisture state and discharge at outlet (Noh et al., 2011). In this study, streamflow observations were available at daily frequency, and hence we computed the weights (Equation 27) at daily steps. On the other hand, we tested different frequencies (daily, every 4 days, every 10 days, weekly, bi-weekly, monthly) for the particle resampling and found that the bi-weekly resampling resulted in the best outputs and thus we used bi-weekly resampling. The challenge with this delayed resampling (not resampling at every assimilation step) is that the degeneracy mentioned earlier may get worse. The magnitude of the effect of degeneracy is measured by the ‘Effective sample size’ (Neff) (Arulampalam et al., 2002) computed as:

$$N_{eff} = 1 / \sum_{i=1}^N (w_i)^2 \quad (29)$$

Small N_{eff} signifies severe degeneracy (Arulampalam et al., 2002). In practice, resampling is carried out when N_{eff} falls below a threshold value (Moradkhani et al., 2005). In this study, the N_{eff} was computed at each step and written out for inspection, but the resampling frequency was fixed to two weeks mentioned above as it resulted in the best outputs. Note after resampling and perturbation, the particle weights are reinitialized to uniform values, $1/N$ for N particles.

2.2.6 *Evaluation and Performance Metrics*

The SWE data assimilation was run in the Green River Watershed above Warren

Bridge for the whole 2009 water year with assimilation of observed SWE every 14 days. This includes assimilation in the accumulation period prior to April 1 and the ablation and melt period after April 1. To evaluate the performance of the snow data assimilation we used “leave one out” validation. We excluded one of the SNOTEL stations from assimilation and examined how much the updated SWE at that station is improved by assimilation of observations from the other stations. This provided a way to quantify the likely improvement in estimation of SWE at unmeasured grid cells where the data assimilation is being used to quantify snow distributed across the watershed. For the Green River watershed, we carried out the assimilation using data from three stations and evaluated the outputs from the grid cell corresponding to the fourth SNOTEL station by comparing with observations at that fourth station.

The model simulated snow water equivalent at each observation location for cases with and without data assimilation, and when the observation at that station is excluded were evaluated using the following performance metrics:

1. Root Mean Square Error (RMSE) between observed and UEB SWE at SNOTEL stations.
2. Nash-Sutcliffe Efficiency (NSE) coefficient of UEB SWE as compared to observed SWE.
3. Normalized Information Contribution (NIC) from data assimilation (DA), in terms of RMSE and NSE.

The Normalized Information Contribution (NIC) from DA quantifies the extent to which the data assimilation improves (or degrades) the model simulation. It is defined as the ratio of the improvement due to data assimilation (over the simulation without data

assimilation) to the maximum possible improvement (Kumar et al., 2014; Kumar et al., 2009). NIC is defined in terms of error stats such as RMSE. We computed two values of NIC in terms of the RMSE and NSE as follows.

$$\text{NIC}_{\text{RMSE}} = (\text{RMSE}_{\text{NoDA}} - \text{RMSE}_{\text{DA}}) / \text{RMSE}_{\text{NoDA}} \quad (30)$$

$$\text{NIC}_{\text{NSE}} = (\text{NSE}_{\text{DA}} - \text{NSE}_{\text{NoDA}}) / (1 - \text{NSE}_{\text{NoDA}}) \quad (31)$$

The subscripts “DA” and “NoDA” in equations 30 and 31 refer to model with and without data assimilation respectively. In short, NIC_{RMSE} represents the reduction in RMSE divided by the maximum possible reduction, whereas NIC_{NSE} is the increase in NSE divided by the maximum possible increase. $\text{NIC} > 0$ implies improvement over no data assimilation (Kumar et al., 2014).

The above evaluation deals with the SWE assimilation and its validation at observation points, which was limited by the few observation sites available. Next, we evaluated the effect of the data assimilation on the integrated watershed response using the streamflow at the watershed outlet. In this case, the snow and streamflow data assimilation was applied from October 1 to April 1. Beyond April 1, the UEB+RDHM model was run in forecast mode and there was no data assimilation. The focus of this evaluation was on the streamflow forecast. Here the SWE assimilation frequency was also 14 days using the EnKF with all SNOTEL sites (no leave one out). Daily observations of streamflow were used to compute the Particle Filter (PF) sample weights (so the PF was run daily) with particle resampling and sample roughening done every 14 days. We evaluated ensemble streamflow forecasts for three simulation / reforecast scenarios:

1. No data assimilation
2. Assimilation of SWE in UEB using EnKF
3. Assimilation of SWE in UEB using EnKF and assimilation of streamflow with PF to update SAC-SMA and rutpix7 states.

For each scenario, the following evaluation metrics were calculated.

1. Daily streamflow error statistics for the whole water year: Root mean square error (RMSE), Nash-Sutcliffe Efficiency (NSE), Mean absolute Error (MAE), and Correlation coefficient between simulation and observation (R).
2. Daily streamflow error statistics for the forecast period (April – September): RMSE, NSE, MAE, R.
3. April – July volume error.

2.3 Results and Discussion

2.3.1 *Assimilation of SWE in UEB*

Figure 2.8 shows plots of the ensemble SWE at the Loomis Park SNOTEL station for the 2009 water year, with assimilation every 14 days continuing for the full year, i.e. both accumulation and melt periods. Plots are also shown for simulated SWE with no data assimilation and the observed SWE at this SNOTEL station. This figure is a sample output at a UEB grid cell and demonstrates the behavior of the ensemble members before and after data is assimilated. Specifically, at the date of data assimilation, the spread of ensemble members contracts towards the observations. Figures 2.9 to 2.12 show the point SWE at each SNOTEL station used in this study. The plots include model outputs with

and without data assimilation, the SNOTEL observed SWE, and simulated SWE when observations at the SNOTEL station are excluded. In all four cases, the simulation without data assimilation underestimates the snow water equivalent, when compared to observations. The simulation without data assimilation also has a temporal shift from observations in some cases. After assimilation, the modeled SWE better tracks the observed SWE as demonstrated by the improved RMSE and NSE values. All NIC values, labeled “NIC_RMSE_DAall” and “NIC_NSE_DAall” in the figures, are greater than zero indicating the data assimilation improves estimation of the model SWE. For the case where data from all stations were assimilated, the NIC in RMSE value, averaged over the four stations, was 0.41 indicating there is about 41% decrease in RMSE due to data assimilation. For the NSE criteria, the average NIC was 0.63; hence, about 63% of the possible maximum improvement in terms of NSE has been achieved through the SWE assimilation. A larger contribution in this value (improved NSE criteria) is coming from the Gunsight Pass and New Fork Lake SNOTEL stations, but the improvement in the other two stations is also significant. Overall, the data assimilation has brought the simulated peak SWE (around April 1) closer to the observed values, and this has a positive effect on melt period streamflow simulations / forecasts.

In the case of the “leave one out” validation where the data assimilation excludes observation at the particular station being evaluated, all locations except Kendall RS SNOTEL station showed improvement over no data assimilation in terms of the RMSE and NSE, “NIC_RMSE_DAExclStn” and “NIC_NSE_DAExclStn”. As can be seen from Figures 2.9 to 2.111, there is appreciable improvement in the modelled SWE from assimilation of data from the other stations, particularly during the accumulation period.

While there is a degradation in performance at all stations during the snow ablation period, Figure 2.12 shows large degradation at the Kendall RS SNOTEL station. The average NIC values for the whole water year at the Kendall RS SNOTEL are negative indicating that when looking at the year as a whole the NIC statistic does not indicate improvement. The poor performance during the melt period has overwhelmed any improvement during the accumulation period. In this ‘leave one out’ scenario, the update depends on the correlation of simulated SWE between the grid cell in question and the points where there are observations. This correlation breaks down during the melt season and is responsible for this melt season degradation in performance. This can be seen in Fig 2.12 where there is a big bump around May 1 in the Kendall RS SWE for the ‘leave one out’ scenario, when the observed snow at this station has significantly declined in Figure 2.12, but the nearby stations being used to update SWE have not declined as much. This indicates a weakness of assimilation during the ablation period.

Figure 2.13 shows the same plots as in Figure 2.12 except that for this plot the SWE assimilation was stopped on April 1. In this case, the simulated SWE traces beyond April 1 are closer to the observations for both assimilation cases (i.e., all stations included and leave-one-out). While the NIC values are still negative for the leave-one-out, their magnitude is much better than those in Figure 2.12. It is important to note here that the breakdown of correlations during the ablation period is likely to apply to the early season shallow snow conditions where intermittent accumulation and melt occurs at some locations. These early season errors possibly affect the performance of the assimilated SWE at some locations.

One of the strengths of the SNOTEL data is its high quality—compared to other

operationally available data; however, the stations are sparsely distributed. The results shown here indicate that the model SWE at a given point is improved through assimilation of SWE at SNOTEL stations away from the grid point, due to the spatial correlation of snow, at least for the accumulation period. This serves as a measure of how estimates of SWE at grid cells across the watershed where there are no observations should also be improved, and this has a positive effect on watershed scale streamflow response at the outlet, evaluated in the next section.

2.3.2 *Streamflow forecast with assimilation of SWE and Q in RDHM*

Figures 2.14 - 2.16 show the 2009 water year outputs of daily streamflow forecast for the three modeling scenarios: 1) no data assimilation (No DA); 2) assimilation of snow water equivalent using the EnKF method (EnKF SWE); 3) assimilation of both SWE and streamflow observation using the PF method (EnKF SWE & PF Q). Note that in these results, ensembles were generated using perturbed observed forcing from Oct 1 to April 1, which is the forecast time. Beyond the April 1 forecast time, the ensembles were generated using historical forcing. The data assimilations, when applicable, were run up to April 1. The figures show individual ensemble traces of streamflow and their mean (ensemble mean). Plots also include the observed streamflow hydrograph at the USGS gage at the watershed outlet. The daily streamflow error statistics for the ensemble mean are also included in the figures. In addition, to focus on the forecast period (April – September), scatter plots of forecast versus observed discharge and the corresponding error statistics are shown in Figure 2.17 for all the three scenarios. The April – July volume errors and their (ensemble) distribution are shown in Figure 2.18.

The key observations from these results are:

- The EnKF snow data assimilation results in better model states at the start of the forecast period, which is manifested in a better streamflow forecast (when compared to the forecast with no data assimilation).
- The assimilation of streamflow with PF does not add noticeable improvement to the streamflow forecast beyond that achieved by the SWE assimilation.
- There was about 94 % reduction in April – July volume error when SWE was assimilated before forecast, while incorporating the streamflow assimilation in addition to SWE assimilation resulted in April – July volume error reduction of about 85%.

The possible reasons for why the EnKF SWE & PF Q did not add any more value beyond that of the EnKF SWE may be insufficient particle size, inadequate representation of the impact of lag between soil moisture and streamflow or other undetected errors. It may also simply be that prior to April 1 there has been little snowmelt and hence little streamflow response to serve as information to update model state and inform a forecast. It is also possible, though unlikely, that the range of uncertainty due to soil moisture is narrower than that of the error distribution of the SWE ensemble and therefore the SWE assimilation compensates for the soil moisture error as well. Pinpointing the exact reason requires further investigation.

The streamflow ensemble mean for all the three cases does not replicate the shape of the hydrograph at the peaks. While the observations show two peaks during the forecast period, the ensemble mean has a single smooth peak. This is to be expected due

to the averaging of multiple ensemble members that were produced using historical weather inputs from all years with data available. Some of the individual ensemble members appear to have multiple peaks. This is due to the different weather patterns in different years. This indicates that, while the average total volume of discharge at the gage is more accurately captured, as shown in the April - July volume error in Figure 2.18, the ensemble streamflow generated from the historical forcing (assumed as representative of the possible distribution of future weather conditions) does not capture the actual shape of the hydrograph. To forecast the actual shape of the hydrograph in any one year requires a forecast of the specific weather pattern for that year. One implication of this result is that this data assimilation is useful for water supply forecasting, but less useful for flood forecasting unless other factors such as input energy are also well quantified.

Finally, to see if the results hold for years other than the 2009 water year, the April to July volume errors were also examined for four additional years 2005 - 2008 (Figure 2.19). Results comparable to 2009 were obtained for 2005, 2006, and 2008. In these years, the assimilation of SWE generally improves the forecasted water volume while it misses the shape of the hydrographs at the peaks (not shown), similar to results for the 2009 water year. And, the PF assimilation of discharge does not add value beyond that achieved through SWE assimilation. In general, the simulation with no data assimilation has negative volume error, suggesting that without data assimilation, forecast streamflow volumes are likely to be underestimated. The SWE assimilation reduces the volume error while the discharge assimilation slightly increases the volume error. For the year 2007, however, the volume error is smallest for the case with no data assimilation. In

this case, the discharge assimilation performs better than the forecast with only SWE assimilation. By looking into the observed streamflow for all the 22 historical years (1989 – 2010), the streamflow for 2007 has the second lowest peak value, next to water year 1992, making it a dry year outlier. This points to the need for further investigation with specific focus on low flow years.

2.4 Summary and Conclusions

In this study, we evaluated assimilation of snow water equivalent (SWE) data from SNOTEL sensors into the Utah Energy Balance (UEB) snowmelt model using the Ensemble Kalman Filter (EnKF). The energy balance snowmelt model was then coupled to a distributed hydrologic model, RDHM, for streamflow forecasting. The streamflow forecasts were conditioned on updated snow and soil moisture states from assimilation of snow and streamflow data in RDHM. We used the Particle Filter (PF) with observed streamflow to update SAC-SMA and rutpix7 states in RDHM. For the snow data assimilation, the model ensemble states were generated using forcing perturbations that account for spatial correlation through an exponential function decreasing with distance. In the PF implementation, a set of SAC-SMA and rutpix7 states and their associated weights constitute the particles that evolved through time.

Results from the SWE assimilation showed that taking advantage of the covariance between SWE at different grid cells, an observation at one point can be used to update SWE at a grid cell away from the observation site without needing interpolation of observations. This suggested the potential for assimilation of SWE data from sparse SNOTEL stations to update SWE over the whole watershed, which we later evaluated

through streamflow forecasts. The improvement in model SWE from assimilation was demonstrated through the reduced RMSE and increased NSE values for three of the four SNOTEL stations, expressed in NIC (Normalized Information Contribution from data assimilation).

The snow assimilation performance was much better during accumulation than ablation. This may be due to the spatial correlations that resulted in data at one point having predictive capability at another point during accumulation diminishing during the energy driven melt process where melt timing is not synchronous. The spatial variability of snowmelt due to elevation and the effect of slope and aspect on radiation is not spatially correlated in the same way as snow accumulation, and hence not readily captured by the data assimilation method.

The ensemble model setup in this study was premised on the assumption that the major source of uncertainty in the model comes from the weather forcing. The ensemble forcing perturbation accounts for the spatial and temporal as well as inter-variable correlations of errors in forcing such as precipitation. The parameters for generating ensemble forcing, such as the variance in precipitation and the spatial and temporal correlation lengths, were input to the model. Further study on the uncertainty in each weather forcing variable and its statistical characteristics such as the error standard deviation, for example by analyzing forcing data from multiple sources, could be beneficial. Future study for accurate estimation of the spatial correlation length should look into its relation to the distance between the SNOTEL stations in a given watershed and other factors such as the effects of topography on precipitation shadows, ridges, and wind shadows that in turn affect the spatial correlation of snow.

Another outcome of this study was that the ensemble streamflow forecasts with assimilation of SWE provided marked improvement over forecasts with no data assimilation. On the other hand, the assimilation of streamflow using the Particle Filter to update the SAC-SMA and rutpix7 states did not provide improvement beyond that achieved through the snow data assimilation. The possible reasons for the failure of the streamflow assimilation to add value may be insufficient particle numbers, inadequate representation of the impact of lag between soil moisture and streamflow or due to narrow soil moisture error range that the snow ensemble distribution covers. It may also be due to the dominant predictor for streamflow being the amount of snow, and prior to the forecast date (April 1) there is generally limited snowmelt with the result that streamflow has not yet responded to the amount of snow present and thus observations do not incorporate any information on the amount of snow present.

Results also show the ensemble forecast mean provides a good estimate of the mean volume of water for water supply forecast, but its smooth shape misses the peaks of the hydrographs, implying that the forecasts may not be capable of capturing the timing and magnitude of flood hydrograph.

Given the high computational demand by the large number of model realizations required for the Particle Filter, and given that it did not result in an improvement in the forecasted streamflow, it is not suggested for operational forecasts at this stage, without further investigation. However, the coupled UEB+RDHM model introduced in this work did result in improvements to forecast streamflow volumes and we suggest does merit consideration for operational use.

References

- Abaza, M., Anctil, F., Fortin, V., Turcotte, R., 2014. Sequential streamflow assimilation for short-term hydrological ensemble forecasting. *Journal of Hydrology* 519 2692-2706.
- Anderson, E., 1976. A point energy and mass balance model of a snow cover, Technical Report NWS 19. Office of Hydrology, National Weather Service: Silver Spring, MD
- Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T., 2002. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on signal processing* 50(2) 174-188.
- Bartelt, P., Lehning, M., 2002. A physical SNOWPACK model for the Swiss avalanche warning: Part I: numerical model. *Cold Regions Science and Technology* 35(3) 123-145.
- Beven, K., Cloke, H., Pappenberger, F., Lamb, R., Hunter, N., 2015. Hyperresolution information and hyperresolution ignorance in modelling the hydrology of the land surface. *Science China Earth Sciences* 58(1) 25-35.
- Biederman, J.A., Somor, A.J., Harpold, A.A., Gutmann, E.D., Breshears, D.D., Troch, P.A., Gochis, D.J., Scott, R.L., Meddens, A.J., Brooks, P.D., 2015. Recent tree die - off has little effect on streamflow in contrast to expected increases from historical studies. *Water Resources Research* 51(12) 9775-9789.
- Brown, R.G., Hwang, P.Y.C., 2012. *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises*, Fourth ed. John Wiley & Sons, Inc.
- Broxton, P., Harpold, A., Biederman, J., Troch, P.A., Molotch, N., Brooks, P.D., 2015. Quantifying the effects of vegetation structure on snow accumulation and ablation in mixed - conifer forests. *Ecohydrology* 8(6) 1073-1094.
- Burgers, G., Jan van Leeuwen, P., Evensen, G., 1998. Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review* 126(6) 1719-1724.
- Burnash, R., Singh, V., 1995. The NWS river forecast system-Catchment modeling. *Computer models of watershed hydrology*. 311-366.
- Burnash, R.J., Ferral, R.L., McGuire, R.A., 1973. A generalized streamflow simulation system, conceptual modeling for digital computers.
- Clark, M.P., Hendrikx, J., Slater, A.G., Kavetski, D., Anderson, B., Cullen, N.J., Kerr, T., Örn Hreinsson, E., Woods, R.A., 2011. Representing spatial variability of snow water equivalent in hydrologic and land-surface models: A review. *Water Resources Research* 47(7) n/a-n/a.
- Clark, M.P., Rupp, D.E., Woods, R.A., Zheng, X., Ibbitt, R.P., Slater, A.G., Schmidt, J., Uddstrom, M.J., 2008. Hydrological data assimilation with the ensemble Kalman filter: Use of streamflow observations to update states in a distributed hydrological model. *Advances in Water Resources* 31(10) 1309-1324.
- Clark, M.P., Slater, A.G., Barrett, A.P., Hay, L.E., McCabe, G.J., Rajagopalan, B., Leavesley, G.H., 2006. Assimilation of snow covered area information into hydrologic and land-surface models. *Advances in Water Resources* 29(8) 1209-1221.

- Cloke, H., Pappenberger, F., 2009. Ensemble flood forecasting: a review. *Journal of Hydrology* 375(3) 613-626.
- Crassidis, J.L., Junkins, J.L., 2011. *Optimal estimation of dynamic systems*. Chapman and Hall/CRC.
- Daly, C., Halbleib, M., Smith, J.I., Gibson, W.P., Doggett, M.K., Taylor, G.H., Curtis, J., Pasteris, P.P., 2008. Physiographically sensitive mapping of climatological temperature and precipitation across the conterminous United States. *International Journal of Climatology* 28(15) 2031-2064.
- Daly, C., Neilson, R.P., Phillips, D.L., 1994. A statistical-topographic model for mapping climatological precipitation over mountainous terrain. *Journal of Applied Meteorology* 33(2) 140-158.
- Day, G.N., 1985. Extended streamflow forecasting using NWSRFS. *Journal of water resources planning and management* 111(2) 157-170.
- Déry, S.J., Yau, M., 2002. Large - scale mass balance effects of blowing snow and surface sublimation. *Journal of Geophysical Research: Atmospheres* (1984–2012) 107(D23) ACL 8-1-ACL 8-17.
- Drécourt, J.-P., 2003. Kalman filtering in hydrological modeling. Hørsholm, Denmark, DAIHM.
- Evensen, G., 2003. The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics* 53(4) 343-367.
- Fowler, H., Blenkinsop, S., Tebaldi, C., 2007. Linking climate change modelling to impacts studies: recent advances in downscaling techniques for hydrological modelling. *International Journal of Climatology* 27(12) 1547-1578.
- Franz, K.J., Hartmann, H.C., Sorooshian, S., Bales, R., 2003. Verification of National Weather Service ensemble streamflow predictions for water supply forecasting in the Colorado River basin. *Journal of Hydrometeorology* 4(6) 1105-1118.
- Franz, K.J., Hogue, T.S., Barik, M., He, M., 2014. Assessment of SWE data assimilation for ensemble streamflow predictions. *Journal of Hydrology* 519 2737-2746.
- Franz, K.J., Hogue, T.S., Sorooshian, S., 2008. Operational snow modeling: Addressing the challenges of an energy balance model for National Weather Service forecasts. *Journal of Hydrology* 360(1) 48-66.
- Geddes, C.A., Brown, D.G., Fagre, D.B., 2005. Topography and vegetation as predictors of snow water equivalent across the alpine treeline ecotone at Lee Ridge, Glacier National Park, Montana, USA. *Arctic, Antarctic, and Alpine Research* 37(2) 197-205.
- Grewal, M.S., Andrews, A.P., 2015. *Kalman Filtering Theory and Practice Using MATLAB*. John Wiley & Sons, Inc., Hoboken, New Jersey.
- Homer, C.G., Dewitz, J.A., Yang, L., Jin, S., Danielson, P., Xian, G., Coulston, J., Herold, N.D., Wickham, J., Megown, K., 2015. Completion of the 2011 National Land Cover Database for the conterminous United States-Representing a decade of land cover change information. *Photogrammetric Engineering and Remote Sensing* 81(5) 345-354.

- Jin, J., Gao, X., Sorooshian, S., Yang, Z.-L., Bales, R., Dickinson, R.E., Sun, S.-F., Wu, G.-X., 1999. One-dimensional snow water and energy balance model for vegetated surfaces. *Hydrological Processes* 13(1415) 2467-2482.
- Jordan, R., 1991. A one-dimensional temperature model for a snow cover: Technical documentation for SNTherm. 89. DTIC Document.
- Koren, V., Reed, S., Smith, M., Zhang, Z., Seo, D.-J., 2004. Hydrology laboratory research modeling system (HL-RMS) of the US national weather service. *Journal of Hydrology* 291(3) 297-318.
- Kumar, S.V., Peters-Lidard, C.D., Mocko, D., Reichle, R., Liu, Y., Arsenault, K.R., Xia, Y., Ek, M., Riggs, G., Livneh, B., 2014. Assimilation of remotely sensed soil moisture and snow depth retrievals for drought estimation. *Journal of Hydrometeorology* 15(6) 2446-2469.
- Kumar, S.V., Reichle, R.H., Koster, R.D., Crow, W.T., Peters-Lidard, C.D., 2009. Role of subsurface physics in the assimilation of surface soil moisture observations. *Journal of Hydrometeorology* 10(6) 1534-1547.
- Kumar, S.V., Reichle, R.H., Peters-Lidard, C.D., Koster, R.D., Zhan, X., Crow, W.T., Eylander, J.B., Houser, P.R., 2008. A land surface data assimilation framework using the land information system: Description and applications. *Advances in Water Resources* 31(11) 1419-1432.
- Labbe, R.R., 2015. Kalman and Bayesian Filters in Python.
- Lahoz, W., Khattatov, B., Menard, R., 2010. Data assimilation: making sense of observations. Springer Science & Business Media.
- Lehning, M., Bartelt, P., Brown, B., Fierz, C., Satyawali, P., 2002. A physical SNOWPACK model for the Swiss avalanche warning: Part II. Snow microstructure. *Cold Regions Science and Technology* 35(3) 147-167.
- Liston, G.E., Sturm, M., 1998. A snow-transport model for complex terrain. *Journal of Glaciology* 44(148) 498-516.
- Liu, Y., Gupta, H.V., 2007. Uncertainty in hydrologic modeling: Toward an integrated data assimilation framework. *Water Resources Research* 43(7).
- Liu, Y., Weerts, A., Clark, M., Hendricks Franssen, H., Kumar, S., Moradkhani, H., Seo, D., Schwanenberg, D., Smith, P., Van Dijk, A., 2012. Advancing data assimilation in operational hydrologic forecasting: progresses, challenges, and emerging opportunities. *Hydrology and Earth System Sciences*, 16 (10), 2012.
- Luce, C.H., 2000. Scale influences on the representation of snowpack processes. PhD Dissertation, Utah State University.
- Luce, C.H., Tarboton, D.G., 2010. Evaluation of alternative formulae for calculation of surface temperature in snowmelt models using frequency analysis of temperature observations. *Hydrology and Earth System Sciences* 14(3) 535-543.
- Mahat, V., Tarboton, D.G., 2012. Canopy radiation transmission for an energy balance snowmelt model. *Water Resources Research* 48(1) W01534.
- Mahat, V., Tarboton, D.G., 2014. Representation of canopy snow interception, unloading and melt in a parsimonious snowmelt model. *Hydrological Processes* 28(26) 6320-6336.

- Mahat, V., Tarboton, D.G., Molotch, N.P., 2013. Testing above- and below-canopy representations of turbulent fluxes in an energy balance snowmelt model. *Water Resources Research* 49(2) 1107-1122.
- Mitchell, K.E., Lohmann, D., Houser, P.R., Wood, E.F., Schaake, J.C., Robock, A., Cosgrove, B.A., Sheffield, J., Duan, Q., Luo, L., Higgins, R.W., Pinker, R.T., Tarpley, J.D., Lettenmaier, D.P., Marshall, C.H., Entin, J.K., Pan, M., Shi, W., Koren, V., Meng, J., Ramsay, B.H., Bailey, A.A., 2004. The multi-institution North American Land Data Assimilation System (NLDAS): Utilizing multiple GCIP products and partners in a continental distributed hydrological modeling system. *Journal of Geophysical Research: Atmospheres* 109(D7) D07S90.
- Moradkhani, H., Hsu, K.L., Gupta, H., Sorooshian, S., 2005. Uncertainty assessment of hydrologic model states and parameters: Sequential data assimilation using the particle filter. *Water Resources Research* 41(5).
- Mott, R., Schirmer, M., Bavay, M., Grünwald, T., Lehning, M., 2010. Understanding snow-transport processes shaping the mountain snow-cover. *The Cryosphere* 4(4) 545-559.
- NOAA's National Weather Service, H.D.M.R., 2008. The HL Research Distributed Hydrologic Model (HL-RDHM) Developer's Manual v. 2.0.
- Nichols, N.K., 2010. Mathematical concepts of data assimilation, In: Lahoz, W., Khattatov, B., Ménard, R. (Eds.), *Data Assimilation: making sense of observations*. Springer, pp. 13-39.
- Noh, S., Tachikawa, Y., Shiiba, M., Kim, S., 2011. Applying sequential Monte Carlo methods into a distributed hydrologic model: lagged particle filtering approach with regularization. *Hydrology and Earth System Sciences* 15(10) 3237.
- Pathiraja, S., Marshall, L., Sharma, A., Moradkhani, H., 2016. Hydrologic modeling in dynamic catchments: A data assimilation approach. *Water Resources Research* 52(5) 3350-3372.
- Pomeroy, J., Essery, R., 1999. Turbulent fluxes during blowing snow: field tests of model sublimation predictions. *Hydrological Processes* 13(18) 2963-2975.
- Rakovec, O., Weerts, A., Hazenberg, P., Torfs, P., Uijlenhoet, R., 2012. State updating of a distributed hydrological model with Ensemble Kalman Filtering: effects of updating frequency and observation network density on forecast accuracy. *Hydrology and Earth System Sciences* 16(9) 3435-3449.
- Reed, S.M., Maidment, D.R., 1999. Coordinate transformations for using NEXRAD data in GIS-based hydrologic modeling. *Journal of Hydrologic Engineering* 4(2) 174-182.
- Reichle, R.H., 2008. Data assimilation methods in the Earth sciences. *Advances in Water Resources* 31(11) 1411-1418.
- Reichle, R.H., McLaughlin, D.B., Entekhabi, D., 2002. Hydrologic data assimilation with the ensemble Kalman filter. *Monthly Weather Review* 130(1) 103-114.
- Semenova, O., Beven, K., 2015. Barriers to progress in distributed hydrological modelling. *Hydrological Processes* 29(8) 2074-2078.
- Sen Gupta, A., Tarboton, D.G., 2016. A tool for downscaling weather data from large-grid reanalysis products to finer spatial scales for distributed hydrological applications. *Environmental Modelling & Software* 84 50-69.

- Slater, A.G., Clark, M.P., 2006. Snow data assimilation via an ensemble Kalman filter. *Journal of Hydrometeorology* 7(3) 478-493.
- Su, H., Yang, Z.L., Niu, G.Y., Dickinson, R.E., 2008. Enhancing the estimation of continental - scale snow water equivalent by assimilating MODIS snow cover with the ensemble Kalman filter. *Journal of Geophysical Research: Atmospheres* 113(D8).
- Tarboton, D.G., Chowdhury, T.G., Jackson, T.H., 1995. A Spatially Distributed Energy Balance Snowmelt Model, In: K. A. Tonnessen, M.W.W.a.M.T. (Ed.), *Biogeochemistry of Seasonally Snow-Covered Catchments (Proceedings of a Boulder Symposium)*. IAHS, pp. 141-155.
- Wever, N., Fierz, C., Mitterer, C., Hirashima, H., Lehning, M., 2014. Solving Richards Equation for snow improves snowpack meltwater runoff estimations in detailed multi-layer snowpack model. *The Cryosphere* 8(1) 257-274.
- Wood, A.W., Lettenmaier, D.P., 2008. An ensemble approach for attribution of hydrologic prediction uncertainty. *Geophysical Research Letters* 35(14).
- Xue, Y., Forman, B.A., Reichle, R.H., 2018. Estimating snow mass in North America through assimilation of AMSR - E brightness temperature observations using the Catchment land surface model and support vector machines. *Water Resources Research*.
- You, J., Tarboton, D., Luce, C., 2014. Modeling the snow surface temperature with a one-layer energy balance snowmelt model. *Hydrology and Earth System Sciences* 18(12) 5061-5076.
- Zhou, Y., McLaughlin, D., Entekhabi, D., 2006. Assessing the performance of the ensemble Kalman filter for land surface data assimilation. *Monthly Weather Review* 134(8) 2128-2142.

Table 2.1 SNOTEL stations with SWE data used in the study

Watershed	SNOTEL Station	Elevation (m)	Latitude	Longitude
Green River above	Kendall RS	2359	43.25	-110.02
Warren Bridge	Gunsight Pass	2993	43.38	-109.88
	Loomis Park	2512	43.17	-110.13
	New Fork Lake	2542	43.12	-109.95

Table 2.2 Parameters of forcing perturbations for UEB ensemble generation

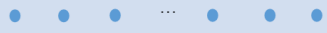
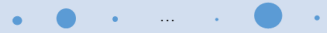
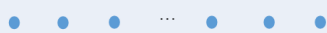
Forcing variable	Forcing perturbation error type	Error standard deviation	Correlation between variables					
			Ta	P	Q _{si}	Q _{li}	V	RH
Temperature (Ta)	Additive	1.2 °C	1					
Precipitation (P)	Multiplicative*	0.05*		1	-0.8	0.5		
Short-wave radiation (Q _{si})				-0.8	1	-0.5		
Long-wave radiation (Q _{li})				0.5	-0.5	1		
Wind speed (V)							1	
Relative humidity (RH)								1

Spatial correlation length = 16 HRAP ~ 76 Km

Temporal decorrelation length = 6 hrs.

* Note: for all forcing except temperature the perturbation type is “Multiplicative” with error standard deviation of 0.05.

Table 2.3 Steps for Particle Filter in SAC-SMA + rutpix7

	DA Step	Illustration*	Description
1	Initialize states and weights	$X_1 \quad X_2 \quad X_3 \quad \dots \quad X_{N-2} \quad X_{N-1} \quad X_N$ $1/N \quad 1/N \quad 1/N \quad \dots \quad 1/N \quad 1/N \quad 1/N$ 	At the start of simulation all states have equal weights.
2	Run model	$X_i \leftarrow \text{RDHM}(X_i, \text{Rmelt})$	Run SAC-SMA and rutpix7 models with inputs of rain + melt from UEB to advance in time for each particle.
3	Update weights	$w_i \leftarrow w_i * \exp\left(\frac{-1}{2}(z - h(x_i))R^{-1}(z - h(x_i))^T\right)$ Where z is observation with error covariance R $w_i \leftarrow \frac{w_i}{\sum_{i=1}^N w_i}$ $X_1 \quad X_2 \quad X_3 \quad \dots \quad X_{N-2} \quad X_{N-1} \quad X_N$ 	At assimilation step of the PF, the particle weights are updated based on their proximity to the observed streamflow at outlet. Weights are then normalized to make all weights sum to 1.
4	Resample states	$X_1 \quad X_2 \quad X_2 \quad \dots \quad X_{N-1} \quad X_{N-1} \quad X_{N-1}$ $1/N \quad 1/N \quad 1/N \quad \dots \quad 1/N \quad 1/N \quad 1/N$ 	Weights are used as probability to resample particles. High weight particles are sampled more frequently (repeated) while low weight particles are sampled less frequently, and those with too little weight are discarded to avoid particle degeneracy.
5	Perturb particles	Add randomly generated small error to SAC-SMA states	Randomly generated small error is added to each resampled particle to reduce particle impoverishment, i.e., loss of diversity and particles converging to common values.

* Sizes of the circles in the third column illustrate relative weights of particles—not to scale

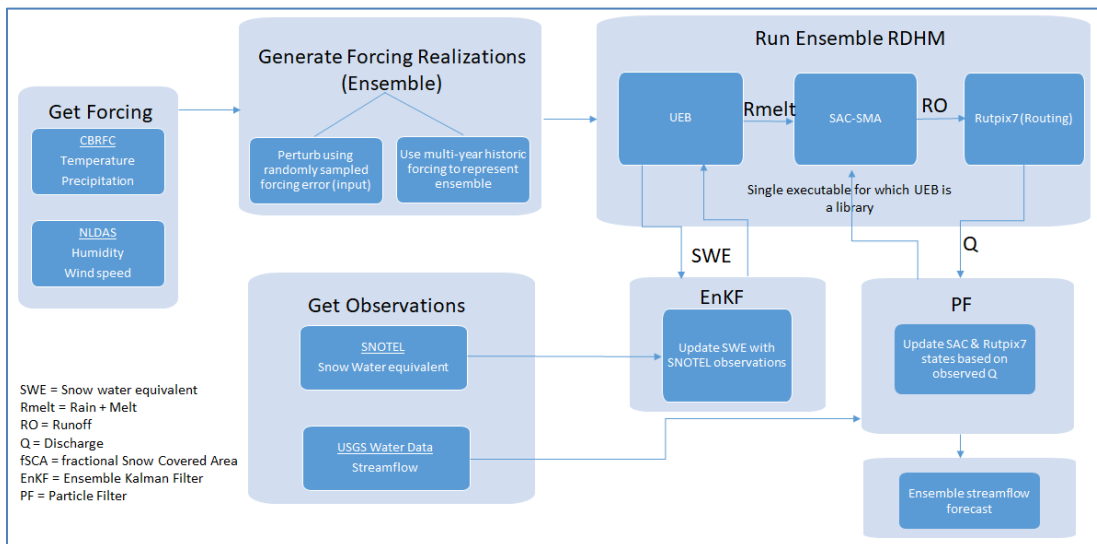


Figure 2.1 Ensemble Streamflow Forecasting Scheme with UEB as RDHM component.

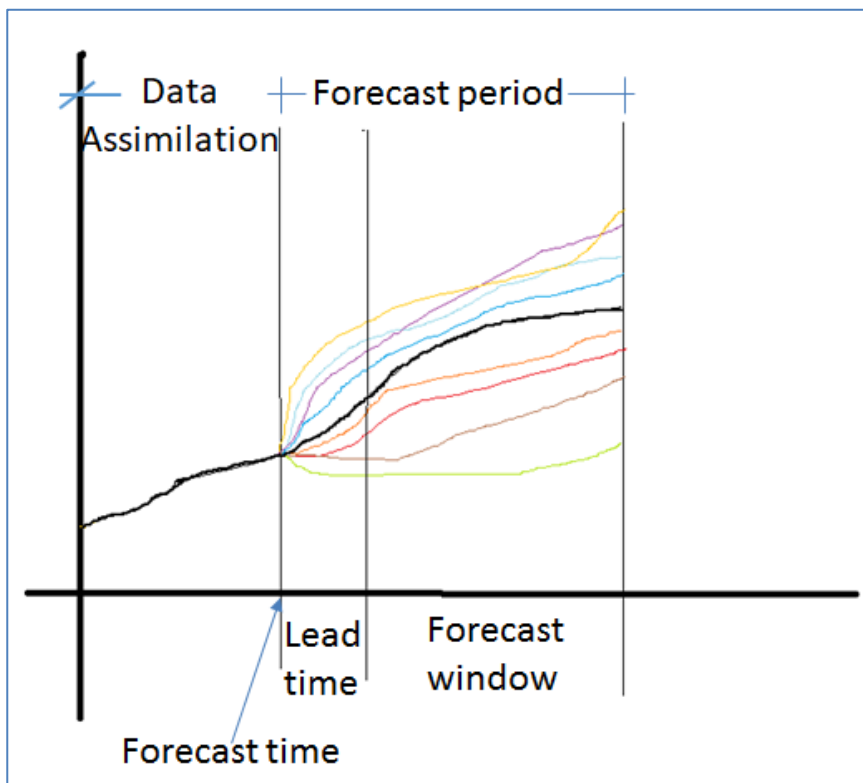


Figure 2.2 Temporal organization of Ensemble Streamflow Forecast Procedure. Adapted from (Franz et al., 2003).

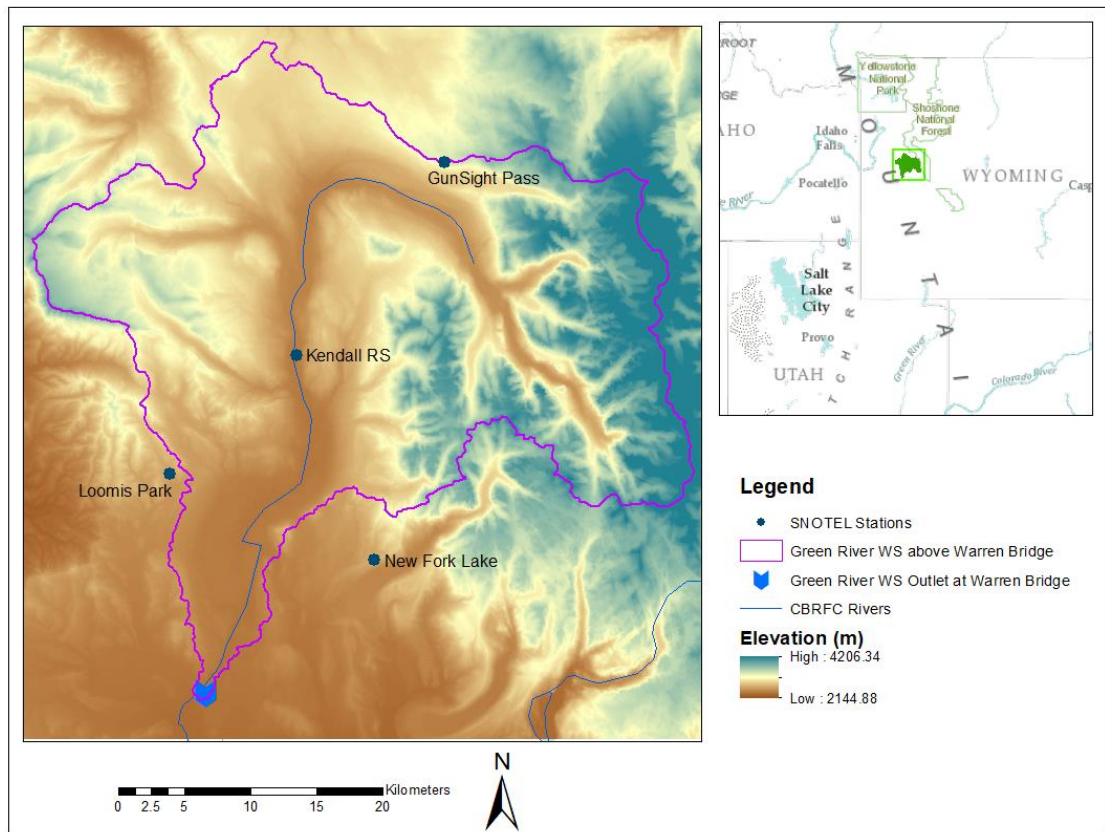


Figure 2.3 Study watershed: Green River above Warren Bridge.

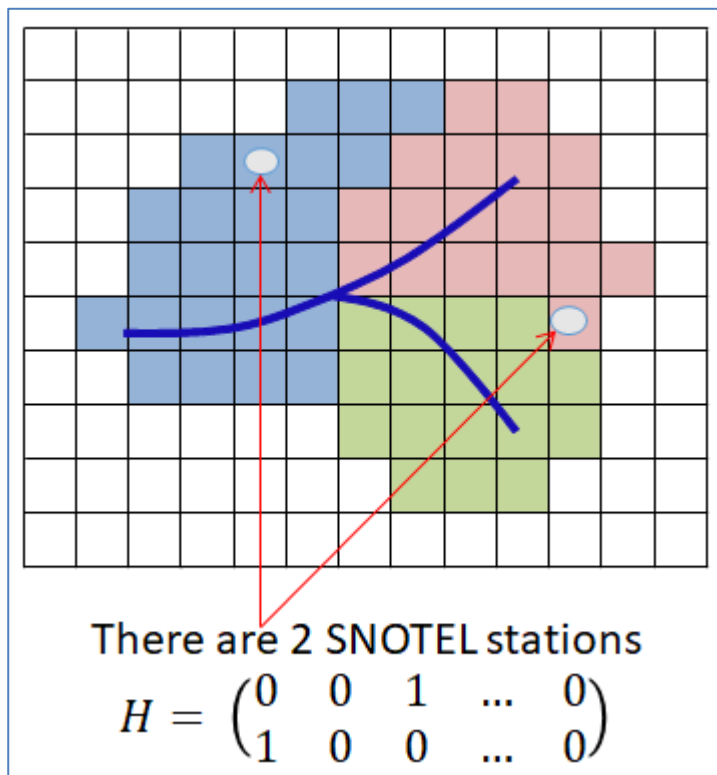


Figure 2.4 Illustration of the observation operator H for a hypothetical watershed with 59 grid cells (filled grid cells with different colors representing sub-watersheds) and 2 grids with observations. The observation operator is a 2 by 59 matrix where all but the two elements corresponding to the grid cells with observations have zero value.

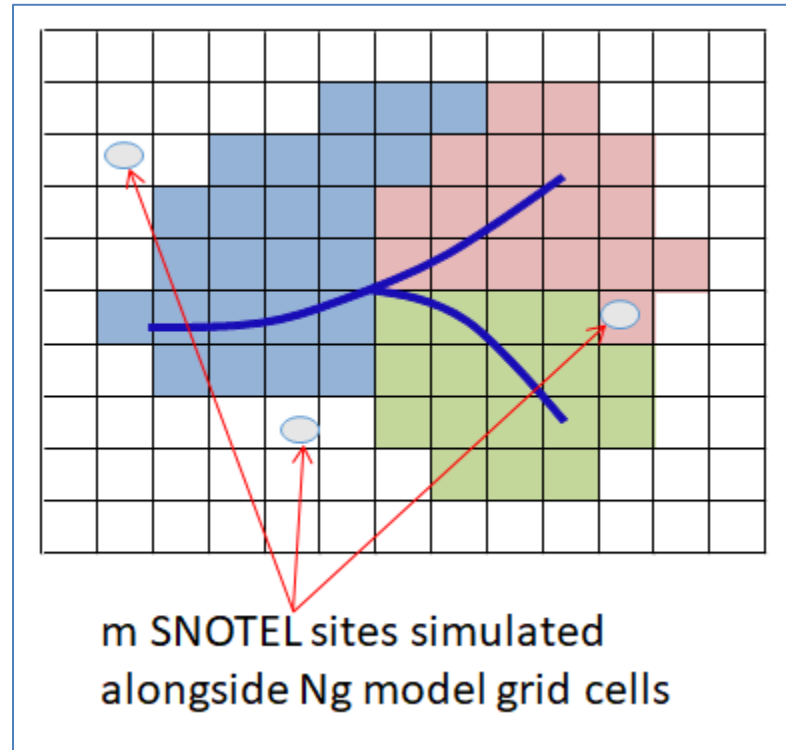


Figure 2.5 Illustration of a hypothetical watershed with three observation stations handled separately from the 59 model grid cells (filled grid cells with different colors represent sub-watersheds). In this case, there is no need for special observation operator; the state matrix in observation space consists of the ensemble of model states for the three observation points.

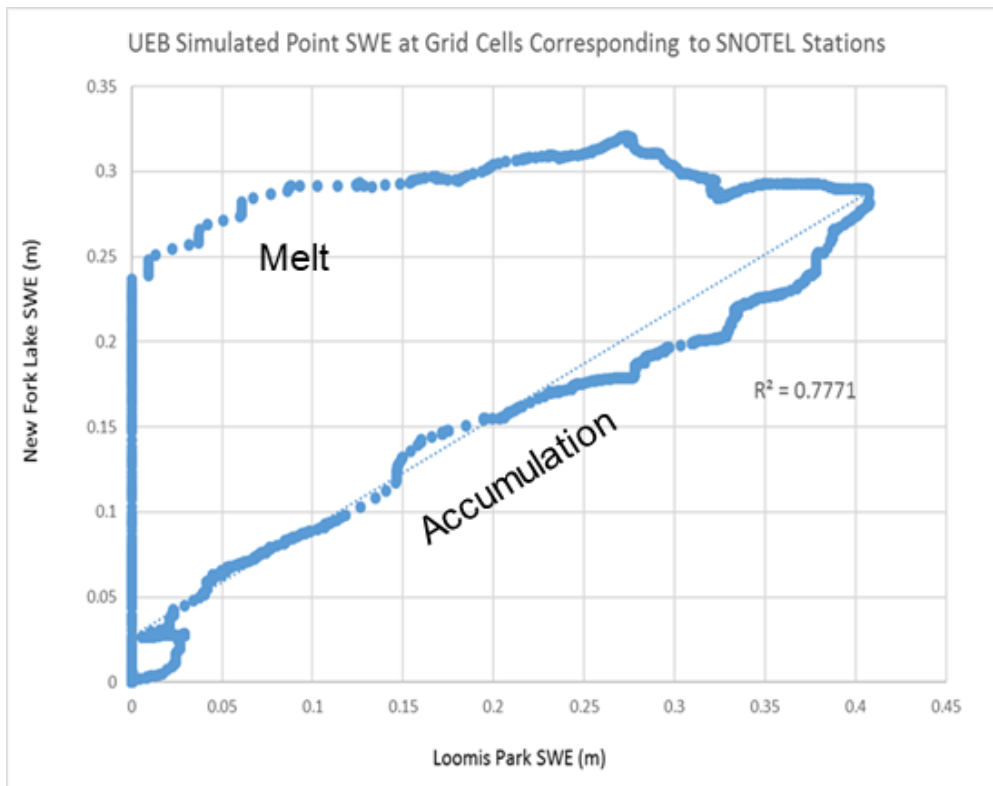


Figure 2.6 Correlation between UEB simulated SWE at two SNOTEL sites.

Ensemble Kalman Filter (EnKF) with point observations simulated at high resolution to assimilate SNOTEL SWE in UEB

- Forcing perturbation to generate UEB ensemble, size E.

Temperature

$$U_k = u_k + \theta_u, \theta_u = N(0, S_u)$$

All other forcing

$$U_k = u_k * \theta_u, \theta_u = N(1, S_u)$$

Spatially correlated forcing multipliers generate UEB ensemble forcing
- Run UEB to advance states into next time step (get background state).

Background state
(size $[Ng+m] \times E$)

$$X_{k+1}^b = M_{k,k+1}(X_k^a, U_k)$$

Model in observation space

$$X_h = f_h(X^b)$$

$$X_h = \{x_i^b\}_{i=1,m}$$

$f_h =$ (Non-linear) observation function

m SNOTEL sites simulated alongside Ng model grid cells
- EnKF to update UEB SWE states with observations

Model error covariance

$$P_{xx} = \frac{1}{E-1} X'(X')^T$$

Innovation (Observation - model difference)

$$d = Z_o - X_h$$

Model State - Observation error cross-covariance

$$P_{xz} = \frac{1}{E-1} X'(X_h')^T = P_{xx} H^T$$

Innovation covariance

$$P_{zz} = \frac{1}{E-1} X_h'(X_h')^T + R = H P_{xx} H^T + R$$

Kalman Gain

$$K = P_{xz} P_{zz}^{-1}$$

Updated State

$$X^a = X^b + K d$$

$R =$ Observation error covariance
 $H =$ (Linear) observation operator
 $X', X_h' =$ Ensemble anomaly $= X - \text{mean}(X)$

Figure 2.7 Summary of Ensemble Kalman Filter steps in UEB.

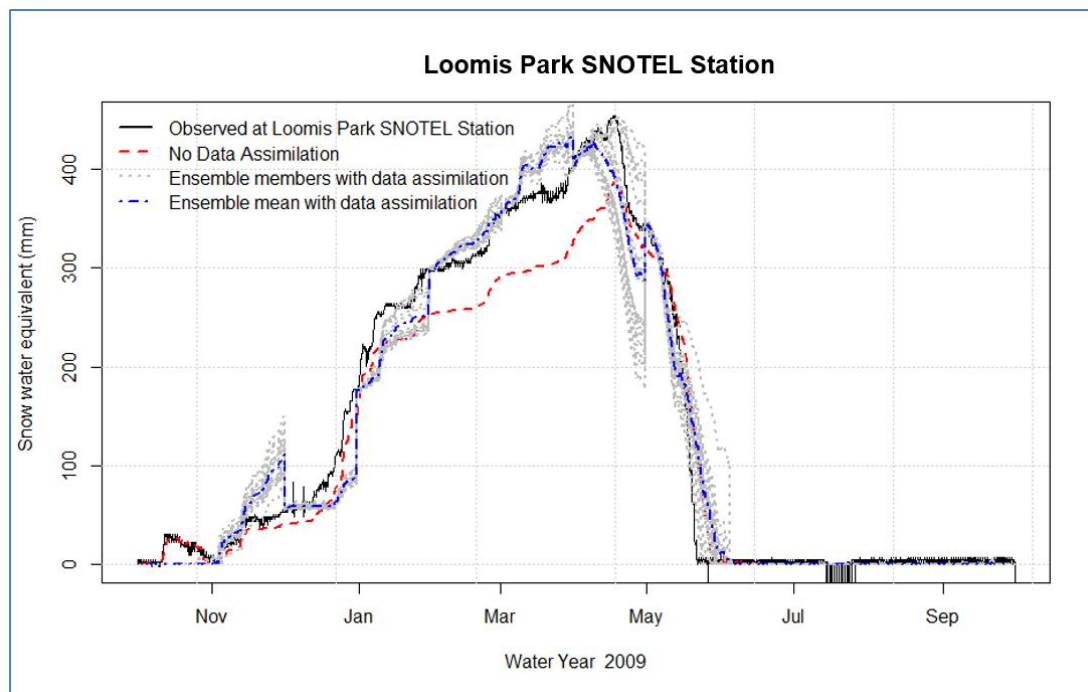


Figure 2.8 Snow water equivalent (SWE) at Loomis Park SNOTEL Station in Green River Watershed. UEB simulated SWE without data assimilation, ensembles with the EnKF assimilation of SWE and their mean, and observed SWE at this SNOTEL station are shown. At the dates of each data assimilation the spread of ensemble members contract towards the observations.

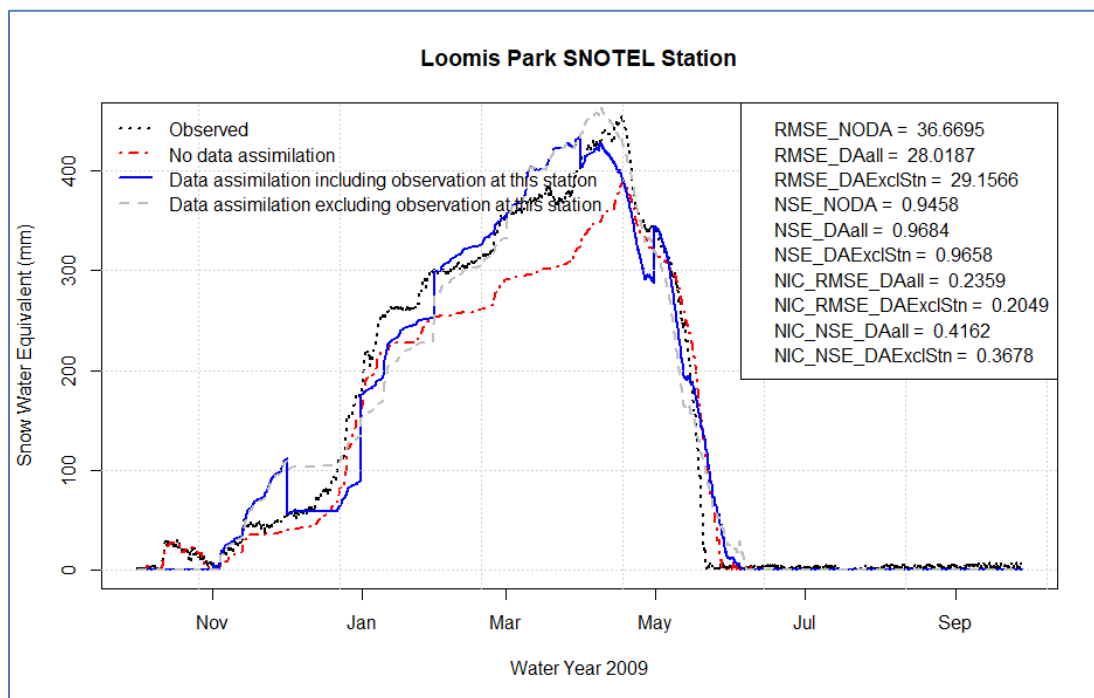


Figure 2.9 Snow water equivalent (SWE) at Loomis Park SNOTEL Station in Green River Watershed. The labels for the error statistics, at the top-right, refer to RMSE with no data assimilation (RMSE_NODA), assimilation of observations from all four stations (RMSE_Daall), and assimilation excluding data at this site (RMSE_DaExclStn). Similar terminology applies to NSE and NIC. The data assimilation outputs are from the EnKF assimilation of SWE and the plots shown are the ensemble mean for each case.

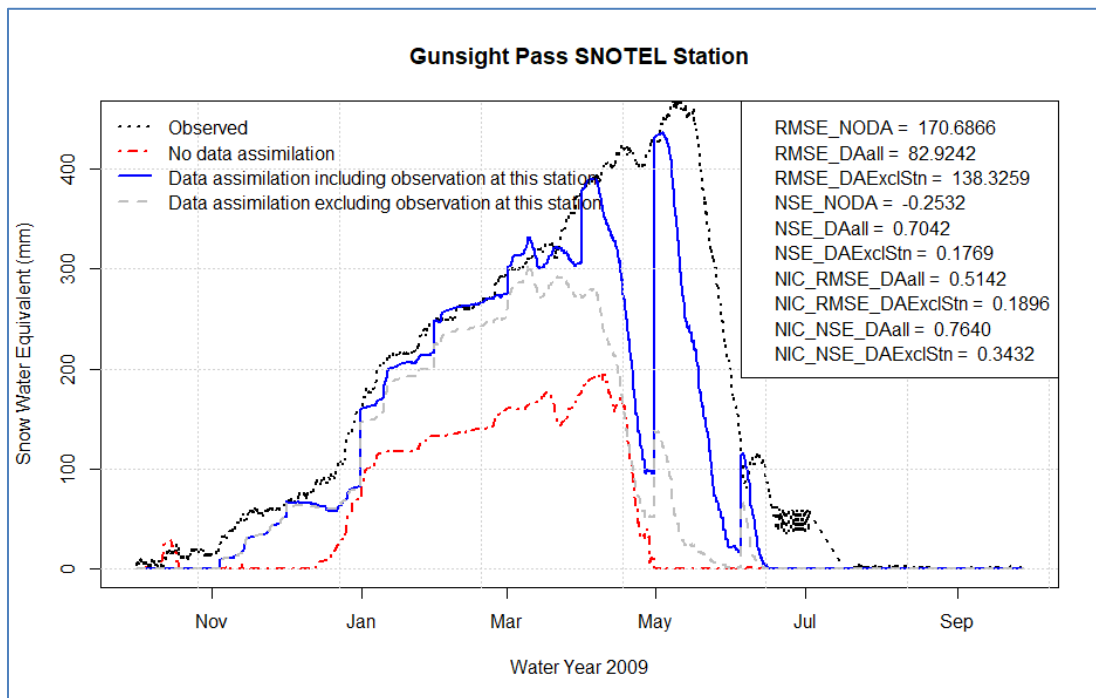


Figure 2.10 Same as Figure 2.9 but for the Gunsight Pass SNOTEL station.

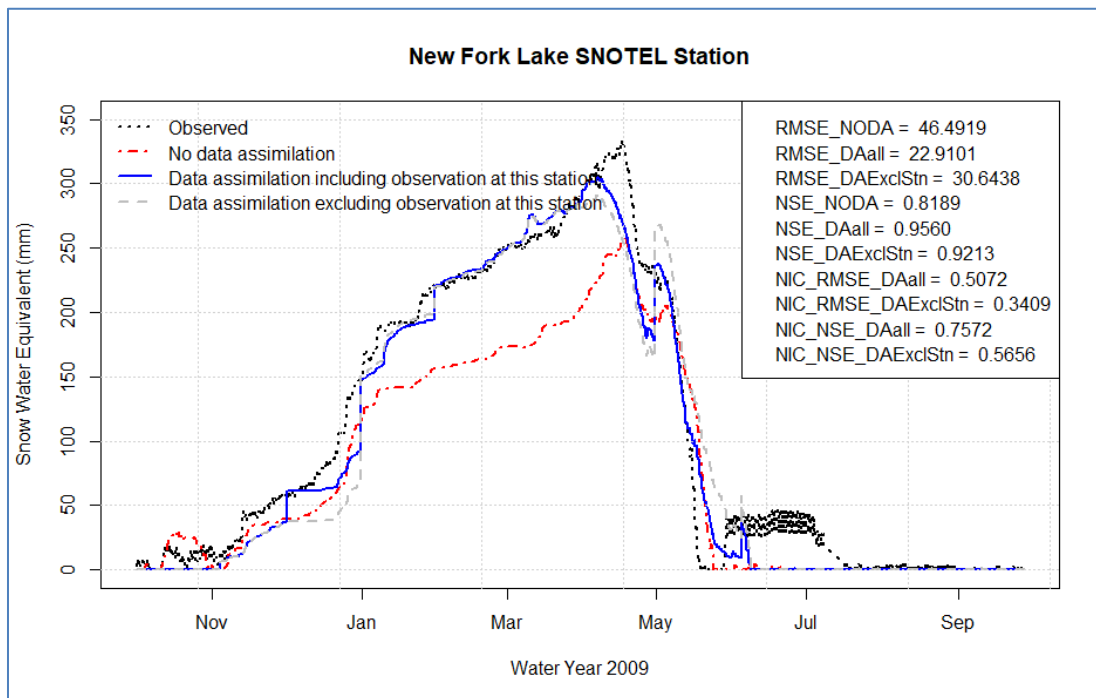


Figure 2.11 Same as Figure 2.9 but for the New Fork Lake SNOTEL station.

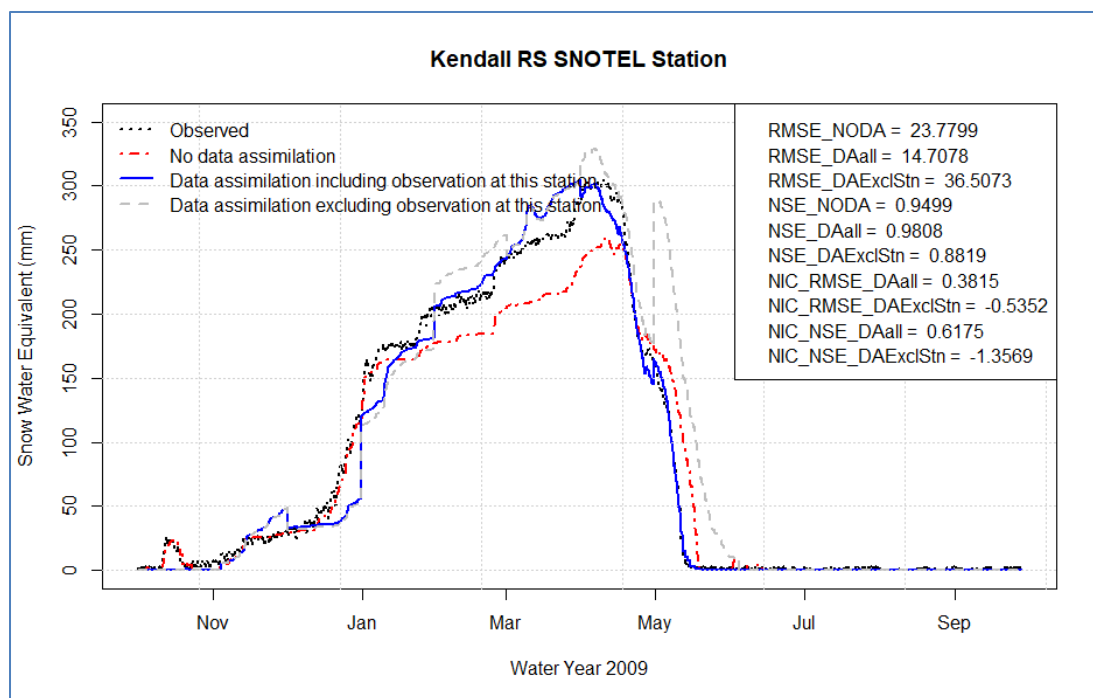


Figure 2.12 Same as Figure 2.9 but for the Kendall RS SNOTEL station.

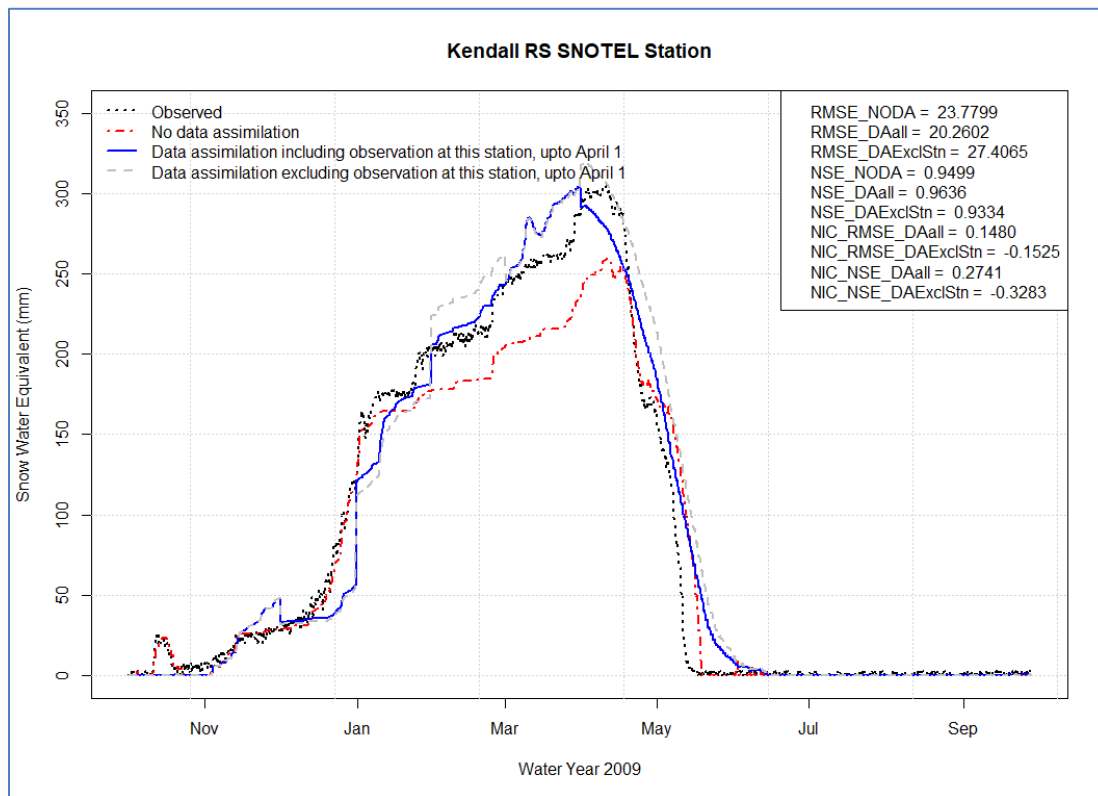


Figure 2.13 Same as Figure 2.12 but the data assimilation extends only up to April 1, 2009.

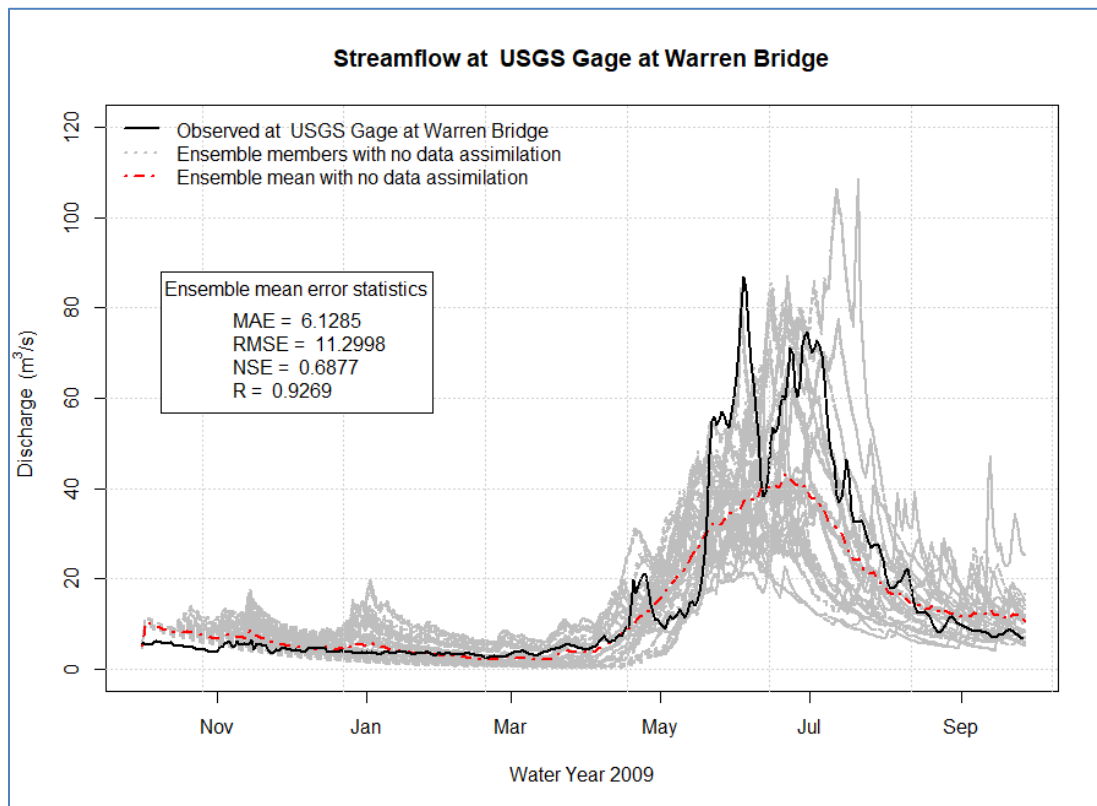


Figure 2.14 Ensemble streamflow forecast at Warren Bridge near Daniel with no data assimilation (NO-DA) for the water year 2009.

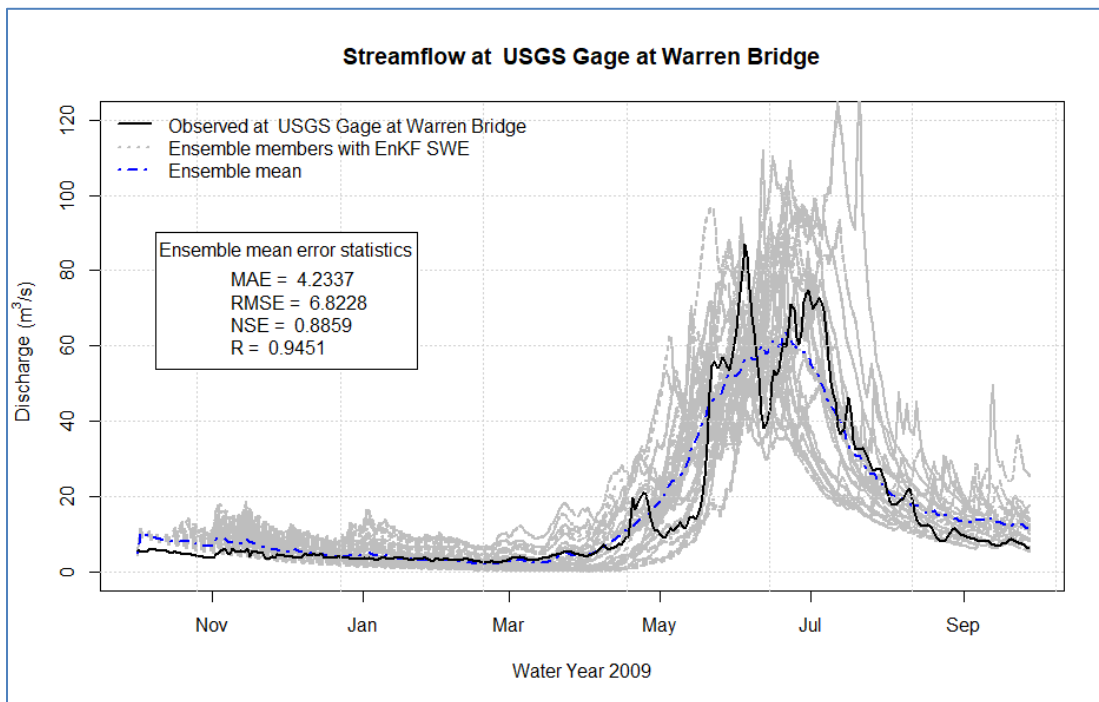


Figure 2.15 Ensemble streamflow forecast at Warren Bridge near Daniel with SNOTEL SWE data assimilated every two weeks using EnKF (EnKF-SWE) for the water year 2009.

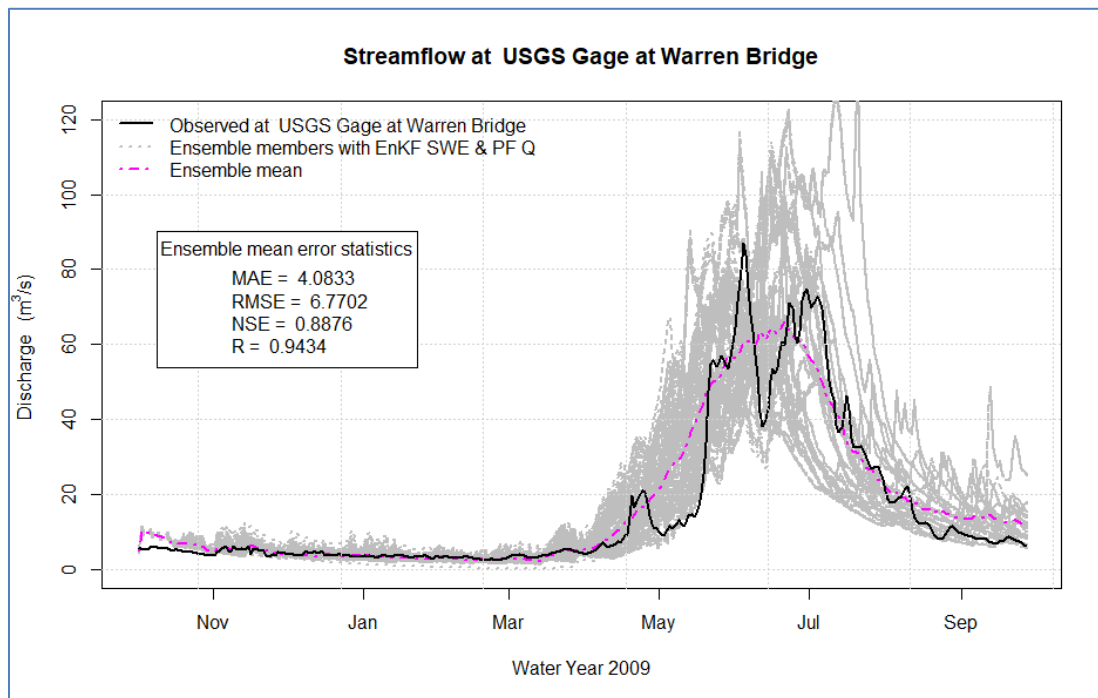


Figure 2.16 Ensemble streamflow forecast at Warren Bridge near Daniel with SNOTEL SWE data assimilated every two weeks using EnKF and assimilation of daily streamflow using PF with biweekly resampling (EnKF-SWE_PF-Q) for the water year 2009.

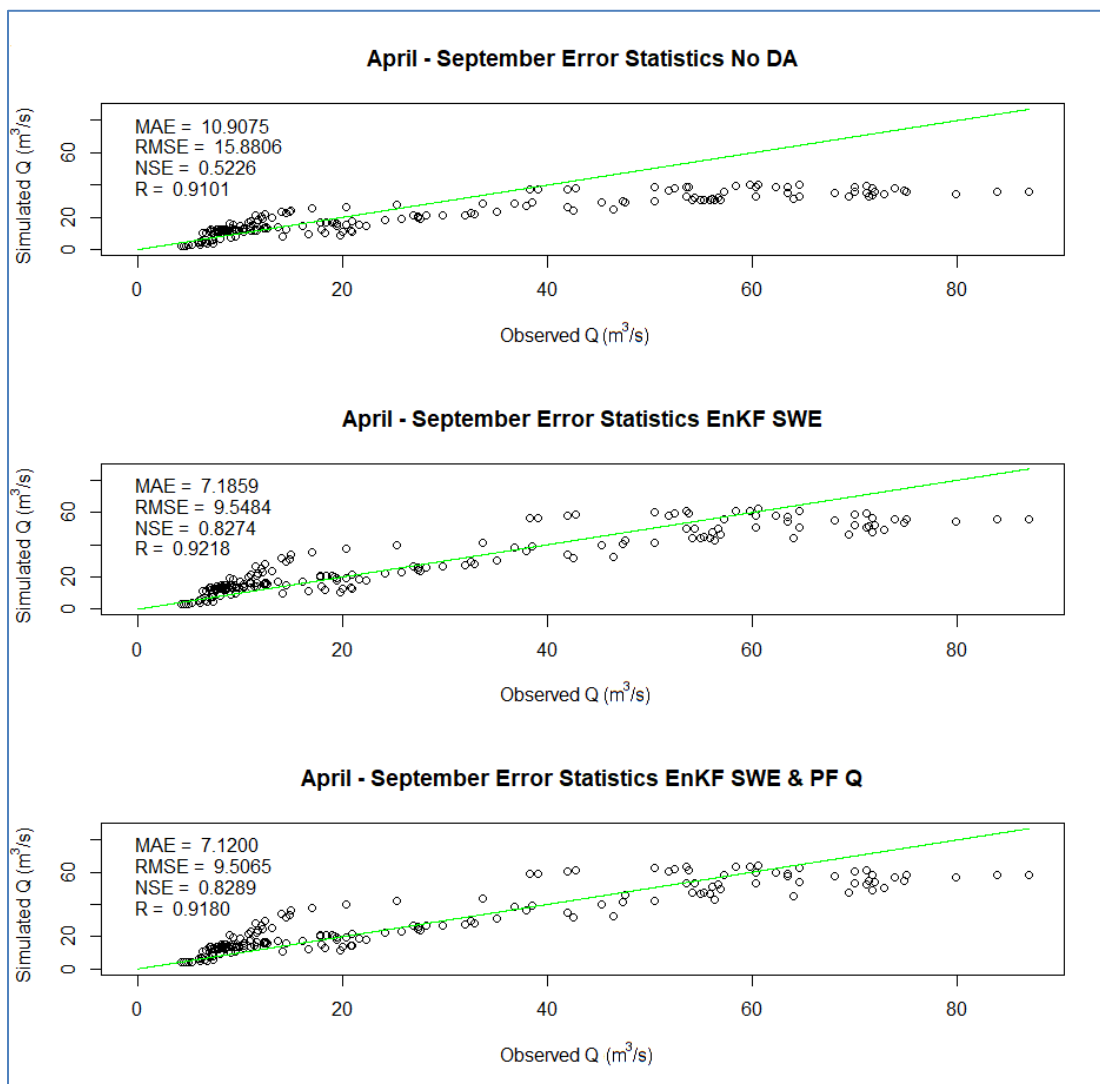


Figure 2.17 Scatter plots and error statistics of daily streamflow for the ensemble mean of April - September 2009 forecast discharge versus observed discharge at Warren Bridge near Daniel for the three scenarios No-DA, EnKF-SWE, and EnKF-SWE_PF-Q. The green line is the 1:1 line.

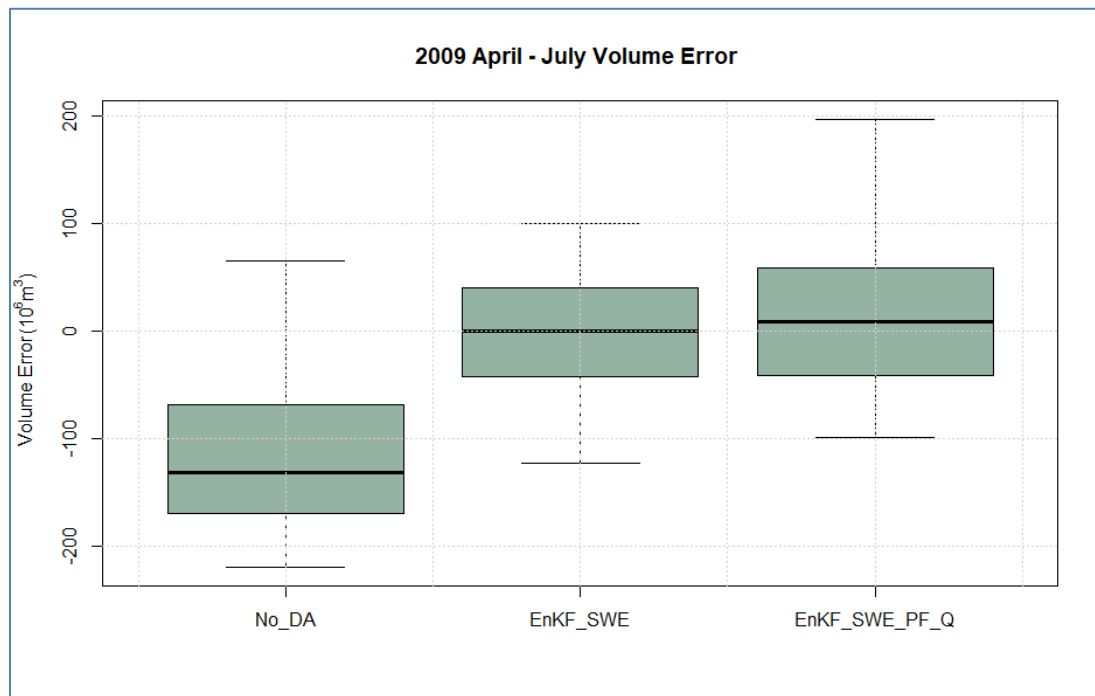


Figure 2.18 April – July volume error of ensemble streamflow forecast at Warren Bridge near Daniel for the three scenarios No-DA, EnKF-SWE, and EnKF-SWE_PF-Q for the water year 2009.

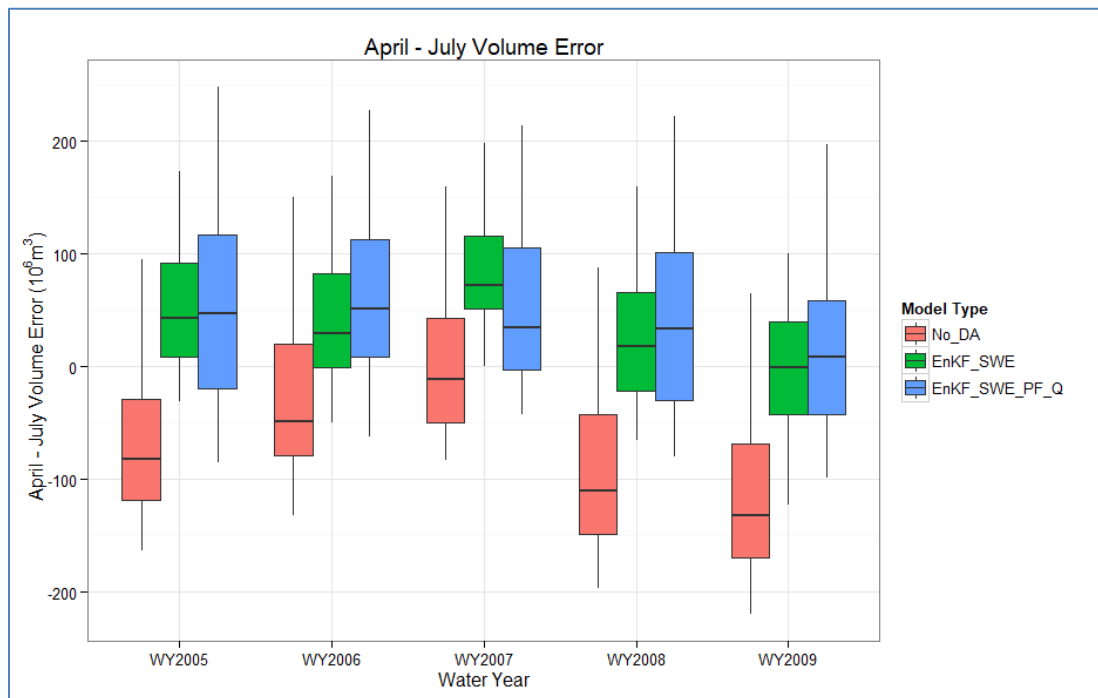


Figure 2.19 April – July volume error of ensemble streamflow forecast at Warren Bridge near Daniel for the three scenarios No-DA, EnKF-SWE, and EnKF-SWE_PF-Q for the water years 2005 - 2009.

CHAPTER 3
DATA SERVICES IN SUPPORT OF PHYSICALLY BASED, DISTRIBUTED
HYDROLOGICAL MODELS²

Abstract

Hydrology researchers and modelers spend considerable time searching for, accessing, organizing, and pre-processing model input data. The task becomes daunting when applying physically based, distributed models in operational contexts such as streamflow forecasting. As the ability to configure and populate models with data could enhance or hinder their use, tools that automate and speed up the pre-processing of input data are required to facilitate the application of physically based hydrologic models. In this paper, we introduce and evaluate a set of web-based, hydrological data services we call HydroDS that are based on the philosophy of providing ‘software as a service.’ HydroDS enables the generation of distributed (gridded) data for variables commonly used in hydrologic models in three widely used file formats: GeoTiff raster, Shapefile, and multi-dimensional NetCDF. HydroDS provides functions for watershed delineation, terrain processing, estimation of canopy variables, and retrieval of climate data. The functions can be used independently or chained together to form a workflow that performs a set of related tasks. A Python client library facilitates the scripting and execution of these workflows from a desktop computer, providing access to data processing tools from this programming environment that is platform independent and accessible to researchers with basic Python programming skills. We evaluated the data

² Authors: Tseganeh Z Gichamo, David G. Tarboton, Pabitra Dash.

services by setting up instances of the Utah Energy Balance (UEB) snowmelt model for multiple watersheds. The evaluations show that HydroDS reduces the time required to setup the model for multiple watersheds and helps avoid error prone manual data processing, thereby enhancing the application of UEB in streamflow forecasts. It also removes the requirements for installation, maintenance, and updating of often platform dependent software, and enhances reproducibility and repeatability and provides the ability to track data processing provenance with workflow scripts.

Keywords—HydroDS, data services, distributed hydrologic modeling, Geographic Information Systems, web services, Cyberinfrastructure, Utah Energy Balance snowmelt model (UEB).

Software Availability

Program name: HydroDS

Description: A set of web-based, hydrologic data services for automated generation of input data for physically based, distributed (gridded) hydrologic models.

HydroDS comprises Python modules for watershed analysis, terrain and land cover data processing, and climate data access and processing. Individual service functions are chained together to form a workflow to perform a set of related tasks.

Platform: CentOS Linux.

Cost: Free (Open source)

Access: <https://github.com/CI-WATER/Hydro-DS/>

Documentation:

<https://github.com/CI-WATER/Hydro-DS/wiki/HydroDS-Web-API-Description>

<https://www.hydroshare.org/resource/130373d3ebde4df489a4781e62211574/>

Developers: Tseganeh Z. Gichamo, David G. Tarboton, Pabitra Dash.

3.1 Introduction

Computer models are used for simulation of the hydrologic cycle to help answer questions related to water resource availability and quality, to assess the effect of change in climate or land cover, and support water resources management, along with many other applications. One of the prime motivators of current hydrological research is the need to understand and quantify the possible impacts on water resources of changes in climate, land cover, land use, population, and urbanization (Fowler et al., 2007). Such studies require modeling of hydrologic processes across a range of spatial scales. This is to say that the extent of the model domain, spacing, and support of the model elements are problem specific. In addition, given that water resource management decisions are made at scales that range from headwater watershed scale to river basins or even continental scales (Kauffeldt et al., 2014), there is a motivation for large scale modeling, which is further enhanced by national or international governing bodies' desire to predict large scale flood disaster or drought events and to develop mitigation strategies (Winsemius et al., 2013). Developers of physically-based, distributed hydrological models in recent years have been working to address the issue of sufficiently resolving local (sub-grid scale) processes in a model of river basin or continental scale (Kollet et al., 2010; Qu and Duffy, 2007; Shi et al., 2013; Wood et al., 2011).

An important challenge associated with the application of physically based, distributed hydrological models is that they require more input data than their conceptual, often lumped, counterparts. While the rationale for high resolution, physically based models is that better results can be achieved through detailed process representation, obtaining the extensive set of input data required by these models is a critical challenge.

Leonard and Duffy (2013) call this set of input data “Essential Terrestrial Variables” (ETV). Obtaining ETV’s in a format organized for use in distributed models is a significant bottleneck in distributed hydrologic modelling. The ability to configure and populate distributed models with data could enhance or hinder their use.

Prior work with regard to hydrologic data availability has focused on the task of enhancing access to data from different providers through web services (Ames et al., 2012; Horsburgh et al., 2009; Tarboton et al., 2009b) using standardized data formats (Almoradie et al., 2013b; Taylor, 2012). However, the data obtained remain the raw data provided by the sources and generally require further processing to generate suitable inputs to hydrological models.

The data pre-processing tools currently available are generally desktop based and often limited by their customization to specific hydrologic models (e.g., Kumar et al., 2009a). The increasing availability of Cyberinfrastructure resources provides an opportunity to extend such data pre-processing ability beyond the desktop environment (Wang et al., 2013) and adopt the paradigm of ‘software as a service.’ Developing data processing tools as web-based services will help to enhance access to these tools for users without necessarily requiring them to be a Cyber expert (Wright et al., 2013). Web services that can be accessed by multiple users not necessarily located at the same geographic location facilitate better collaborative problem solving (Nyerges et al., 2013; Wang, 2010). In addition, they encourage the use of standardized data formats (e.g., WaterML and NetCDF) by multiple models.

In this paper, we introduce a set of web-based, hydrological data processing services we call HydroDS. HydroDS provides a number of data processing functionalities

including watershed delineation, terrain processing, estimation of canopy variables based on the National Land Cover Database (Homer et al., 2015), and retrieval of weather forcing data. Data are stored and shared in three widely used data formats: GeoTiff raster, shapefile, and multi-dimensional NetCDF. The data services are comprised of functions that can be used independently or form workflows that integrate a number of related tasks. We also developed a Python client library that facilitates the scripting and execution of these workflows from a desktop computer, providing access to data processing tools from an accessible and relatively easy to use programming environment that is platform independent. Data processed by HydroDS can be automatically transferred to HydroShare, a platform for sharing of hydrologic data and models (Tarboton et al., 2014a).

In the next section, we provide background information on prior work dealing with data access and processing for hydrologic modeling and the need for web-based data services that motivates this work. In Section 3.3, we report the functional requirements, design, and implementation of HydroDS. In Section 3.4, we evaluate the data services using a case study of setting up instances of the Utah Energy Balance snowmelt model (UEB) for multiple watersheds. Summary and conclusions are given in Section 3.5.

3.2 Background

3.2.1 *Geospatial Data Analyses for Hydrologic Models*

Providing access to hydrological data from different repositories through web services has been the focus of the Consortium of Universities for the Advancement of Hydrologic Science, Inc. - Hydrologic Information System (CUAHSI-HIS) (Horsburgh et

al., 2009; Tarboton et al., 2009b). CUAHSI-HIS provides software tools for publishing and retrieving time series data through standardized web services in an XML format called WaterML (Beran et al., 2009; Tarboton et al., 2011; Valentine et al., 2012; Valentine et al., 2007). WaterML2 was later developed as an Open Geospatial Consortium (OGC) standard for hydrologic time series data representation and exchange across multiple information systems (Taylor, 2012). Standardized web services and protocols facilitate interoperability between different data service providers and consumers (clients) for easy access and retrieval of data.

Client applications can search for and download data made available through the CUAHSI-HIS data services. HydroDesktop, the CUAHSI-HIS data access client (Ames et al., 2012), provided an early ‘one-stop shopping’ platform to hydrologists by enabling map based selection of a watershed (or an extent of the domain of interest) and data download, extraction, and analysis. This desktop functionality has now been replaced by the CUAHSI data client (<http://data.cuahsi.org/>) web tool for CUAHSI HIS data selection and extraction. Agencies such the U.S. Geological Survey (USGS: <http://waterservices.usgs.gov/>) and National Oceanic and Atmospheric Administration - National Centers for Environmental Information (NOAA- NCEI: <http://www.ncdc.noaa.gov/cdo-web/webservices>), and other data and model service providers have also made data from their repositories accessible using web services and data standards such as WaterML and other OGC web service standards (Almoradie et al., 2013a). These systems help reduce the time spent by researchers searching for and downloading data.

Despite their growing availability, hydrological data obtained through web

services from sources such as CUAHSI-HIS, USGS, NOAA, or other organizations often need pre-processing to generate suitable inputs to hydrological models. In addition, CUAHSI-HIS compliant data services are currently limited to time series data at fixed geographic locations (e.g., points) using the Observation Data Model (ODM) (Horsburgh et al., 2008); no support is provided in CUAHSI services for multi-dimensional space-time data such as those stored in Network Common Data Form (NetCDF) data format (Rew et al., 2014). Hence, part of the data pre-processing tasks for distributed hydrological models involves organizing data in the input format suitable for the specific model (often arrays of space-time data). Input data pre-processing often starts with geospatial analyses, including watershed delineation, stream network generation, and specification of modeling units such as Hydrologic Response Units (HRU) or structured or unstructured grids of required spatial resolution. Then, input variables based on the watershed terrain, land cover characteristics, and climate forcing are mapped to the modeling units (Carlson et al., 2014). This mapping of continuous or discrete values to model units may require aggregation or interpolation in both space and time.

Extraction of hydrological variables from digital elevation models (e.g., terrain slope, aspect, topographic wetness index) also forms part of the pre-processing. In addition, some model parameters need to be generated (or estimated) from observations. For example, land cover variables such as canopy indices have to be derived based on land cover type maps or from remote sensing images, and friction coefficients have to be estimated from the vegetation and geomorphological information of river reaches. These data pre-processing tasks can take a significant portion of the hydrological modeler's time and effort, and automated data pre-processor tools have been shown to considerably

reduce the time required for model scenario setup and execution (Berry et al., 2014). An additional benefit is reproducibility, and the opportunity to support best of practice pre-processing methods, rather than expedient methods that may be selected by a user essentially preparing model inputs manually using general purpose tools available to them.

3.2.2 *Web based Data and Modelling Services*

At present, most geospatial analyses are carried out using desktop-based GIS tools. Some of these GIS tools are stand-alone software products such as the ArcGIS software suite from ESRI (<http://www.esri.com/>) or the open source software QGIS (<http://www.qgis.org/en/site/>) and GRASS (<http://grass.osgeo.org/>). Others are integrated with the hydrologic models they prepare inputs for. There are commercial and open source modeling software that support input data pre-processing as an integral part of hydrologic modeling. One example of commercial software is the MIKE SHE model's GIS-based graphical user interface and GIS database from DHI (<http://www.dhigroup.com/>). An example of open source model data processing tools is PIHM-GIS (Bhatt et al., 2008; Bhatt et al., 2014; Kumar et al., 2009a), in which a GIS framework where model input pre-processing and input and output visualization are carried out is tightly coupled to the Penn State Integrated Hydrologic Model (PIHM - <http://www.pihm.psu.edu/>).

With the increasing availability of Cyberinfrastructure, there is an opportunity to extend model input data pre-processing tools to web-based services. Such web-based services could build on similar works for general-purpose geospatial and hydrologic data analysis such as CyberGIS and HydroTerre. CyberGIS is a web-based approach to the

delivery of GIS functionality as data and software services (Wang, 2010; Wang et al., 2013; Wright et al., 2013). CyberGIS supports large scale, data intensive modelling problems with spatial analysis tools that require more than just a few processing cores. HydroTerre is a web-based hydrologic model data service (<http://www.hydroterre.psu.edu/HydroTerre/Help/Ethos.aspx>) that has made available about 200 TB of “Essential Terrestrial Variables,” including elevation, soils, geology, land cover, precipitation and atmospheric conditions, sub-watersheds and National Hydrography Dataset (NHD) stream reaches. Data are indexed by USGS NHD Hydrological Unit Code level-12 (HUC-12) sub-watersheds and can be downloaded to support detailed hydrologic modelling using PIHM or other models (Leonard and Duffy, 2013). There are also recent developments such as EcoHydroLib and WaterHUB that deal with specific models. EcoHydroLib provides workflows to set up instances for the RHESSys model and preserve metadata that enable reproducibility of the model instances (Miles, 2014). A web based modeling service is provided by WaterHUB (<http://water-hub.org/>), which allows parameterized SWAT (Soil Water Assessment Tool) models and their input data to be uploaded, run on HPC resources and shared among users (Merwade et al., 2012).

The web-based technologies underlying these services enable taking advantage of an extensive pool of high performance computation resources, distributed data storage facilities, analysis tools from multiple service/tool providers to deal with ‘spatial big data’ (Evans et al., 2013), and collaboration between researchers (possibly) remotely located from each other. From a user/client point of view, spatial analysis capabilities are readily accessible through the web without requiring any local software installation. This

eliminates data size limitation of PCs, the need to install software, and operating system (platform) dependence. A web-based development environment facilitates better collaborative problem solving (Nyerges et al., 2013; Wang, 2010), eases access to analytic tools (e.g., geospatial analyses) for non-experts (Wright et al., 2013), and enables implementation of ‘science gateway’ functionalities that provide access to HPC centers (Wilkins-Diehr et al., 2008). An example web-based development environment, integrating multiple of the above functionalities is HydroShare (<http://www.hydroshare.org/>).

HydroShare is a collaborative environment for sharing hydrologic data and models taking advantage of modern information communication technology and Cyberinfrastructure. HydroShare supports the capability for users to store their work in the hydrologically oriented resource formats including time series, geographic features and rasters, and model programs and instances. HydroShare resources created by one user may be shared with others, and HydroShare’s web service application programming interface (API) enables programmatic access to create and/or work directly with resources stored in the system (Horsburgh et al., 2015; Tarboton et al., 2014a; Tarboton et al., 2014b).

3.3 HydroDS: A set of Web-based Hydrologic Data Processing Services

3.3.1 *Required Functionality*

The following data services were identified as required to support the input pre-processing for physically based gridded models commonly used in surface water hydrology (i.e., ETVs). They are heavily influenced by the UEB model data pre-

processing tasks shown in Figure 3.1, a workflow of tasks that are required to be executed to get the inputs for the UEB model for a given watershed and specific modeling period. The data services are needed to support generation of such a workflow automatically and as web services so that a user does not need to undertake these tasks manually on a desktop PC. The UEB model was selected as a first test case model because of the need to be able to efficiently set up multiple UEB models for use in water supply forecasting research, and a desire to use this as impetus for developing general purpose model setup capability. As, the UEB model requires data that are commonly used in distributed hydrologic models (precipitation, temperature, relative humidity, wind speed, radiation), the services developed here have potential to be applicable to other gridded models. In addition, the three data formats used by HydroDS (shapefile, GeoTiff, and NetCDF) are among the most widely used formats for representing these classes of data.

- Select a model domain (geographic location of watershed of interest) and, if necessary, delineate the watershed draining to an outlet point.
- Compute hydrological variables from a digital elevation model (DEM), including slope, aspect, topographic wetness index, etc.
- Estimate canopy variables and vegetation indices such as the leaf area index based on the National Land Cover Database (NLCD).
- Perform coordinate system conversions, resampling, and sub-setting to the desired model scale including grid spacing, support, and extent.
- Retrieve weather data from national data sources (e.g., Daymet, NASA

NLDAS, etc.) and process and map to model elements.

- Convert between data formats (e.g., GeoTiff raster to NetCDF and vice versa).
- Carry out arithmetic operations on array data stored in NetCDF or GeoTiff formats.
- Create HydroShare resources from data generated by HydroDS. The data may be individual files such as a watershed delineated from a DEM or a set of model inputs and/or outputs. Also, support moving existing resources in HydroShare to HydroDS for processing.
- Create a model instance input package (e.g., all of the required input files to execute a model for a selected geospatial domain).
- Miscellaneous file manipulation services such as upload, download, delete, zip, show metadata of a resource, etc.
- Authentication and user access control for security and for providing user customized services.
- Saving work within a storage space allocated for a user and manage the contents of this storage.

3.3.2 *Design and Architecture*

Figure 3.2 shows the high-level organization of HydroDS, comprising HydroDS Services and the HydroDS Client Library. The HydroDS Services consists of data processing and user space and account management tools that were designed to meet the requirements listed above. In addition, we added functions for common hydrological data processing

tasks such as interpolation, resampling and projection of geospatial data. Some of the data accessible through HydroDS are staged on the HydroDS servers for fast access. However, temporally variable data such as meteorological forcing needs to be periodically updated by harvesting the data for recent years after it has become available.

The HydroDS tools are a set of Python functions for accessing and processing of data in raster (GeoTiff), vector (shapefile), and multi-dimensional space-time (NetCDF) formats. Each tool contains one or more atomic data processing functions, each function with a single task. Thus, for the tools that comprise HydroDS, the design and implementation approach we followed was that each function is a stand-alone service that gets executed separately. A user may input either a vector (shapefile) or raster (mask) of the domain to be modeled and for which data are to be retrieved. Users may also specify that the model domain be derived as the watershed draining to a user selected outlet location.

The watershed and terrain services are based on functions from the TauDEM (Tarboton et al., 2009a; Tarboton, 2015; Tesfa et al., 2011) and GDAL geospatial libraries (GDAL Development Team, 2014). The watershed tools delineate the watershed upstream of the outlet location after extracting a subset of the DEM and resampling it to the required grid cell size. The terrain functions involve processing of a raw DEM and extraction of hydrological variables such as slope, aspect, etc. The watershed and terrain functions deal with rasters and shapefiles, and, hence, functions for creating and editing these file formats also make up part of the services. Currently, a DEM data file containing the one arc-second (about 30 m) spatial resolution National Elevation Dataset covering the western U.S. (which was the focus of the research project supporting this

work) is available on the HydroDS server as the starting point for the watershed and terrain functions. For areas in the Contiguous U.S. (CONUS) but outside of the western U.S., there is a function to download, at run time, the one arc-second DEM from USGS web services (<ftp://rockyftp.cr.usgs.gov/vdelivery/Datasets/Staged/NED/1/IMG/>) based on user-passed boundary information in geographic coordinates. If a user wants to use different DEM data than those currently served by HydroDS, they can upload their own DEM, or move a raster resource from HydroShare to their user space in HydroDS.

The land cover services use the 2011 National Land Cover Database (Homer et al., 2015) together with a look-up table of canopy variables for each land cover category (Appendix A) to map the land cover variables into the watershed grid. The variable values are limited by the land cover classes identified and the empirical canopy variable values available. These can be updated when more and/or better information become available. For example, vegetation variables from remotely sensed Moderate Resolution Imaging Spectroradiometer (MODIS: <https://modis.gsfc.nasa.gov/>) products can be uploaded and used by the modeler.

The climate services provide access to and processing capabilities for data in NetCDF format. Daily data for precipitation, maximum and minimum temperature, vapor pressure, shortwave radiation, snow water equivalent, and day length from Daymet (Thornton et al., 2014) with 1 km spatial resolution covering the CONUS for the period 2005 - 2015 are currently available in the HydroDS server to facilitate efficient access. Hourly data of precipitation, temperature, surface pressure, shortwave and longwave radiation, zonal and meridional wind speed, and specific humidity from the National Land Data Assimilation System (NLDAS) (Mitchell et al., 2004) with horizontal

resolution of 0.125-degree latitude/ longitude covering the CONUS are available for the period 2005 – 2015. The NLDAS data are organized in yearly NetCDF files for efficiency.

The Account Management functions provide user authentication services as well as ability for the users to control the files in their user space, with functionality to upload or download data to or from their user space in HydroDS. With the linkage between HydroDS and HydroShare, a user is able to transfer data processed in HydroDS to HydroShare. This provides a mechanism by which data and model packages created by one user may be shared with others (Tarboton et al., 2014a; Tarboton et al., 2014b).

The HydroDS Client Library is a set of Python functions that can be invoked from the user's computer to make calls to HydroDS. For each data service function on the server side, a corresponding interface is implemented in the HydroDS Client Library. The HydroDS Client Library makes it easier to access these data services and thus facilitates scripting and execution of workflows that use the services from a programming environment on a desktop computer. The HydroDS Client Library can also be used by desktop applications to access the data services. An example client software that interacts with HydroDS through the Client Library is shown in Figure 3.3. This Google Map-based graphical user interface (GUI) program was developed using Python to enable calling HydroDS' watershed delineation function by graphically specifying the area of interest.

When using the Python client, the only software required by a user is a Python interpreting environment with the Python 'requests' module (<http://docs.python-requests.org/en/latest/>) installed. The data services are executed on the server side where needed service libraries and dependencies have been installed and configured, freeing the

user from these dependency configuration challenges. Transmission of function calls and data transfer between client and server uses REST HTTP protocols over the web.

3.4 Case Study: Evaluation of HydroDS with Input Data Preparation for the Utah Energy Balance Snowmelt Model

3.4.1 *Motivation*

The Colorado Basin River Forecast Center (CBRFC) provides streamflow forecasts for watersheds in the Colorado River and Great Salt Lake basins where a significant portion of the annual surface water input comes from snowmelt that primarily falls in the mountainous headwater watersheds. Currently, the CBRFC uses the NWS River Forecasting System (NWSRFS) that consists of the SNOW-17 snowmelt model, the Sacramento Soil Moisture Accounting (SAC-SMA) model, and a channel routing model based on the kinematic wave equations (Anderson, 2006; Anderson, 1973; Burnash and Singh, 1995; Peck, 1976).

The motivation for this case study arose from the desire to evaluate the UEB snowmelt model (Tarboton and Luce, 1996) as a potential replacement for the SNOW-17 model in the NWSRFS. One of the issues that needed to be addressed in order to be able to use UEB in the streamflow forecast system was whether the input data available for the energy balance model were of sufficient quality and could be efficiently prepared for forecast watersheds. In this study, we evaluated how much improvement is achieved through the use of HydroDS to acquire and pre-process the UEB input data when compared to desktop-based GIS tools. We quantified the improvement in terms of the time it takes to prepare input data using each approach, as well as demonstrate the value

of the data services to facilitate repeatability and reproducibility and tracking the provenance through an automated workflow script. In addition, the use of web services helped avoid the need for individual users to have a local data copy and data organizing software.

The UEB model is a parsimonious, physically based, point energy and mass balance model with a single ground snow layer and a vegetation component that accounts for major snow processes in forested watersheds without undue complexity involved in parameterization of multiple snow layers (Luce and Tarboton, 2010; Mahat and Tarboton, 2012, 2013, 2014; Mahat et al., 2013). It uses a modified force-restore approach to balance above surface energy exchanges with conduction into the snow and model surface temperature distinct from the single layer average snowpack temperature. It is driven by temperature, precipitation, radiation, humidity, and wind speed. Spatial variability of snow is accounted for either by using snow areal depletion curves (Luce and Tarboton, 2004) or by using a gridded approach (Sen Gupta et al., 2015).

3.4.2 *Study Watersheds and Input Data Pre-processing*

Currently, the CBRFC models are structured into watersheds that flow to NWS streamflow forecast points. As such, the modeling units are forecast watersheds, for which input data are structured independently. This makes the procedure manageable. To apply the UEB model for streamflow forecasting in the Colorado River Basin, we needed to set up a model instance for each forecast watershed. Making a model setup for each watershed using desktop tools currently in use can be time-consuming, error prone, and hard to reproduce. Recognizing that the same set of data set-up operations need to be carried out for each watershed, it is feasible to assume that a workflow script to pre-

process input data for one watershed can be reused for multiple watersheds with only slight modifications to the script. The modification only involves specifying the watershed's boundary and outlet coordinates.

A number of headwater watersheds in the Colorado River basin and the Great Salt Lake basin were selected to set up UEB inputs (Figure 3.4). These watersheds were selected as typical streamflow forecast entities in the CBRFC. The HydroDS tasks required to be executed to get complete UEB model inputs for a given watershed are shown in the flow chart in Figure 3.5. These steps are encapsulated in the workflow script, which reduces the task of a modeler to execution of a single script file. The workflow script is provided as a HydroShare resource at (Gichamo, 2019). The major inputs to this workflow script for a given watershed are the coordinates of the watershed boundary, outlet location, the start and end time of model period, and the spatial reference (projection) information in the form of EPSG Code (<http://spatialreference.org/ref/epsg/>). The commands in the workflow script can also be called interactively from any Python command line, or, as mentioned earlier the service function can be called from a user application such as shown in Figure 3.3.

3.5 Results and Discussion

Table 3.1 shows the time it takes for preparation of UEB input data for the Logan River watershed for the water year 2009 for three methods: manually on desktop PC; using automating scripts on desktop PC; and using HydroDS through a workflow script. Running the HydroDS script for a different watershed only requires modification of the watershed boundary, location of the outlet, and projection information, as mentioned

earlier, and it takes 10 minutes to run the pre-processing and package it and put it into HydroShare. It takes comparable total time (between 9 and 15 minutes) to prepare inputs for the other study watersheds using HydroDS as shown in Table 3.2

Preparing the inputs manually by a graduate student (the first author) with multiple years of experience using desktop-based GIS software takes more than 5 hours, which is cut to 2 hours and 45 minutes by simply automating the desktop tasks using scripts and that is further reduced to only 10 minutes when using HydroDS. The scripts that were used in the desktop environment are similar to those implemented in HydroDS. Thus, the difference between the time it takes HydroDS to prepare the inputs versus the time taken by scripts on a desktop PC can partly be attributed to the efficient organization of the data in HydroDS. On a desktop PC, even when using scripts that automate the processes, user intervention is necessary, for instance to locate the delineated watershed and point it to the scripts that run weather forcing pre-processing, because all the other inputs (terrain, canopy, weather forcing) have to be mapped onto the watershed grid file that defines the modeling domain.

Preparing the script to run the HydroDS took about 30 minutes (for someone who is already well familiar with the system), which is a one-time task, after which the same script can be re-used for different watersheds by only changing the watershed boundary and outlet coordinates. We also report in Table 1 the time it took to download data to a desktop PC separately, because, theoretically at least, this is a one-time operation—note most of the time here is taken by NLDAS weather data. We did not account for the time required to harvest the Daymet, NLDAS, NED DEM, and NLCD land cover data into HydroDS data servers. However, we note here that, while the HydroDS data disks, at the

time of this writing, can store up to 10 TB of data, the desktop PC on which the test was carried out has a hard disk with a capacity of 500 GB. Thus, once the pre-processing of the inputs was finished, the intermediate files had to be deleted to free up storage space. Therefore, if we need to carry out similar operations, say in few months, downloading the data again might be necessary.

Another observation in Tables 3.1 and 3.2 is that the time for weather forcing data processing is dominated by the wind data from NLDAS. This is because the NLDAS data has hourly temporal resolution for the entire CONUS compared to the daily temporal resolution of the Daymet data. In addition, the hourly data for each NLDAS weather forcing variable comes in an individual NetCDF file. To increase efficiency of HydroDS, the NLDAS data in HydroDS were pre-organized so that one NetCDF file contains data for a year for each variable, which considerably reduces the amount of processing effort. Therefore, ignoring the time for downloading data into the desktop PC, much of the difference in the NLDAS data processing time between HydroDS and the desktop PC arises from the prior organization of NLDAS data in HydroDS. This is an optimal option because the NLDAS data, after harvesting from NASA servers, were processed and organized only once before being stored on the HydroDS server. Then multiple users can benefit from this organization, thus avoiding redundant and potentially error-prone data processing by different users or by the same user multiple times.

While the services demonstrably reduced the time and effort required to prepare UEB inputs, in the long run, a more useful benefit arises from the fact that the workflow script maintains the provenance of the data processing. For instance, few months after first using the script to prepare the Logan River watershed, we came back and used the

script again with no additional work required and retrieved the exact same result. In addition, by changing the watershed boundary and outlet coordinates and the model period, the same script can be used for a different watershed. This way, HydroDS facilitates reproducibility. In addition, it provides the ability to take advantage of a pre-configured system where the user need not be concerned about the organization of the server side functions, data, software, and hardware where the dependencies are already sorted out. By providing the capability to automate the data processing steps, preserving provenance, and enhancing the reproducibility and repeatability of the hydrologic data processing, HydroDS thus provides a number of benefits of standard workflow systems (Goble and De Roure, 2009), while simplifying the responsibility of the user to handling a single Python workflow script.

On the other hand, based on our observations using the services, the provision of individual access to all atomic functions in the Python HydroDS Client Library to call individual tasks appears to be not that useful, as the workflow scripts combining multiple related tasks are often the ones that are applied. Therefore, provision of coarser grained convenience functions, e.g., watershed delineation, and hiding the component functions, e.g., subset DEM, may be more productive.

The biggest limitation of HydroDS, as it stands currently, is the fact that the services are limited to gridded data such as those used in the UEB model. A number of hydrologic models use unstructured grids or other modeling units such as Hydrologic Response Unit (HRU). The data processing services need to accommodate for such modeling configurations if they are to be used by the wide range of models currently used by hydrologic community. A related, but less critical, limitation of HydroDS is that it

only supports GeoTiff, Shapefile and NetCDF file formats. The Hierarchical Data Format (HDF - <https://www.hdfgroup.org/>) is as widely used as NetCDF and would add additional flexibility to HydroDS if it were supported. An alternative is to add HydroDS functions for conversion of data from NetCDF to other standardized data formats such as HDF and vice versa.

Another limitation of this study is that all the watersheds evaluated were headwater watersheds whose final (pre-processed and ready to be used in the model) input data have relatively small file sizes (less than 2 GB). The work in this paper deals with large basins such as the Colorado River basin by breaking them down into CBRFC forecast watersheds and handling data processing for smaller, individual watersheds. Dealing with individual forecast watersheds with relatively small sizes was a design choice that keeps the size of the data and the computational resources for pre-processing of a single watershed easily manageable, while taking advantage of automation to address multiple watersheds. Characterizing how the services perform when increasing the sizes of the watersheds, for example by integrating multiple adjacent watersheds, may be an important next step. In such a scenario, the size of the weather forcing data increases more rapidly than the other data types, and weather data processing services, which currently use serial codes, may have to deal with large datasets in NetCDF format, which could necessitate implementation of parallel processing. Additional work is also required to deal with the potential increase in processing time due to increase in size of processed data. For example, a mechanism for queuing and batch processing of large operations with asynchronous notifications to a user that the batch of tasks from a workflow script is completed would be useful, as it would not be feasible for the user to wait for the web

services to return when the execution time extends beyond the ~10 minutes reported in this paper.

Currently, extending the available service functionalities requires obtaining appropriate credentials and familiarity with the development environment and the underlying technologies including Django, GDAL, TauDEM, NetCDF library, and NetCDF operators (NCO) in addition to Python programming skills. Future developments should consider a simplified way to extend the services to cover more geospatial processing tools and data. One way to enable a relatively easy extension of the services by addition of new functionalities is adding a Software Development Kit (SDK) as a component of the HydroDS services. The SDK could be as simple as providing sample source codes to modify for new functions or support more advanced features such as tools and libraries to serve as building blocks for new tools/functions.

Finally, while these results demonstrate that HydroDS helps reduce the time and effort required for accessing and pre-processing model input data, the task of deciding on what hydrological questions to ask depends on the researcher's prior experience. In this study, deciding the case study involved a number of iterations.

3.6 Summary and Conclusions

HydroDS, a set of web-based data services providing access to hydrologic data and geospatial analysis capabilities for distributed hydrological models requiring gridded inputs was introduced in this paper. The services comprise functions for important hydrologic data processing tasks such as watershed delineation, terrain processing, estimation of canopy variables based on the NLCD, and accessing and processing of

climate data from Daymet and NLDAS. The services are composed of single task functions that can be used independently or can be chained together to form a workflow for complete generation of model inputs. A platform independent and easy to use Python library, the HydroDS Client Library, provides access to the web services. Through the HydroDS Client Library, the services can be used in a Python script or desktop application that obtains processed data from HydroDS and performs further analysis. Accessing the services requires only Python, which means that users can access them from any computing platform with Python support.

HydroDS was demonstrated by setting up Utah Energy Balance snowmelt model (UEB) instances for watersheds in the Colorado River and Great Salt Lake basins. The cases demonstrate how HydroDS helps reduce the time and effort spent by researchers for discovering, accessing, and pre-processing hydrologic model input data. A considerable part of the time saved by using HydroDS instead of desktop-based data processing comes from better organization of data in HydroDS. The Python scripting-based data processing workflows also enhance repeatability and reproducibility because the same script can be re-used. The script needs to be modified only to specify the watershed boundaries and outlet location when used for a different watershed. As the workflow script also captures all the steps towards the final model input, its provenance is preserved in the script. The ‘software as a service’ paradigm of the web services allows multiple users not to bother about the storage and organization of data, which is done in the server, and software and hardware dependencies are already sorted out.

Based on our observations using the services, the provision of access, through the HydroDS Client Library, to the atomic functions to do individual tasks appears to be not

that useful; rather the workflow scripts combining multiple coarser granular functions were more productive. The work in this paper deals with large basins such as the Colorado River Basin by breaking them down into CBRFC forecast watersheds and handling data processing for smaller, individual watersheds. This was a design choice that worked well for this study. Future studies should address the alternative approach of processing river basins such as the Colorado Basin as a whole. Future work should also extend the services to provide inputs for unstructured grid models and models using HRUs (or other equivalent tessellations of the landscape) for HydroDS to support a wider range of hydrologic models. Future development should consider provision of Software Development Kit (SDK) in HydroDS to enable (a relatively) easy extension of the services with new functionalities.

References

- Almoradie, A., Jonoski, A., Popescu, I., Solomatine, D., 2013a. Web Based Access to Water Related Data Using OGC WaterML 2.0. *International Journal of Advanced Computer Science and Applications*, this issue.
- Almoradie, A., Jonoski, A., Stoica, F., Solomatine, D., Popescu, I., 2013b. Web-based flood information system: case study of Somesul-Mare, Romania. *J. Environ. Eng. Manage* 12 1065-1070.
- Ames, D.P., Horsburgh, J.S., Cao, Y., Kadlec, J., Whiteaker, T., Valentine, D., 2012. HydroDesktop: Web services-based software for hydrologic data discovery, download, visualization, and analysis. *Environmental Modelling & Software* 37 146-156.
- Anderson, E., 2006. Snow accumulation and ablation model–SNOW-17, NWSRFS Users Manual Documentation, Office of Hydrologic Development, NOAA’s National Weather Service.
- Anderson, E.A., 1973. National Weather Service river forecast system--snow accumulation and ablation model. *TECHNICAL MEMORANDUM NWS HYDRO-17, NOVEMBER 1973*. 217 P.
- Beran, B., Goodall, J., Valentine, D., Zaslavsky, I., Piasecki, M., 2009. Standardizing access to hydrologic data repositories through web services, *Advanced Geographic Information Systems & Web Services*, 2009. GEOWS'09. International Conference on. IEEE, pp. 64-67.

- Berry, P., Bonduá, S., Bortolotti, V., Cormio, C., Vasini, E.M., 2014. A GIS-based open source pre-processor for georesources numerical modeling. *Environmental Modelling & Software* 62(0) 52-64.
- Bhatt, G., Kumar, M., Duffy, C.J., 2008. Bridging the Gap between Geohydrologic Data and Distributed Hydrologic Modeling, In: Sánchez-Marrè, M., Béjar, J., Comas, J., Rizzoli, A., Guariso, G. (Eds.), *iEMSs 2008: International Congress on Environmental Modelling and Software Integrating Sciences and Information Technology for Environmental Assessment and Decision Making 4th Biennial Meeting of iEMSs*. International Environmental Modelling and Software Society (iEMSs): Barcelona, Catalonia pp. 743-750.
- Bhatt, G., Kumar, M., Duffy, C.J., 2014. A tightly coupled GIS and distributed hydrologic modeling framework. *Environmental Modelling & Software* 62(0) 70-84.
- Burnash, R., Singh, V., 1995. The NWS river forecast system-Catchment modeling. *Computer models of watershed hydrology*. 311-366.
- Carlson, J.R., David, O., Lloyd, W.J., Leavesley, G.H., Rojas, K.W., Green, T.R., Arabi, M., Yaeger, L., Kipka, H., 2014. Data provisioning for the Object Modeling System (OMS), *Proceedings - 7th International Congress on Environmental Modelling and Software: Bold Visions for Environmental Modeling*, iEMSs 2014, pp. 230-237.
- Evans, M.R., Oliver, D., Yang, K., Shekhar, S., 2013. *Enabling Spatial Big Data via CyberGIS: Challenges and Opportunities*. *CyberGIS: Fostering a New Wave of Geospatial Innovation and Discovery*. Springer Book.
- Fowler, H., Blenkinsop, S., Tebaldi, C., 2007. Linking climate change modelling to impacts studies: recent advances in downscaling techniques for hydrological modelling. *International Journal of Climatology* 27(12) 1547-1578.
- GDAL Development Team, 2014. *GDAL - Geospatial Data Abstraction Library, Version 1.11.1*. Open Source Geospatial Foundation.
- Gichamo, T.Z., 2019. HydroDS UEB model input setup Python scripts, HydroShare, <http://www.hydroshare.org/resource/cad5dd0c4106489e87db2e8366dd66b1>
- Goble, C., De Roure, D., 2009. The impact of workflow tools on data-centric research, In: Tony Hey, Stewart Tansley, Tolle, K.i. (Eds.), *The fourth paradigm: data-intensive scientific discovery*. MICROSOFT RESEARCH: REDMOND, WASHINGTON, pp. 137 - 145.
- Homer, C.G., Dewitz, J.A., Yang, L., Jin, S., Danielson, P., Xian, G., Coulston, J., Herold, N.D., Wickham, J., Megown, K., 2015. Completion of the 2011 National Land Cover Database for the conterminous United States-Representing a decade of land cover change information. *Photogrammetric Engineering and Remote Sensing* 81(5) 345-354.
- Horsburgh, J.S., Morsy, M.M., Castronova, A.M., Goodall, J.L., Gan, T., Yi, H., Stealey, M.J., Tarboton, D.G., 2015. Hydroshare: Sharing diverse environmental data types and models as social objects with application to the hydrology domain. *JAWRA Journal of the American Water Resources Association*.
- Horsburgh, J.S., Tarboton, D.G., Maidment, D.R., Zaslavsky, I., 2008. A relational model for environmental and water resources data. *Water Resources Research* 44(5).

- Horsburgh, J.S., Tarboton, D.G., Piasecki, M., Maidment, D.R., Zaslavsky, I., Valentine, D., Whitenack, T., 2009. An integrated system for publishing environmental observations data. *Environmental Modelling & Software* 24(8) 879-888.
- Kauffeldt, A., Wetterhall, F., Pappenberger, F., Salamon, P., Thielen, J., 2014. Technical review of large-scale hydrological models for implementation in operational flood forecasting schemes on continental level.
- Kim, D., Kaluarachchi, J., 2014. Predicting streamflows in snowmelt-driven watersheds using the flow duration curve method. *Hydrology and Earth System Sciences* 18(5) 1679-1693.
- Kollet, S.J., Maxwell, R.M., Woodward, C.S., Smith, S., Vanderborght, J., Vereecken, H., Simmer, C., 2010. Proof of concept of regional scale hydrologic simulations at hydrologic resolution utilizing massively parallel computer resources. *Water Resources Research* 46(4).
- Kumar, M., Bhatt, G., Duffy, C.J., 2009. An efficient domain decomposition framework for accurate representation of geodata in distributed hydrologic models. *International Journal of Geographical Information Science* 23(12) 1569-1596.
- Leonard, L., Duffy, C.J., 2013. Essential Terrestrial Variable data workflows for distributed water resources modeling. *Environmental Modelling & Software* 50 85-96.
- Luce, C.H., Tarboton, D.G., 2004. The application of depletion curves for parameterization of subgrid variability of snow. *Hydrological Processes* 18(8) 1409-1422.
- Luce, C.H., Tarboton, D.G., 2010. Evaluation of alternative formulae for calculation of surface temperature in snowmelt models using frequency analysis of temperature observations. *Hydrology and Earth System Sciences* 14(3) 535-543.
- Mahat, V., Tarboton, D.G., 2012. Canopy radiation transmission for an energy balance snowmelt model. *Water Resources Research* 48(1) W01534.
- Mahat, V., Tarboton, D.G., 2014. Representation of canopy snow interception, unloading and melt in a parsimonious snowmelt model. *Hydrological Processes* 28(26) 6320-6336.
- Mahat, V., Tarboton, D.G., Molotch, N.P., 2013. Testing above- and below-canopy representations of turbulent fluxes in an energy balance snowmelt model. *Water Resources Research* 49(2) 1107-1122.
- Marsh, C.B., Pomeroy, J.W., Spiteri, R.J., 2012. Implications of mountain shading on calculating energy for snowmelt using unstructured triangular meshes. *Hydrological Processes* 26(12) 1767-1778.
- Merwade, V., Feng, W., Zhao, L., Song, C.X., 2012. WaterHUB: a resource for students and educators for learning hydrology, *Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment: Bridging from the eXtreme to the campus and beyond*. ACM, p. 59.
- Miles, B., Band, L.E., 2015. Ecohydrology Models without Borders?, In: R. Denzer, R.A., G. Schimak and J. Hřebíček (Ed.), *International Symposium on Environmental Software Systems. Infrastructures, Services and Applications*. Springer, pp. 311-320.
- Miles, B.C., 2014. Small-scale residential stormwater management in urbanized

- watersheds: A geoinformatics-driven ecohydrology modeling approach. Ph.D. Thesis, The University of North Carolina at Chapel Hill.
- Miller, W.P., Piechota, T.C., Gangopadhyay, S., Pruitt, T., 2011. Development of streamflow projections under changing climate conditions over Colorado River basin headwaters. *Hydrology and Earth System Sciences* 15(7) 2145.
- Mitchell, K.E., Lohmann, D., Houser, P.R., Wood, E.F., Schaake, J.C., Robock, A., Cosgrove, B.A., Sheffield, J., Duan, Q., Luo, L., Higgins, R.W., Pinker, R.T., Tarpley, J.D., Lettenmaier, D.P., Marshall, C.H., Entin, J.K., Pan, M., Shi, W., Koren, V., Meng, J., Ramsay, B.H., Bailey, A.A., 2004. The multi-institution North American Land Data Assimilation System (NLDAS): Utilizing multiple GCIP products and partners in a continental distributed hydrological modeling system. *Journal of Geophysical Research: Atmospheres* 109(D7) D07S90.
- Molotch, N.P., Bales, R.C., 2005. Scaling snow observations from the point to the grid element: Implications for observation network design. *Water Resources Research* 41(11) W11421.
- Nyerges, T.L., Roderick, M.J., Avraam, M., 2013. CyberGIS design considerations for structured participation in collaborative problem solving. *International Journal of Geographical Information Science* 27(11) 2146-2159.
- Peck, E.L., 1976. Catchment modeling and initial parameter estimation for the National Weather Service river forecast system. Office of Hydrology, National Weather Service.
- Qu, Y., Duffy, C.J., 2007. A semidiscrete finite volume formulation for multiprocess watershed simulation. *Water Resources Research* 43(8).
- Rew, R., Davis, G., Emmerson, S., Davies, H., Hartnett, E., Heimbigner, D., Fisher, W., 2014. NetCDF Documentation (<http://www.unidata.ucar.edu/software/netcdf/docs/>). Unidata, University Corporation for Atmospheric Research (UCAR) Community Programs (UCP).
- Sen Gupta, A., Tarboton, D., Hummel, P., Brown, M., Habib, S., 2015. Integration of an energy balance snowmelt model into an open source modeling framework. *Environmental Modelling & Software* 68 205-218.
- Shi, Y., Davis, K.J., Duffy, C.J., Yu, X., 2013. Development of a Coupled Land Surface Hydrologic Model and Evaluation at a Critical Zone Observatory. *Journal of Hydrometeorology* 14(5).
- Tarboton, D., Schreuders, K., Watson, D., Baker, M., 2009a. Generalized terrain-based flow analysis of digital elevation models, Proceedings of the 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation, Cairns, Australia, pp. 2000-2006.
- Tarboton, D.G., 2015. Terrain Analysis Using Digital Elevation Models (TauDEM). Utah Water Research Laboratory, Utah State University, Logan, Utah.
- Tarboton, D.G., Horsburgh, J., Maidment, D., Whiteaker, T., Zaslavsky, I., Piasecki, M., Goodall, J., Valentine, D., Whitenack, T., 2009b. Development of a community hydrologic information system, 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation, ed. RS Anderssen, RD Braddock and LTH Newham, Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in

- Simulation, pp. 988-994.
- Tarboton, D.G., Idaszak, R., Horsburgh, J., Heard, J., Ames, D., Goodall, J., Band, L., Merwade, V., Couch, A., Arrigo, J., 2014a. HydroShare: Advancing Collaboration through Hydrologic Data and Model Sharing, Proceedings of the 7th International Congress on Environmental Modelling and Software, San Diego, California, USA, International Environmental Modelling and Software Society (iEMSs), ISBN, pp. 978-988.
- Tarboton, D.G., Idaszak, R., Horsburgh, J.S., Heard, J., Ames, D., Goodall, J.L., Band, L.E., Merwade, V., Couch, A., Arrigo, J., Hooper, R., Valentine, D., Maidment, D., 2014b. A Resource Centric Approach for Advancing Collaboration Through Hydrologic Data and Model Sharing, 11th International Conference on Hydroinformatics, HIC 2014: New York City, USA.
- Tarboton, D.G., Luce, C.H., 1996. Utah energy balance snow accumulation and melt model (UEB). Citeseer.
- Tarboton, D.G., Maidment, D., Zaslavsky, I., Ames, D., Goodall, J., Hooper, R.P., Horsburgh, J., Valentine, D., Whiteaker, T., Schreuders, K., 2011. Data Interoperability in the Hydrologic Sciences, Proceedings of the Environmental Information Management Conference, pp. 132-137.
- Taylor, P., 2012. OGC WaterML 2.0: Part 1-Timeseries. Open Geospatial Consortium Implementation Standard, OGC 10-126r3, 149pp.
- Tesfa, T.K., Tarboton, D.G., Watson, D.W., Schreuders, K.A., Baker, M.E., Wallace, R.M., 2011. Extraction of hydrological proximity measures from DEMs using parallel processing. *Environmental Modelling & Software* 26(12) 1696-1709.
- Thornton, P.E., Thornton, M.M., Mayer, B.W., Wilhelmi, N., Wei, Y., Devarakonda, R., Cook, R.B., 2014. Daymet: Daily Surface Weather Data on a 1-km Grid for North America, Version 2. Data set. Available on-line [<http://daac.ornl.gov>] from Oak Ridge National Laboratory Distributed Active Archive Center, Oak Ridge, Tennessee, USA., p. Medium: X.
- Valentine, D., Taylor, P., Zaslavsky, I., 2012. WaterML, an information standard for the exchange of in-situ hydrological observations, EGU General Assembly Conference Abstracts, p. 13275.
- Valentine, D., Zaslavsky, I., Whitenack, T., Maidment, D.R., 2007. Design and implementation of CUAHSI WATERML and WaterOneFlow Web services, Proceedings of the Geoinformatics 2007 Conference, San Diego, California, pp. 5-3.
- Wang, S., 2010. A CyberGIS Framework for the Synthesis of Cyberinfrastructure, GIS, and Spatial Analysis. *Annals of the Association of American Geographers* 100(3) 535-557.
- Wang, S., Anselin, L., Bhaduri, B., Crosby, C., Goodchild, M.F., Liu, Y., Nyerges, T.L., 2013. CyberGIS software: a synthetic review and integration roadmap. *International Journal of Geographical Information Science* 27(11) 2122-2145.
- Wilkins-Diehr, N., Gannon, D., Klimeck, G., Oster, S., Pamidighantam, S., 2008. TeraGrid science gateways and their impact on science. *Computer* 41(11) 32-41.
- Winsemius, H., Van Beek, L., Jongman, B., Ward, P., Bouwman, A., 2013. A framework for global river flood risk assessments. *Hydrology and Earth System Sciences*

17(5) 1871-1892.

- Wood, E.F., Roundy, J.K., Troy, T.J., van Beek, L.P.H., Bierkens, M.F.P., Blyth, E., de Roo, A., Döll, P., Ek, M., Famiglietti, J., Gochis, D., van de Giesen, N., Houser, P., Jaffé, P.R., Kollet, S., Lehner, B., Lettenmaier, D.P., Peters-Lidard, C., Sivapalan, M., Sheffield, J., Wade, A., Whitehead, P., 2011. Hyperresolution global land surface modeling: Meeting a grand challenge for monitoring Earth's terrestrial water. *Water Resources Research* 47(5) W05301.
- Wright, D.J., Kopp, S., Brown, C., 2013. Esri Position Paper – 2013 CyberGIS '13 All Hands Meeting.
- You, J., Tarboton, D., Luce, C., 2014. Modeling the snow surface temperature with a one-layer energy balance snowmelt model. *Hydrology and Earth System Sciences* 18(12) 5061-5076.

Table 3.1 UEB input processing time for the Logan River Watershed for the WY 2009

Data preparation method	Time (min)						Copying Data
	Watershed Delineation	Preparing Terrain Variables	Preparing Canopy Variables	Preparing Weather Forcing Data		Total	
				Daymet	NLDAS (wind)		
Manual on desktop PC	60	15	40	125*	75**	315 (5.25 hrs)	120 min
Using scripts on desktop PC	25	7	18	40	75	165 (2.75 hrs)	all of it NLDAS data copy
Using HydroDS***	1	0.5	0.5	2	6	10 (0.17 hrs)	NA

* Not including time for data download and time spent troubleshooting the errors that occurred during the data processing. The whole processes, including error troubleshooting, took about 6 hours.

** Using scripts. It proved to be too laborious to process this manually—after about an hour of trying, decided to write the script, which took about two hours, then finish the work from the script.

*** Preparing HydroDS Script takes about 30 minutes.

Table 3.2 UEB input data pre-processing time using HydroDS for the WY 2009

Study Watershed	Time (min)					Total
	Watershed Delineation	Preparing Terrain Variables	Preparing Canopy Variables	Preparing Weather Forcing Data		
				Daymet	NLDAS (wind)	
Animas above Durango	1	0.5	0.5	2	9	13
Blue above Dillon	0.5	0.5	0.5	1.0	9	11.5
Dolores above Mcphee	1	0.5	0.5	1.5	11	14.5
Frazer at Winter Park	0.25	0.25	0.25	1.0	9	10.75
Green above Warren Bridge	0.5	0.5	0.5	1.5	6	9
Logan above First Dam	1	0.5	0.5	2	6	10
Uncompahgre above Ridgeway	0.5	0.25	0.25	1.0	12	14
Williams Fork above Williams Fork	0.25	0.25	0.25	1.0	13	14.75

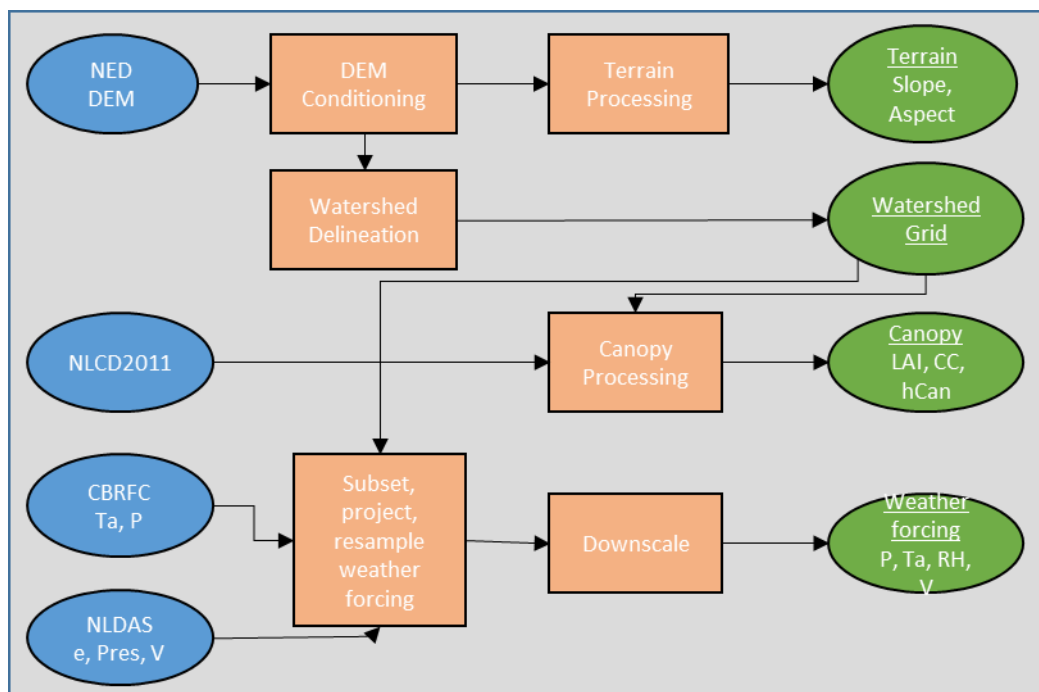


Figure 3.1 Workflow for the Utah Energy Balance snowmelt model (UEB) input preparation.

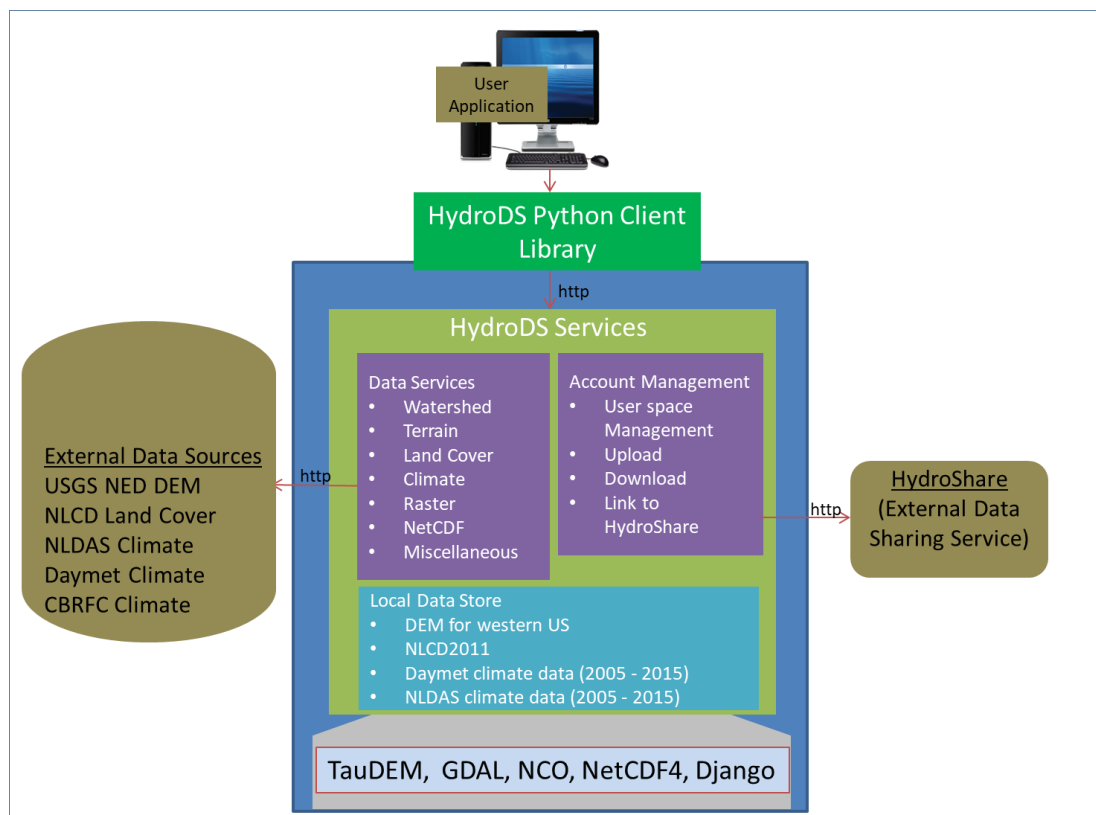


Figure 3.2 High-level architecture of HydroDS. HydroDS comprises HydroDS Services and HydroDS Python Client Library.

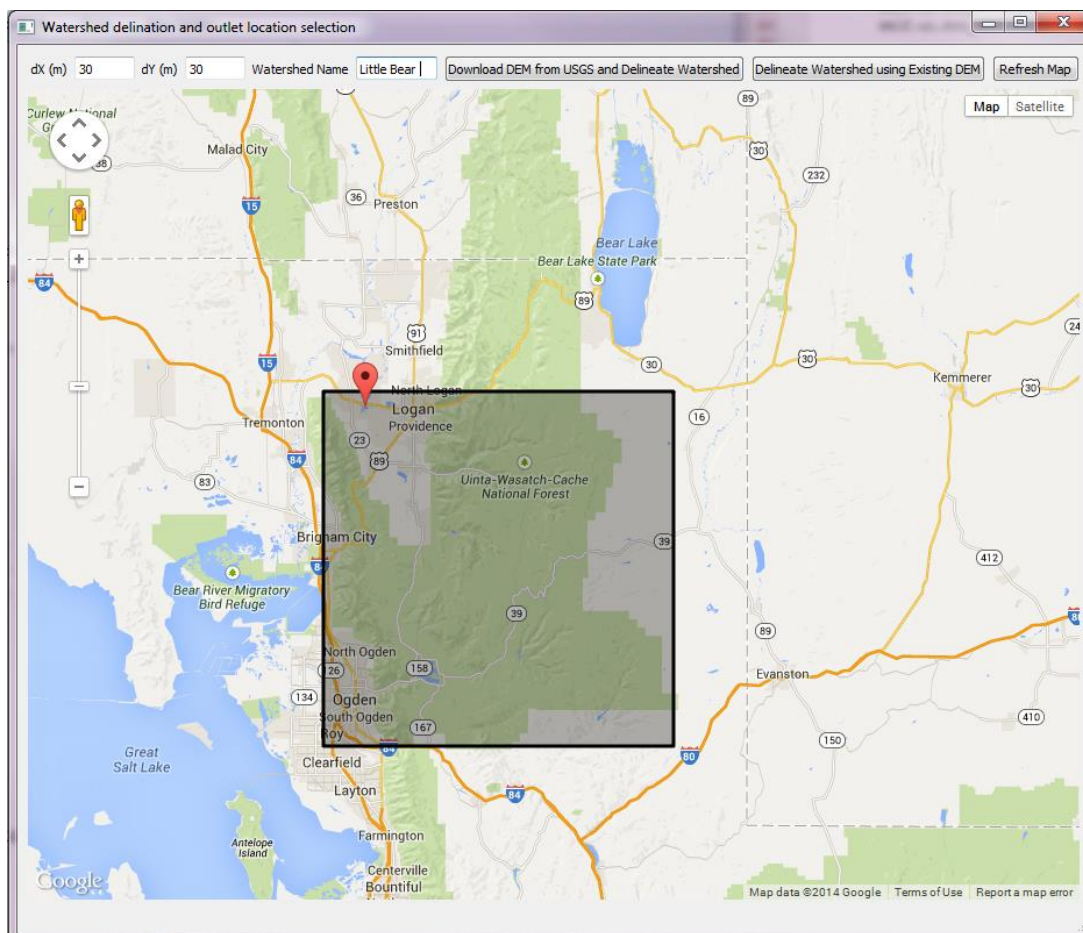


Figure 3.3 A desktop, Google Map-based GUI program for accessing USGS DEM and watershed delineation through HydroDS. Google Map drawing tools are used to specify the bounding box around the watershed of interest and Google Map marker is used to select an approximate watershed outlet location.

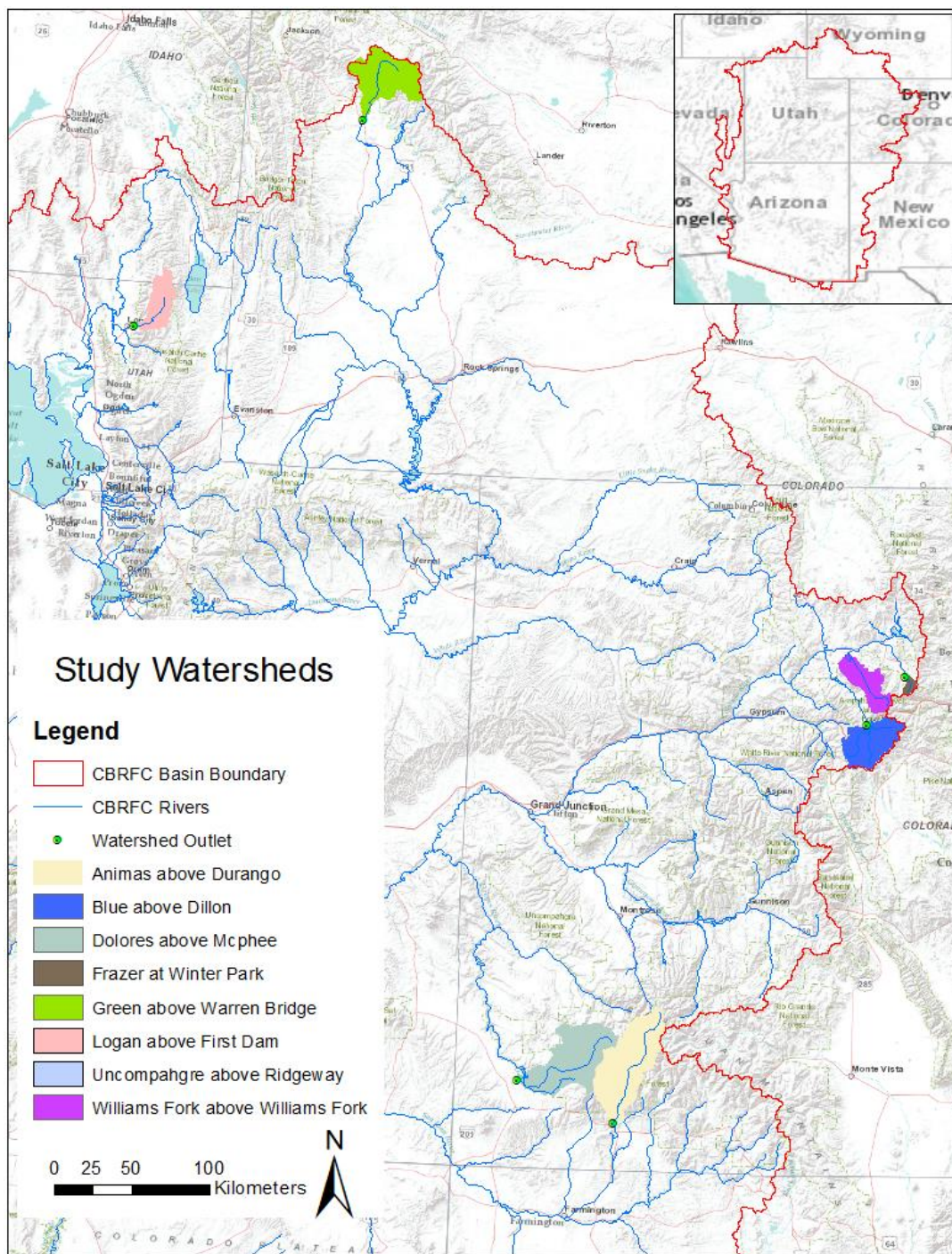


Figure 3.4 Map showing study watersheds. These watersheds were selected as typical streamflow forecast entities in the CBRFC.

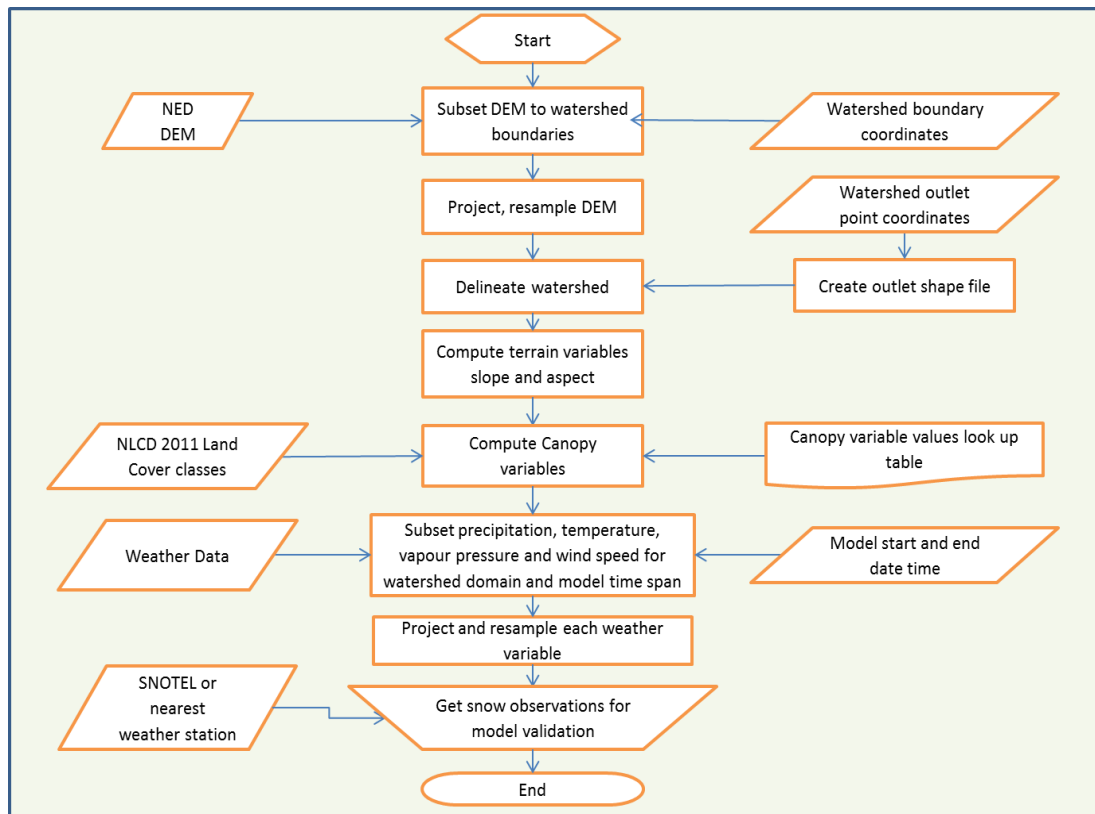


Figure 3.5 Flow chart of the Utah Energy Balance snowmelt model (UEB) input data preparation steps.

CHAPTER 4

UEB PARALLEL: DISTRIBUTED SNOW ACCUMULATION AND MELT
MODELING USING PARALLEL COMPUTING³**Abstract**

The Utah Energy Balance (UEB) model is a distributed snow accumulation and melt model that supports the detailed simulation of snow processes on a fine grid over a watershed. To enhance the computational efficiency of this model, we developed and compared two parallel versions of the model, one using the Message Passing Interface (MPI) and the other using NVIDIA's Compute Unified Device Architecture (CUDA) code on Graphics Processing Unit (GPU). For the implementation with MPI, we tested the performance of the model with an increase in the number of processing cores by calculating speed-up and efficiency and comparing to the ideal speed-up from Amdahl's law. Model simulation timing tests show that when running as a serial code (using a single process), about 1.7% of the total simulation time was spent on input/output (IO) read/write operations. The effect of these IO operations becomes more pronounced with increased number of processes. As a result, although the computation kernel scales well as the number of processors increases, the efficiency of the parallel code as a whole degrades. Using IO operations that carry out reading and writing of multiple arrays at once, instead of making multiple reading/writing of single arrays, improved the performance to some degree. The CUDA GPU implementation demonstrated that satisfactory performance with CUDA GPU can be obtained without necessarily requiring

³ Authors: Tseganeh Z Gichamo, David G. Tarboton.

a major re-work of the existing UEB MPI code.

Keywords—Utah Energy Balance snowmelt model (UEB); Message Passing Interface (MPI); Compute Unified Device Architecture (CUDA); Graphics Processing Unit (GPU); Parallel IO.

Software Availability

Program name: UEB Parallel.

Description: Parallel version of the Utah Energy Balance snowmelt model (UEB).

Platform: Platform independent. Tested on Microsoft Windows & Linux (CentOS 6.x).

Source language: C++ / CUDA.

Cost/License: Free / Open source, GNU General Public License.

Developers: Tarboton research group, Utah State University.

Availability: <http://github.com/dtarb/ueb>

4.1 Introduction

Hydrological models are used to predict environmental flow of water under diverse drivers of change that are complex and heterogeneous. One of the prime motivators of current hydrological research is the need to understand and quantify the possible impacts on water resources of changes in climate, land cover, land use, population and urbanization (Fowler et al., 2007). Such studies may require modeling of the hydrologic processes at various scales, ranging from headwater watersheds to river basins scales. As the terrestrial water cycle is affected by its interactions with the atmospheric and oceanic processes, hydrological models at river basin or global scale may also need to consider the various pathways of water in the global cycle and magnitudes of feed-backs between different layers/components of the cycle (Levine and Salvucci, 1999; Maxwell et al., 2014; Paniconi and Putti, 2015). A few years ago, Wood

et al. (2011) made a call for "Hyperresolution global land surface modeling" to sufficiently resolve local processes in a model of global or continental scale.

This task of modeling the hydrologic cycle at large scale with sufficient resolution processes poses multiple challenges. One of these challenges is the desire to use high-performance computing (HPC) resources to reduce computational time or increase the level of detail (and hence complexity) at which these problems are investigated. On the other hand, the availability of HPC resources is increasing. This, coupled with the recognition of the scientific needs for undertaking large scale hydrologic simulation, has led to development of simulation models that implement parallel processing technologies. For example (Kollet et al., 2010) present results of a study where an integrated multi-dimensional modeling problem with a number of unknowns in the order of 10^9 was solved within a feasible simulation time. The challenge for hydrological modelers is thus shifting from the lack of computing resources to reconfiguring their modeling software to be able to take advantage of these new resources.

It should be noted here that parallel programming in hydrologic and environmental modeling is not a new opportunity or issue (e.g., Paglieri et al., 1997; Rao, 2004). However, in the past decade a strong argument has been made that the basic approach to software development should incorporate concurrency (multi-processes) programming because of the "power wall," the upper limit imposed on the clock speed of single core due to overheating of high-frequency cores and other efficiency/optimization considerations (Brodtkorb et al., 2013; Sutter, 2005; Sutter and Larus, 2005). Concurrency programming has also been spurred by programming interfaces, i.e., standard definitions that abstract away most of the low level operations and a number of library

implementations of these interfaces, thereby freeing a research programmer to focus on domain-specific modeling issues (e.g., MPI <http://www.mpi-forum.org/>, OpenMP <http://openmp.org/wp/>).

Given the desire to apply more physically based, distributed, high-resolution hydrologic models, and given the opportunities offered by the parallel programming standards and libraries, the question then becomes what method to choose for a given model and what factors affect efficient scaling of the modeling code. In this study, we evaluated parallel processing implementations of the Utah Energy Balance (UEB) snow accumulation and melt model. Implementation of parallel programming methods in UEB is expected to facilitate its application in areas such as streamflow forecast where there is an increased interest to incrementally evaluate and adopt energy balance snowmelt models (Franz et al., 2008).

We evaluated two implementations: one using the MPICH2 library of the Message Passing Interface (MPI) specification (Gropp et al., 2005), and the other using NVIDIA's Compute Unified Device Architecture (CUDA) code on Graphics Processing Units (GPUs) (Nickolls et al., 2008). The MPI is a distributed memory programming approach that promises good efficiency for the distributed UEB model that requires independent data for different model grid cells. On the other hand, the CUDA code, with its compatibility to C++, enhances the accessibility of general-purpose GPUs that have ability to handle compute-intensive tasks. The computational performances of the parallel codes were compared using simulations of the Logan River Watershed, Utah, for a period of five years. For the implementation with MPI, we evaluated the speed-up and efficiency of the code with increasing number of processors and compared the speed-up trend with

the ideal speed-up from Amdahl's law (Amdahl, 1967, 2007; Gustafson, 1988). With regard to the application of GPUs, Neal et al. (2010) had found earlier that, even though their GPU code was faster and more efficient than their MPI implementation, the development time it required was prohibitive. In contrast, Tristram et al. (2014) report that not only GPUs were more cost efficient for their application, but also achieving satisfactory speed-up with GPUs did not require major refactoring of their existing code. For the CUDA implementation of the UEB code in this study, we also evaluated if satisfactory performance could be achieved by the GPU code without major refactoring of the code.

This paper is organized as follows. In the next section, brief discussion of factors to consider in parallel programming based on review of recent literature is given (focusing on hydrologic models). In Section 4.3, Methods, we describe the UEB model, the algorithms for the parallel implementations, the modeling case study to test the performance of the parallel codes, and the performance metrics. Results of the tests and discussion are given in Section 4.4 followed by conclusions in Section 4.5.

4.2 Factors to Consider in Parallel Programming

The choice of a particular parallel programming approach may depend on a number of factors including familiarity with the programming interface, ease of adaptation of existing serial code to a parallel version and the data/memory configuration of the modeling problem to solve. Neal et al. (2010) investigated the application for 2D flood inundation modeling of three of the commonly used programming methods: shared memory Open Multi-Processing (OpenMP), distributed memory Message Passing

Interface (MPI) and Graphics Processing Unit (GPU). They tested the three approaches with respect to applicability to a given problem code, parallel code efficiency achieved, and required implementation effort (development time). They concluded that the MPI approach was the most suitable compromise between efficiency achieved and programming complexity involved. They found that even though the GPU code was the fastest and most efficient, the development time it required was prohibitive.

Another important factor in parallel programming of simulation models is domain/data decomposition among processes. Data partitioning schemes often try to address the issue of load balancing between multiple processors. A good example is a 2D flood inundation model where some of the grid cells in the flood plain remain dry for part of the model time, resulting in some idle process time. Data partitioning schemes should strive to minimize such idle times (Brodtkorb et al., 2012; Sanders et al., 2010). With respect to hydrological models, domain decomposition is related to the flow dependencies (upstream-to-downstream) between computational sub-domains, where the computational sub-domain can be a Representative Elementary Watershed (REW), Hydrologic Response Unit (HRU), a structured/ unstructured grid cell, or a river reach. In addition, load imbalance may arise from the spatial variability of processes considered such as areas covered with snow versus those with no snow, upstream hill slope versus riparian area or river channel, presence and type of vegetation, etc. Some of the approaches used in recent research are described below.

The first one is the use of multiple layers in which simulation units (grids) are divided based on their degree of dependency on upstream units. Accordingly, units that do not depend on other units are placed at highest priority layer, and units that depend

only on a single unit are placed in the second highest priority layer, and so on (Liu et al., 2014). Another approach involves dynamic (run-time) allocation of a data partition to an available idle processor once the partition no longer depends on upstream processes (Li et al., 2011). Dividing a 2D model domain into strips where the only inter-process communications are at the boundaries between two adjacent sub-domains is another approach (Tarboton et al., 2009c; Tesfa et al., 2011). A different approach, particularly useful for a model with a tightly coupled set of processes, is collecting all the governing equations into a global system of equations which are solved by a matrix solver (Qu and Duffy, 2007). Such a matrix solver may divide the global matrix into sub-matrices that are mapped to multiple processes. The examples above do not form an exhaustive list; however, they indicate that, generally, the choice of a given domain decomposition would be dictated by the specific modeling problem (Small et al., 2013). Presently, a researcher or a modeler has to consider their own problem domain and decide whether to use a method similar to those listed above or develop their own. The ability to automatically choose certain domain decomposition method given a problem domain type is a potential area of future study

The extent to which a parallel program's performance increases with the increase of computing resources (e.g., number of processes) depends primarily on the fraction of code that has to be executed in serial. This is the essence of Amdahl's law (Amdahl, 1967, 2007; Gustafson, 1988) based on which the maximum possible speed-up for a given problem is computed as a function of the number of processes and the serial fraction of the code. The effect of the serial fraction of the code becomes more important with increasing parallelization. Yavits et al. (2014) review a number of research works

that deal with the effect of data transfer between the serial and parallel portions of the code and the inter-process communications on the maximum speed-up determined from Amdahl's law. They provide models that revise Amdahl's law incorporating terms that represent arithmetic intensity—the ratio of total compute operations to data transfer computations, data transfer synchronization between the serial and parallel portions of the program, and inter-process communication and synchronization. Accordingly, even if a program has a parallelizable fraction close to 1, high inter-process communication and/or high serial to parallel data synchronization might render it unsuitable for extensive parallelization. For such problems, they suggest using fewer large cores rather than a large number of small cores.

Finally, the cost efficiency of the parallel method will have to be considered. As indicated earlier, Neal et al. (2010) concluded the developer time for programming using graphics cards to be prohibitive in their modeling case. However, total cost should also include the price of the computing hardware units and operating costs. Tristram et al. (2014) report results from a parallel hydrologic uncertainty model with multiple ensembles using GPUs. They compared the CPU and GPU performances with respect to speed-up, the cost of processors and the cost of power usage, and found GPUs to be more cost efficient for their application. Regarding programmer's time, they showed that to achieve satisfactory speed-up with GPUs, major refactoring of their existing code was not necessary. In addition, the performance was further enhanced with an optimization involving memory access configuration. The general purpose programming toolkits such as CUDA (<https://developer.nvidia.com/cuda-toolkit>) and OpenCL (<https://www.khronos.org/opencv/>) coupled with the cheap graphics cards on commodity

computers make GPU programming more accessible to scientific research programmers (Garland et al., 2008). However, realizing the full benefits still requires learning efficient program organization and optimizations such as latency hiding by overlapping computation with Input Output (IO) operations, wise management of register and caches, memory layout configuration (De La Asunción et al., 2012; Tristram et al., 2014) which requires more effort and time (Brodtkorb et al., 2013).

4.3 Methods

4.3.1 *Utah Energy Balance (UEB) Model*

The Utah Energy Balance (UEB) model tracks the accumulation and ablation of a single snow layer at the ground surface by energy and mass balance computations (Tarboton and Luce, 1996). The model accounts for canopy snow interception, partitioning of incoming solar and atmospheric radiation through canopy layer, and turbulent fluxes of latent and sensible heat within and below canopy layer (Mahat and Tarboton, 2012, 2013; Mahat et al., 2013). The snow surface temperature is computed using the modified Force-Restore method that characterizes the conduction of heat into the snowpack as a function of the temperature gradient between the snow surface and the average temperature of the snowpack, and by taking into account the temperature profile of the snowpack in the past 24 hours (Luce and Tarboton, 2010; You et al., 2014). Recently, glacier melt processes were modeled with UEB (Sen Gupta and Tarboton, 2013). The model equations and further descriptions can be found in previous publications (Mahat and Tarboton, 2012; Mahat et al., 2013; Tarboton and Luce, 1996).

In UEB, the equations describing the state-flux relationships are applicable for a

model element with uniform (or representative) values of terrain characteristics (slope, aspect, etc.), canopy variables, and meteorological forcing. For spatially distributed modeling, earlier research explored the use of depletion curves to deal with sub-watershed variability (Luce et al., 1999). The recent work by Sen Gupta and Tarboton (2013) configured the model to be run as fully distributed using Cartesian grids. In the gridded model, the model computations are carried out separately on individual grid cells with the only interaction between grid cells occurring when aggregating outputs from watersheds (or sub-watersheds). This configuration makes UEB amenable to parallel computation with domain decomposition that is to be constrained only by the aggregation operations.

The UEB model is driven by temperature, precipitation, radiation, humidity, and wind speed. The Network Common Data Form (NetCDF), a data format that enables storage and manipulation of multi-dimensional arrays (<http://www.unidata.ucar.edu/>) is used as input/output format for UEB. NetCDF includes a set of libraries and tools that enable array-oriented data access with advantages that include concurrent access (multiple readers), platform independence, efficient sub-setting, and data appending (i.e., additional data are added to a file without redefining it or copying the whole content). A NetCDF file is self-contained in that the metadata to describe the contents of the data and other ancillary information are stored within the file (Rew et al., 2014). A benefit of using NetCDF is that many array-oriented data sets come in some form of gridded binary format compatible with NetCDF. The choice to use NetCDF as input/output format for UEB was driven by the vision to couple the model with data sources and other models that follow the same standard.

4.3.2 *UEB Parallel*

The flow chart for the parallel version of the UEB model developed by this work with MPI implementation is shown in Figure 4.1. Many of the tasks, including the weather forcing IO operations, are contained in a code block named “Run UEB in the grid cell for all time steps.” This step loops through all grid cells, runs the model for all time steps of the simulation period for a given grid cell, proceeds to the next grid cell and runs the model for the whole simulation period, and so on. The operations at each grid cell are carried out independently from the other grid cells. This block of code takes more than 99% the total simulation time. This code block was, therefore, parallelized with MPI in which the active grid cells, excluding the no-compute cells, were divided into the total number of processes. When the total number of grid cells is not evenly divisible by the number of available processes, some processes may be allocated an extra grid, leaving the others with an idle time of one grid cell computation. For large sized problems, this idle time is expected to be small.

At the end of the simulation, the processes collectively write model outputs to a NetCDF file, one output file for each output variable. This NetCDF write requires synchronization among the processes as they access the NetCDF file simultaneously. One factor we evaluated in this study was the extent to which the IO operations can be parallelized, and the degree to which the synchronized access to data in NetCDF files by multiple processes can affect the overall performance of the parallel codes. For this implementation of the UEB model with MPI, we compared an IO reading/writing in which multiple arrays of data are handled at once to the ‘looping through the grids’ approach mentioned above that accesses a single array at a time (multiple arrays versus

one-array-at-a-time).

The flow chart for the model implementation with GPU is shown in Figure 4.2. In this case, a UEB class was defined first as a class that encapsulates watershed, terrain, and canopy variables as data members and the UEB simulation functions as methods (member functions). As the result, the array of active grid cells consists of an array of UEB class instances. This configuration was chosen because it was easier to copy arrays of objects/structures between the host/CPU and device/GPU nodes than copying individual variable arrays. The CPU (host) allocates GPU (device) memory and copies the data to the device. All the snow process computations are carried out by the GPU functions, i.e., kernels, and outputs are copied back to the CPU node that writes them into NetCDF files. In this case, in contrast to the MPI implementation, few CPU nodes carry out the IO operations. We implemented the CUDA code in such a way that the changes to the UEB MPI code were kept minimal, as can be seen from comparison of Figures 4.1 and 4.2. The objective here was to evaluate if the observation by Tristram et al. (2014) that implementing GPU code with satisfactory performance may not necessarily require major re-work on an existing programming code also applies to UEB.

An important difference between the GPU implementation and the one with MPI is that in the GPU case the time loop is outside of the grid cell loop, i.e., simulations at all grid cells are carried out for a few time steps (typically a few days) before advancing to the next step. This way, the highly parallel nature of individual grid cell computations is taken advantage of without having to copy all the weather forcing data to the device at once. Copying weather forcing data for all time steps at once would require large memory at the device that could be difficult to allocate.

In both implementations, the active grid cells were evenly distributed among MPI processes or GPU threads, when that was possible. In the UEB model, there is spatial variability in grid cell properties, such as vegetation covered versus no vegetation, snow covered versus no snow that affects the number of iterations to converge to a particular solution in a given cell, and that introduces some variability in the total compute time among processes/threads. This is considered to be part of simulation overhead and will diminish the efficiency, compared to the ideal case of both parallel implementations. Two other sources of overhead are reading configuration files at the beginning of the program and an outputs aggregation step where watershed average/total quantities are computed. The aggregation operations involve inter-process communication.

4.3.3 *Test Case Study: The Logan River Watershed*

We used simulations of snow accumulation and melt for five years—October 1, 2007- September 30, 2012—in the Logan River watershed, Utah, to evaluate the performance of the parallel codes. The Logan River watershed is a 554 km² watershed located in the Bear River Range of Utah in the western U.S. The watershed elevation ranges from 1497 to 3025 m with mean elevation of 2271 m. Most of the upland watershed is covered by shrubs, grass, and forest and is primarily used for grazing while the lower reaches of the river support irrigation. The average precipitation varies between 450 – 1500 mm per year with most of it in the form of snow. The river peaks late in the spring from snowmelt. Figure 4.3 shows the location map of the Logan River watershed and its digital elevation map.

The input data were setup as follows. The watershed domain was delineated from the 30 m USGS National Elevation Dataset Digital Elevation Model (NED DEM) using

the terrain analysis software TauDEM (Tarboton et al., 2009a; Tarboton, 2015; Tesfa et al., 2011). Terrain variables slope and aspect were calculated from the DEM in ESRI's ArcGIS software (www.esri.com). The canopy leaf area index (LAI) values were obtained from the Moderate Resolution Imaging Spectroradiometer (MODIS) product MOD15A2. The other vegetation variables including canopy height and fraction of land covered with vegetation were determined using the National Land Cover Database 2011 (NLCD 2011) (Homer et al., 2015). Weather forcing data were obtained from SNOTEL stations in and nearby the watershed. These data were gridded with the bilinear interpolation method to grid size of 120 m, the model resolution, and downscaled (adjusted for elevation) using a methodology described by Sen Gupta and Tarboton (2016). The focus of this study was evaluation of the performance of the parallel codes; thus, no calibration or validation of model parameters was performed apart from verification to make sure the parallel models' outputs are consistent with those of the original serial version.

4.3.4 *Description of Computing Resources*

Both versions of the UEB Parallel model were written as a platform independent code with the C++ programming language. The GPU version uses CUDA for the device codes. Performance tests were made in Linux clusters with up to 128 processes for the MPI version, while the GPU code was tested on a Linux machine with a CPU node linked to one GPU node. The specifications of the computing resources are as follows:

- Linux cluster for MPI: Intel(R) Xeon(R) CPU E5-4620 0 @ 2.20GHz with a total of 32 cores/node (64 cores/node with hyper-threading enabled).
- Linux nodes with GPU: Intel E5-2670 (Sandy Bridge) CPU + NVIDIA K20x GPU.

4.3.5 *Performance Metrics*

Total simulation time, speed-up, and efficiency were used to test the performance of the parallel models. Speed-up is computed as the ratio of the total simulation time by a single processor to that by multiple processors (P number of processors). Speed-up varies with the total of number of processors. Efficiency is the speed-up divided by the number of processors (Eager et al., 1989). For a given unit of work, the efficiency may change with the number of processors due to an increased overhead and/or inter-process communication. According to Amdahl's law (Amdahl, 1967, 2007; Gustafson, 1988) the maximum possible speed-up (ideal speed-up) is constrained by the fraction of the code that has to be executed serially, and hence is executed by all processors. For the UEB model, this was assumed to be the code outside of the "Loop through active grid cells" portion of the code described above. In reality, however, some of the codes inside the loop could also contribute to it. These are generally considered 'overhead,' and this is partly why the "ideal speed-up" is called "ideal"—because, in practice, the overheads further reduce the speed-up.

Equations 1 - 3 below give Speed-up, Efficiency, and representation of Amdahl's law, given a number of processors P. Equations 1 and 3 represent the ratio of two similar units, often computed as the simulation time per one processor divided by simulation time per multiple processors. Both equations (1 & 3) apply to processors that are of uniform type, of similar core size and speed. The units of the numerator and the denominator in equation 3 can thus be considered as that of time per unit computation core.

$$S_p = \frac{T_1}{T_p} \quad (1)$$

$$E_p = \frac{S_p}{P} \quad (2)$$

$$S_{pmax} = \frac{1}{f_s + \frac{1-f_s}{P}} \quad (3)$$

Where: S_p = Speed-up by P number of processors

T_1 = Execution time for a single processor

T_p = Execution time for P number of processors

E_p = Efficiency for P number of processors

P = number of processors

S_{pmax} = maximum speed-up based on Amdahl's law

f_s = Fraction of code that can only be executed serially.

In the case of UEB with MPI, with much of the code in the parallelizable “Loop through active grid cells” block as described earlier, speed-up approaching the number of processors (P) is to be expected given little input overhead and blocking communications (i.e., inter-process communications that force processes to wait for each other). Some degradation from maximum efficiency is expected due to variability in the processing time for each cell. An increase in the workload by increasing the size of the watershed and/or the duration of simulation would primarily increase the tasks inside the loop and is expected to increase the speed-up per total number of processes (i.e., efficiency).

When computing the ideal speed-up and efficiency using Amdahl's law above, we considered the time for the inter-process communications to be part of the fraction of the

code executed in serial, but we assumed IO operations were parallelizable. An alternative analysis is to consider the inter-process communications and IO operations separately. To demonstrate the effect of IO operations on the parallel performance of UEB, we use a slightly modified equation from (Yavits et al., 2014 p. 7 Eqn 13) for symmetric compute cores of uniform compute ability. The modification here is that we assume the inter-process communication to be negligible in UEB, hence the term for inter-process communication is dropped.

$$S_{pmaxr} = \frac{1}{(1 - f_p) + \frac{f_p}{P} + \frac{T_s}{T_1}} \quad (4)$$

Where: f_p = Fraction of code that can be executed in parallel

P = number of processors

T_1 = Execution time for a single processor

T_s = IO synchronization time (Sequential-to-parallel data synchronization time in (Yavits et al., 2014))

S_{pmaxr} = maximum speed-up based on Amdahl's law, revised to account for IO

The term $\frac{T_s}{T_1}$ in equation (4) is referred to as “Synchronization Intensity” in (Yavits et al., 2014)—it accounts for the time spent by reading/writing and sharing (synchronizing) data among processors. The effect of Synchronization Intensity is to decrease the ideal speed-up, and its importance increases for larger number of parallel cores.

4.4 Results and Discussion

Figure 4.4 shows plots of the total simulation time, speed-up, efficiency, and ratio

of IO time to total simulation time as a function of the number of processes for the simulation using MPI. It can be seen that the slope of the speed-up curve decreases with increase in number of processes. Figure 4.4b also compares the speed-up against the maximum speed-up based on Amdahl's law. Excluding the code inside the "loop through the grids" which also includes IO, the remaining part of the code takes less than 0.01% of the total simulation time. This initially suggested a highly parallelizable code, which led to expectation of speed-up close to the ideal speed-up. The actual speed-up curve, however, is much lower than the ideal speed-up curve, and its slope decreases with increasing number of processes.

Figures 4.4b and 4.4c also include speed-up and efficiency plots for the model computation core, i.e., excluding IO operations. As can be seen from the figures, the speed-up of the computational core is closer to the ideal speed-up. The reason for the poor performance of the total code compared to the computation core is that the IO is not as readily parallelizable as the rest of the code. For the serial version of the program, the IO takes about 1.7 % of the total execution time of the code. Because this fraction of code is not being parallelized, it affects the performance of the whole model with increasing importance as the number of processes is increased, as shown in Figure 4.4d.

The deviation of the computational core speed-up from the ideal line can be caused by any of, or a combination of, the factors that were considered "overhead" during the design in Section 4.3.2. In addition, after the total active grid cells were evenly divided between the processes, a few processes would be allocated one additional grid cell to simulate. This means some processes may have to stay idle for a duration of one grid cell computation. The time it takes for a full computation of one grid cell, on

average, is about 2.5 seconds.

Bridging the gap between the good scaling of the computational core and the poor scaling for the total model run caused by poor IO scaling is important as the IO starts to dominate with increasing processes so much so that beyond 64 processes, increasing the number of processes may not be justifiably useful. While the parallel NetCDF4, which is based on HDF5's MPI IO feature, would enable synchronized file access by multiple processes, efficient IO scaling requires coupling it with some file read strategies that take advantage of the synchronization (Chilan et al., 2006). Figure 4.5 shows the results for a modified UEB IO reading/writing in which multiple arrays of data were handled at once instead of the 'looping through the grids' approach used in Figure 4.4. As can be seen in Figure 4.5, the performance improves appreciably, particularly for the higher number of processes. For 64 processes, the speed-up increases from 24 to 42 while the efficiency increases from about 0.38 to 0.67. Similarly, for 128 processes, the speed-up and efficiency increase from 31 to 63 and 0.25 to 0.49 respectively. This approach would reduce the total file access operations; however, it may require larger memory per core to be effective.

Figure 4.6 is a re-plot of Figure 4.5a with the ideal speed-up revised according to equation (4) to account for the effect of IO operations. This figure indicates that any further performance improvement would only come from better IO parallelization.

The model run times for the GPU version with CUDA code are shown in Table 4.1. We do not have multiple simulations in GPU, so there are no comparable plots for GPU to those for MPI. In this case, the IO operations were carried out by the CPU (host) while the numerical simulations were performed by the GPU (device). In addition to IO

operations, data synchronization between the host and device is required. Inputs are read by the host, copied from the host to the device, and simulated outputs are copied back to the host, which writes out to NetCDF output files. Table 4.1 also includes run times from the MPI implementations in Linux cluster with one and 64 processes, as well as run times on desktop PC with a single processor. The run time on desktop PC with a single processor is equivalent to that of a serial code on desktop PC.

The GPU implementation presents a comparable (slightly better) speed-up when compared to the MPI code executed on the CPU cluster of 64 processes (Table 4.1). Our result leads us to a similar observation by Tristram et al. (2014) that implementing GPU code with a performance comparable to other parallel methods may not necessarily require a major re-work of an existing programming code. In our case, the compute core and much of the data partitioning code remained the same, as it was written in C++, which is compatible to CUDA specification (NVIDIA, 2015). Given the fact that GPUs provide superior performance per total resource cost (price of hardware + power usage, see e.g., <http://www.fmslib.com/mkt/gpus.html>), and considering the comparatively short developer time for some existing codes like UEB, it appears to be a worthwhile alternative to the MPI implementation.

The last column of Table 4.1 shows the speed-up of each implementation with respect to the run time on desktop PC with a single processor (the equivalent serial version). This is different from the speed-ups and efficiencies reported in Figures 4.4 to 4.6. While the numbers in Figures 4.4 – 4.6 help evaluate the performance of the parallel code with increasing processes, the speed-up values in Table 4.1 represent measures of the actual performance enhancement achieved by implementing concurrency

programming for utilizing clusters of CPU and/or GPU resources, compared to a serial code on a desktop PC (Brodtkorb et al., 2013). Thus, Table 4.1 shows that our parallel implementations help achieve speed-ups of 8 – 10 times over a serial code on a desktop PC. This represents the actual enhancement of hydrologic research due to improved access to high performance computation resources—they point to the actual reduction in time spent simulating a UEB model instance from using the Linux cluster or the GPU resources rather than the desktop PC. And, such reduction in modeling time facilitates the evaluation and adoption of physically based models like UEB in operational settings such as streamflow forecast where computational time can be critical. It is interesting to note from the first column of Table 1 that the “model setting” which involves reading model domain, terrain, and parameter files by all processes, serially, has a very large value for MPI with 64 processes. This large overhead is quite unexpected and contradicts our assumption earlier that the overhead gets negligible with increasing number of processes. Still, it does not change the conclusion about the overall speed-up.

4.5 Summary and Conclusions

The implementation and evaluation of parallel computation in a distributed snow accumulation and melt model UEB was presented in this paper. Two parallel implementations of UEB were evaluated: one using the Message Passing Interface (MPI) and the other using CUDA GPU. Continuous simulation of the Logan River watershed for five years was used to test the performance of the parallel codes, and the speed-up and efficiency as function of number of processes and plots of speed-up compared to the ideal speed-up based on Amdahl’s law were used as performance metrics of the parallel codes.

For the MPI implementation, results show the importance of input/output (IO) operations in the degradation of efficiency with increase in the number of processes. In the serial code, the IO accounts for about 1.7% of the code, the impact of which becomes more pronounced with increased number of processes. This was verified using the revised form of Amdahl's law (Yavits et al., 2014) that separately accounts for IO operations and inter-process communications. The plot of this revised form of Amdahl's law indicates further performance increase of the parallel code could only be possible with improving the performance of the IO operations. The performance of the MPI implementation improves when utilizing an IO strategy that reduces the number of file accesses by reading and writing multiple arrays of data in one go.

The CUDA GPU implementation achieves slightly better speed-up with one GPU node when compared with the MPI implementation executed using 64 processes. The GPU implementation was done without major refactoring of the original code as the simulation core and much of the data partitioning code remain the same. This shows that, for some models such as UEB, obtaining a CUDA GPU performance comparable to other parallel methods does not necessarily require a major re-work of an existing programming code. Given the fact that GPUs provide superior performance per total resource cost (price of hardware + power usage), this makes it a worthwhile alternative to the MPI implementation.

Overall our parallel implementations help achieve speed-ups of 8 – 10 times over a serial code on a desktop PC. This represents the actual enhancement to our hydrologic research due to improved access to high performance computation resources—they point to the reduction in time spent simulating a UEB model instance by using the Linux

cluster or the GPU resources instead of the desktop PC.

Most distributed physically based hydrological models are data intensive. This work demonstrates the importance of including IO operations within the parallelizable code section and using efficient IO handling together with distributed computing resources to do large-scale hydrological modeling. Efficient IO scaling requires adopting file read/write strategies that take advantage of parallel file access or separate files for different processes. The approach we used in this paper is a simple one, which involves reading and writing multiple arrays of data in one-step, and it could be limited by the availability of memory per a core. There is a need for more advanced IO strategy that also accounts for the available memory.

Finally, the modeling work presented here is only for the case of snow accumulation and melt. Outputs from this model are used as input to runoff and river routing models. We can qualitatively predict that coupling UEB to a runoff model would increase the arithmetic intensity because more computations would be done without significantly increasing the IO volume. Additional inter-process communications would be introduced, but would likely be smaller than the added arithmetic operations. Therefore, it would be interesting to extend this study to examine if a better efficiency may be achieved with the coupled model.

References

Amdahl, G.M., 1967. Validity of the single processor approach to achieving large scale computing capabilities, Proceedings of the April 18-20, 1967, spring joint computer conference. ACM, pp. 483-485.

- Amdahl, G.M., 2007. Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities, Reprinted from the AFIPS Conference Proceedings, Vol. 30 (Atlantic City, NJ, Apr. 18–20), AFIPS Press, Reston, Va., 1967, pp. 483–485, when Dr. Amdahl was at International Business Machines Corporation, Sunnyvale, California. Solid-State Circuits Society Newsletter, IEEE 12(3) 19–20.
- Brodtkorb, A.R., Hagen, T.R., Sætra, M.L., 2013. Graphics processing unit (GPU) programming strategies and trends in GPU computing. *Journal of Parallel and Distributed Computing* 73(1) 4–13.
- Brodtkorb, A.R., Sætra, M.L., Altinakar, M., 2012. Efficient shallow water simulations on GPUs: Implementation, visualization, verification, and validation. *Computers & Fluids* 55 1–12.
- Burnash, R., Singh, V., 1995. The NWS river forecast system-Catchment modeling. *Computer models of watershed hydrology*. 311–366.
- Chilan, C.M., Yang, M., Cheng, A., Arber, L., 2006. Parallel I/O performance study with HDF5, a scientific data package. *TeraGrid 2006: Advancing Scientific Discovery*.
- De La Asunción, M., Mantas, J.M., Castro, M.J., Fernández-Nieto, E.D., 2012. An MPI-CUDA implementation of an improved Roe method for two-layer shallow water systems. *Journal of Parallel and Distributed Computing* 72(9) 1065–1072.
- Eager, D.L., Zahorjan, J., Lazowska, E.D., 1989. Speedup versus efficiency in parallel systems. *Computers, IEEE Transactions on* 38(3) 408–423.
- Fowler, H., Blenkinsop, S., Tebaldi, C., 2007. Linking climate change modelling to impacts studies: recent advances in downscaling techniques for hydrological modelling. *International Journal of Climatology* 27(12) 1547–1578.
- Franz, K.J., Hogue, T.S., Sorooshian, S., 2008. Operational snow modeling: Addressing the challenges of an energy balance model for National Weather Service forecasts. *Journal of Hydrology* 360(1) 48–66.
- Garland, M., Le Grand, S., Nickolls, J., Anderson, J., Hardwick, J., Morton, S., Phillips, E., Zhang, Y., Volkov, V., 2008. Parallel computing experiences with CUDA. *IEEE micro*(4) 13–27.
- Gropp, W., Lusk, E., Ashton, D., Balaji, P., Buntinas, D., Butler, R., Chan, A., Goodell, D., Krishna, J., Mercier, G., 2005. *MPICH2 user's guide*. Mathematics and Computer Science Division-Argonne National Laboratory, Version 0.4.
- Gustafson, J.L., 1988. Reevaluating Amdahl's law. *Communications of the ACM* 31(5) 532–533.
- Homer, C.G., Dewitz, J.A., Yang, L., Jin, S., Danielson, P., Xian, G., Coulston, J., Herold, N.D., Wickham, J., Megown, K., 2015. Completion of the 2011 National Land Cover Database for the conterminous United States-Representing a decade of land cover change information. *Photogrammetric Engineering and Remote Sensing* 81(5) 345–354.
- Kauffeldt, A., Wetterhall, F., Pappenberger, F., Salamon, P., Thielen, J., 2014. Technical review of large-scale hydrological models for implementation in operational flood forecasting schemes on continental level.

- Kollet, S.J., Maxwell, R.M., Woodward, C.S., Smith, S., Vanderborght, J., Vereecken, H., Simmer, C., 2010. Proof of concept of regional scale hydrologic simulations at hydrologic resolution utilizing massively parallel computer resources. *Water Resources Research* 46(4).
- Levine, J.B., Salvucci, G.D., 1999. Equilibrium analysis of groundwater–vadose zone interactions and the resulting spatial distribution of hydrologic fluxes across a Canadian Prairie. *Water Resources Research* 35(5) 1369-1383.
- Li, T., Wang, G., Chen, J., Wang, H., 2011. Dynamic parallelization of hydrological model simulations. *Environmental Modelling & Software* 26(12) 1736-1746.
- Liu, J., Zhu, A., Liu, Y., Zhu, T., Qin, C.-Z., 2014. A layered approach to parallel computing for spatially distributed hydrological modeling. *Environmental Modelling & Software* 51 221-227.
- Luce, C.H., Tarboton, D.G., 2010. Evaluation of alternative formulae for calculation of surface temperature in snowmelt models using frequency analysis of temperature observations. *Hydrology and Earth System Sciences* 14(3) 535-543.
- Luce, C.H., Tarboton, D.G., Cooley, K.R., 1999. Sub-grid parameterization of snow distribution for an energy and mass balance snow cover model. *Hydrological Processes* 13(12) 1921-1933.
- Mahat, V., Tarboton, D.G., 2012. Canopy radiation transmission for an energy balance snowmelt model. *Water Resources Research* 48(1) W01534.
- Mahat, V., Tarboton, D.G., 2013. Representation of canopy snow interception, unloading and melt in a parsimonious snowmelt model. *Hydrological Processes* n/a-n/a.
- Mahat, V., Tarboton, D.G., Molotch, N.P., 2013. Testing above- and below-canopy representations of turbulent fluxes in an energy balance snowmelt model. *Water Resources Research* 49(2) 1107-1122.
- Maxwell, R.M., Putti, M., Meyerhoff, S., Delfs, J.O., Ferguson, I.M., Ivanov, V., Kim, J., Kolditz, O., Kollet, S.J., Kumar, M., 2014. Surface - subsurface model intercomparison: A first set of benchmark results to diagnose integrated hydrology and feedbacks. *Water Resources Research* 50(2) 1531-1549.
- Neal, J.C., Fewtrell, T.J., Bates, P.D., Wright, N.G., 2010. A comparison of three parallelisation methods for 2D flood inundation models. *Environmental Modelling & Software* 25(4) 398-411.
- Nickolls, J., Buck, I., Garland, M., Skadron, K., 2008. Scalable Parallel Programming with CUDA. *Queue* 6(2) 40-53.
- NVIDIA, 2015. *Cuda C Programming Guide*.
- Paglieri, L., Ambrosi, D., Formaggia, L., Quarteroni, A., Scheinine, A.L., 1997. Parallel computation for shallow water flow: A domain decomposition approach. *Parallel computing* 23(9) 1261-1277.
- Paniconi, C., Putti, M., 2015. Physically based modeling in catchment hydrology at 50: Survey and outlook. *Water Resources Research* 51(9) 7090-7129.
- Qu, Y., Duffy, C.J., 2007. A semidiscrete finite volume formulation for multiprocess watershed simulation. *Water Resources Research* 43(8).
- Rao, P., 2004. A parallel hydrodynamic model for shallow water equations. *Applied mathematics and computation* 150(1) 291-302.

- Rew, R., Davis, G., Emmerson, S., Davies, H., Hartnett, E., Heimbigner, D., Fisher, W., 2014. NetCDF Documentation (<http://www.unidata.ucar.edu/software/netcdf/docs/>). Unidata, University Corporation for Atmospheric Research (UCAR) Community Programs (UCP).
- Sanders, B.F., Schubert, J.E., Detwiler, R.L., 2010. ParBreZo: A parallel, unstructured grid, Godunov-type, shallow-water code for high-resolution flood inundation modeling at the regional scale. *Advances in Water Resources* 33(12) 1456-1467.
- Sen Gupta, A., Tarboton, D.G., 2013. Using the Utah Energy Balance Snowmelt model to quantify snow and glacier melt in the Himalayan region.
- Small, S.J., Jay, L.O., Mantilla, R., Curtu, R., Cunha, L.K., Fonley, M., Krajewski, W.F., 2013. An asynchronous solver for systems of ODEs linked by a directed tree structure. *Advances in Water Resources* 53 23-32.
- Sutter, H., 2005. The free lunch is over: A fundamental turn toward concurrency in software. *Dr. Dobbs' journal* 30(3) 202-210.
- Sutter, H., Larus, J., 2005. Software and the concurrency revolution. *Queue* 3(7) 54-62.
- Tarboton, D.G., Luce, C.H., 1996. Utah energy balance snow accumulation and melt model (UEB). Citeseer.
- Tarboton, D.G., Schreuders, K., Watson, D., Baker, M., 2009. Generalized terrain-based flow analysis of digital elevation models, *Proceedings of the 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation*, Cairns, Australia, pp. 2000-2006.
- Tarboton, D.G., 2015. *Terrain Analysis Using Digital Elevation Models (TauDEM)*. Utah Water Research Laboratory, Utah State University, Logan, Utah.
- Tesfa, T.K., Tarboton, D.G., Watson, D.W., Schreuders, K.A., Baker, M.E., Wallace, R.M., 2011. Extraction of hydrological proximity measures from DEMs using parallel processing. *Environmental Modelling & Software* 26(12) 1696-1709.
- Tristram, D., Hughes, D., Bradshaw, K., 2014. Accelerating a hydrological uncertainty ensemble model using graphics processing units (GPUs). *Computers & Geosciences* 62 178-186.
- Winsemius, H., Van Beek, L., Jongman, B., Ward, P., Bouwman, A., 2013. A framework for global river flood risk assessments. *Hydrology and Earth System Sciences* 17(5) 1871-1892.
- Wood, E.F., Roundy, J.K., Troy, T.J., van Beek, L.P.H., Bierkens, M.F.P., Blyth, E., de Roo, A., Döll, P., Ek, M., Famiglietti, J., Gochis, D., van de Giesen, N., Houser, P., Jaffé, P.R., Kollet, S., Lehner, B., Lettenmaier, D.P., Peters-Lidard, C., Sivapalan, M., Sheffield, J., Wade, A., Whitehead, P., 2011. Hyperresolution global land surface modeling: Meeting a grand challenge for monitoring Earth's terrestrial water. *Water Resources Research* 47(5) W05301.
- Yavits, L., Morad, A., Ginosar, R., 2014. The effect of communication and synchronization on Amdahl's law in multicore systems. *Parallel computing* 40(1) 1-16.
- You, J., Tarboton, D., Luce, C., 2014. Modeling the snow surface temperature with a one-layer energy balance snowmelt model. *Hydrology and Earth System Sciences* 18(12) 5061-5076.

Table 4.1 Run times (seconds) and speed-up for different computing resources

Computing Resource	Model setting	Input reading	Output Writing	Computation	Host-Device data copy	Total *	Speed-up **
1CPU + 1GPU on Linux cluster	1.2	168.6	43.3	2321.5	109.6	2644.2	10.1
64 MPI processes on CPU Linux cluster	210.2	157.8	595.7	2354.9	NA	3318.7	8.0
1 MPI process on CPU Linux cluster	1.0	21.1	134.4	141730.0	NA	141886	0.2
1 process on desktop PC (CPU)	0.1	321.4	451.7	25812.3	NA	26586	1.0

* Including overhead

**Compared to a single desktop PC CPU

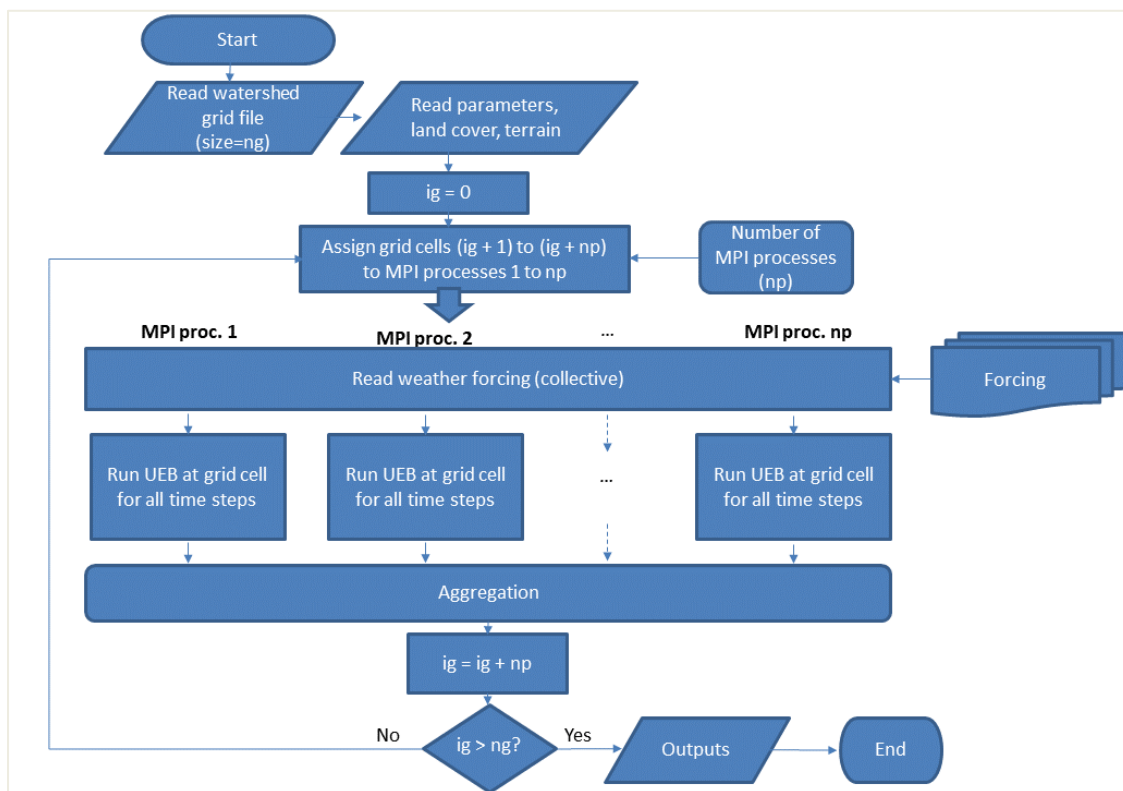


Figure 4.1 Flow chart for the parallel version of the UEB model (UEB Parallel) with MPI.

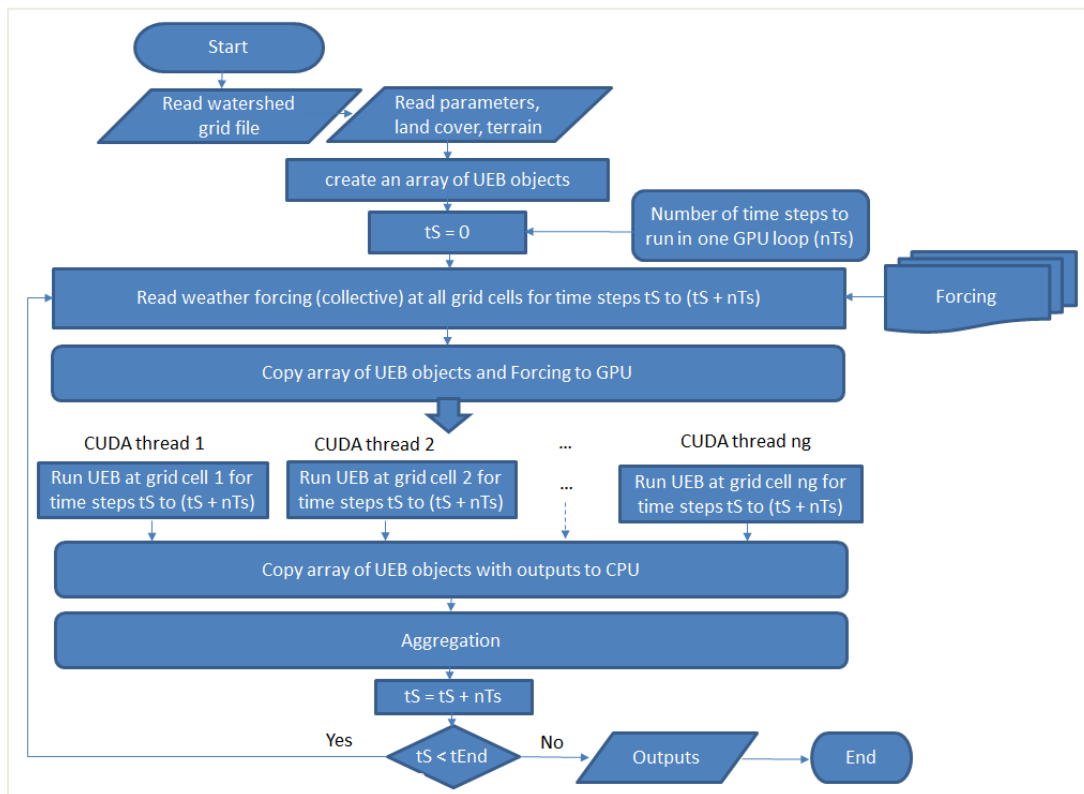


Figure 4.2 Flow chart for the parallel version of UEB model (UEB Parallel) with GPU.

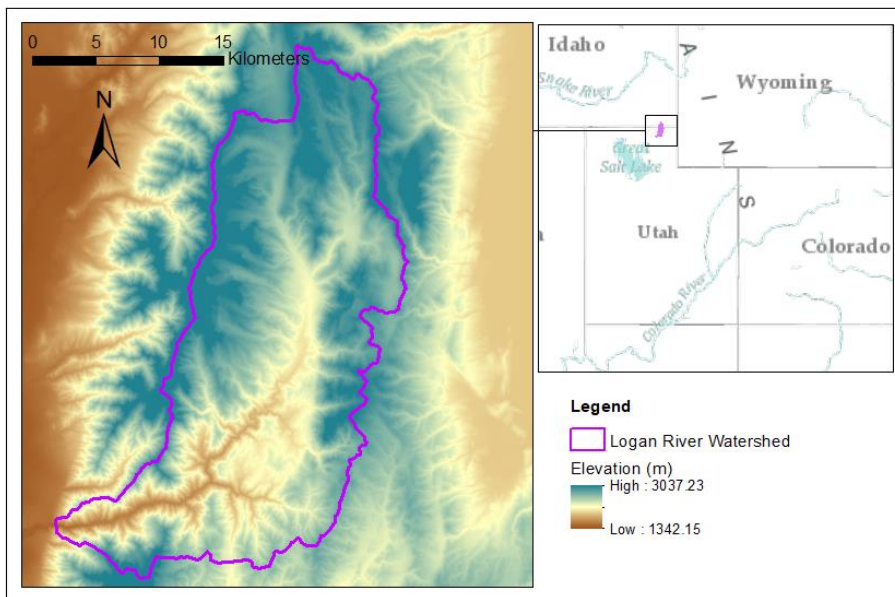


Figure 4.3 Study area: Location map of the Logan River watershed and its elevation (DEM).

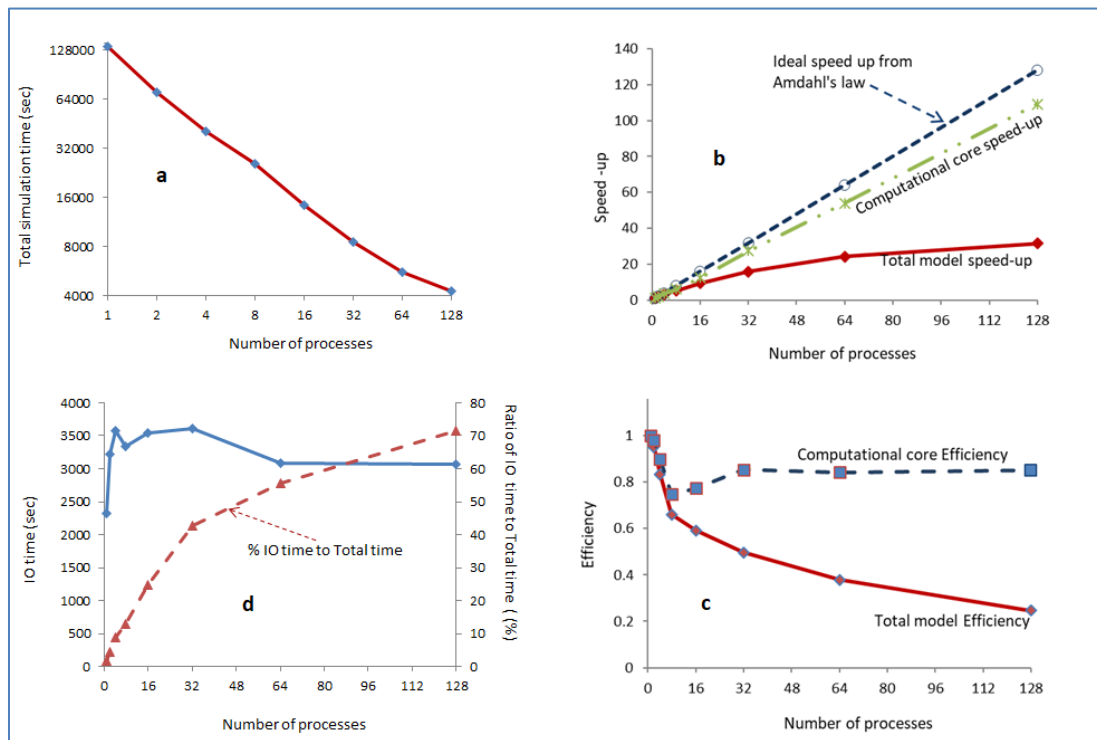


Figure 4.4 Total simulation time (a), speed-up (b), efficiency (c), and ratio of IO time to total time (d) vs the number of processes for the test in Linux Cluster with MPI.

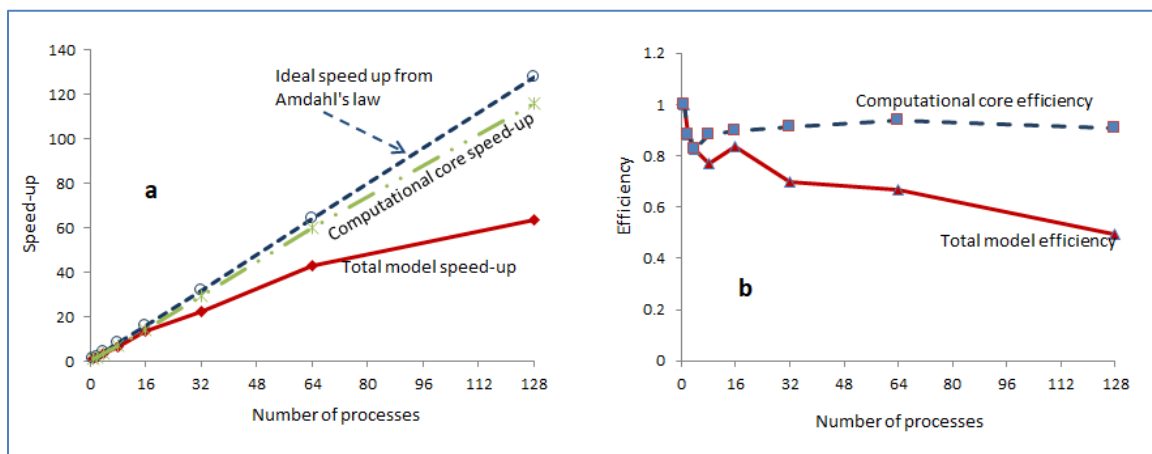


Figure 4.5 Speed-up (a) and efficiency (b) vs the number of processes for the test in Linux Cluster with MPI and reading multiple arrays at a time.

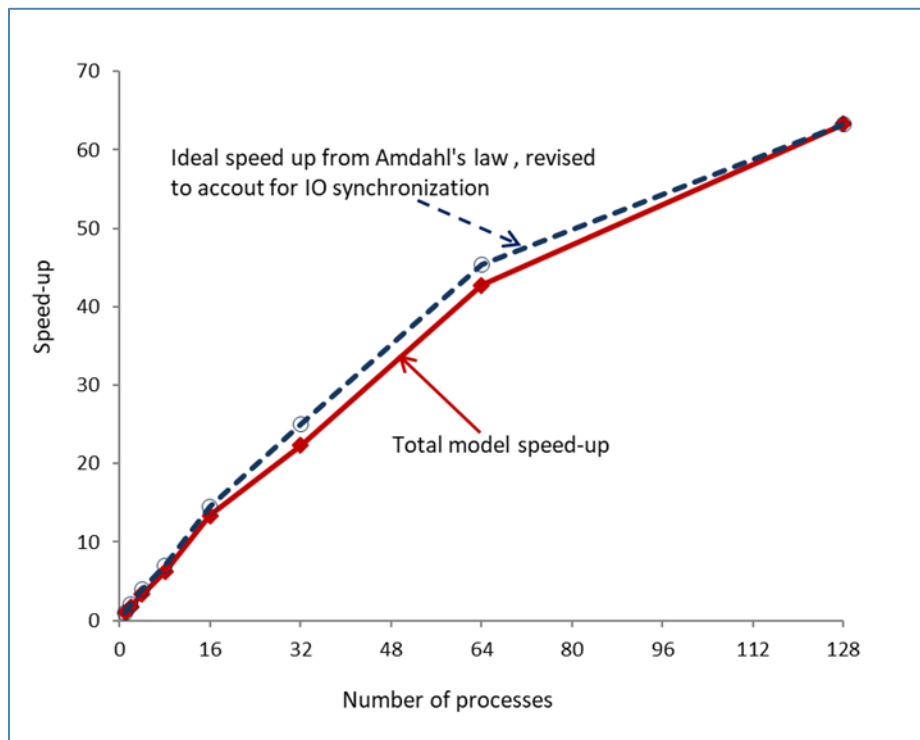


Figure 4.6 Speed-up and ideal speed-up from Amdahl's law revised to account for IO operation.

CHAPTER 5

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.1 Summary and Conclusions

This dissertation evaluated opportunities to enhance the application of an energy balance snowmelt model for streamflow forecasting. Chapters 2 presented work to assimilate snow and streamflow observations into the model to arrive at better snow and soil moisture states at forecast date to get an improved streamflow forecasts. Chapters 3 and 4 involved the development and evaluation of Cyberinfrastructure resources to address issues that often present an impediment to the application of detailed, physically based models in hydrologic modeling environment. The Cyberinfrastructure study focused on the design, implementation, and evaluation of a set of web-based, hydrological data processing services on one hand, and implementation and evaluation of parallel versions of the Utah Energy Balance snowmelt model (UEB) on the other.

The work in Chapter 2 introduces an integrated modeling framework that coupled the UEB model to the Research Distributed Hydrologic Model (RDHM) incorporating data assimilation and ensemble streamflow forecasting. This was to evaluate an energy balance snowmelt model in a framework similar to the National Weather Service (NWS) streamflow forecasting system. This integrated framework with an energy balance snowmelt model and data assimilation contributes towards the examination of more physically based models for prediction of watershed snowpack states and streamflow outputs. Such evaluations enhance the incremental adoption of physically based models in streamflow forecast operationally.

First, I evaluated assimilation of snow water equivalent (SWE) data from SNOTEL sensor networks into the UEB model using the Ensemble Kalman Filter (EnKF) to get improved snowpack states in the distributed model. The ensemble runs were carried out using forcing perturbations that account for spatial correlation through an exponential function decreasing with distance. By using a 'leave-one-out' approach, I evaluated the extent to which the simulated SWE at a given point is improved through assimilation of SWE at a SNOTEL station away from the grid point. Results showed that due to the covariance between the SWE at different grid cells, observations at one point can be used to update SWE at a different grid cell. This enabled assimilation of SWE data from sparse SNOTEL stations to be used to update the distributed SWE over the whole watershed. The improved snowpack states were then expected to result in improved snowmelt driven streamflow at the watershed outlet. The assimilation performance better during accumulation than ablation. This was likely due to the spatial correlations that resulted in data at one point having predictive capability at another point during accumulation diminishing during the energy driven melt process caused by the spatial variability in elevation.

In addition, I evaluated ensemble streamflow forecasting using the coupled UEB+RDHM models with assimilation of snow and streamflow data prior to the forecast date April 1 of the water year. The goal of the data assimilation prior to streamflow forecast was to arrive at the best possible model states on forecast time (April 1) so that the uncertainty in the forecast streamflow were reduced. Results showed that the ensemble streamflow forecast with assimilation of SWE provides marked improvement over the same forecast with no data assimilation. On the other hand, the assimilation of

streamflow using the Particle Filter to update the SAC-SMA and rutpix7 states did not provide any improvement beyond that achieved through the snow data assimilation. Results also showed that the ensemble forecast mean provides a good estimate of the mean volume of water for water supply forecast, but its smooth shape misses the peaks of the hydrographs, implying that the forecasts were not able to capture timing and magnitude of the flood hydrograph. While the ensemble of historical forcing may capture the likely range of forecast weather conditions, forecasting a flood hydrograph requires a forecast of the specific weather for that year.

In Chapter 3, HydroDS, a set of web-based data services providing access to hydrologic data and geospatial analysis capabilities for distributed hydrological models was introduced. The services comprise functions for important hydrologic data processing tasks such as watershed delineation, terrain processing, estimation of canopy variables based on the NLCD, and accessing and processing of climate data from Daymet and NLDAS. The services are composed of single task functions that can be used independently, or can be chained together to form a workflow for complete generation of model inputs. A platform independent Python client library provides an interface to the web services. Through the Python client library, the services can be used in a Python script or desktop application that obtains processed data from HydroDS and performs further analysis.

HydroDS was demonstrated by setting up UEB model instances for watersheds in the Colorado River and Great Salt Lake basins. The cases demonstrate how HydroDS helps reduce the time and effort spent by researchers for discovering, accessing, and pre-processing hydrologic model input data. A considerable part of the time saved by using

HydroDS instead of desktop-based data processing comes from better organization of data in HydroDS. These results are significant because the ability to access and configure input data could enhance or hinder the application of distributed, physically based hydrologic models, particularly in operational contexts where input preprocessing time could be critical, i.e., there is no the luxury of getting reanalysis data. And, the efforts to evaluate and incorporate energy balance snowmelt model in streamflow forecasts, presented in Chapter 2, need to be coupled with enhancing the capability to efficiently handle the data required for forecast watersheds, by taking advantage of Cyberinfrastructure resources that are being increasingly available to everyday modelers. In addition to reducing the time and effort required to prepare model inputs, it was shown that the Python workflow script maintains the provenance of the data processing and enhance repeatability and reproducibility. The script needs to be modified only to specify the watershed boundaries and outlet location when used for a different watershed. The ‘software as a service’ paradigm of the web services allows multiple users not to bother about the storage and organization of data, which is done in the server, and software and hardware dependencies are already sorted out.

The implementation and evaluation of two parallel computation methods in a distributed snowmelt model UEB was presented Chapter 4. The first implementation used the Message Passing Interface (MPI) and the other uses CUDA GPU. For the MPI implementation, results showed the importance of input/output (IO) operations in the degradation of efficiency with increasing the number of processes. In the serial code, the IO accounted for about 1.7% of the code, the impact of which becomes more pronounced with increased number of processes. This was verified using the revised form of

Amdahl's law that separately accounts for IO operations and inter-process communications. The plot of this revised form of Amdahl's law indicates further performance increase of the parallel code could only be possible with improving the performance of the IO operations. The performance of the MPI implementation improved when utilizing an IO strategy that reduced the number of file accesses by reading and writing multiple arrays of data in one go.

The CUDA GPU implementation achieved comparable speed-up using one GPU node when compared to that of the MPI implementation with 64 processes. The GPU implementation was done without major refactoring of the original code as the compute core and much of the data partitioning code remained the same as the MPI implementation. This shows that, for some models such as UEB using CUDA GPU, obtaining performance comparable to other parallel processing methods does not necessarily require major re-work on an existing programming code. Given the fact that GPUs provide superior performance per total resource cost (price of hardware + power usage), this makes it a worthwhile alternative to the MPI implementation.

Overall, the parallel implementations helped achieve speed-ups of 8 – 10 times over a serial code on a desktop PC. This represents the actual enhancement to our hydrologic research due to improved access to high performance computation resources, i.e., they point to the reduction in time spent running a UEB model instance by using the Linux cluster or the GPU resources instead of the desktop PC. Such reductions in simulation time facilitate not only the application of energy balance models that require more computational resources than the temperature index snowmelt models currently in use in NWS forecast centers, but also enable adoption of the computationally intensive

ensemble/particle methods that are the basis of the data assimilation approaches presented in Chapter 2.

5.2 Recommendations

The approach used for snow data assimilation in Chapter 2, i.e., sparse SNOTEL data assimilated over the whole watershed, depends on the existence of correlation between the ensemble SWE at a point with observations and those at grid cells without observations. The model ensembles were generated in this study from perturbation of weather forcing. The perturbed forcing errors were assumed to have a spatial correlation that was represented by an exponential decay function of the distance between grid cells. The correlation length on which this function depends was model input and was specified through trial and error. Further study is required for accurate estimation of the correlation length and its relation to the distance between the SNOTEL stations in a given watershed and other factors such as the effects of topography on precipitation shadows, ridges, and wind shadows that in turn affect the spatial correlation of snow. The variance in precipitation and other forcing errors was also input to the model. Examination of the uncertainty in precipitation and its statistical characteristics such as the error standard deviation may be an important further study to tune the results. Moreover, in addition to snow mass, data related to snow energy need to be assimilated to improve the melting season outputs. Remote sensing data are available for variables such as snow surface temperature, but the assimilation of such data may require a model reformulation that takes into account the effect of such factors as strong non-linearity and threshold behavior of snowmelt processes.

The parameters used for the generation of ensembles from forcing were arrived through trial and error, and they indicate that the forcing errors have weak temporal correlations, and the correlations among the errors in different forcing variables do not appear to be significant. However, a more systematic study to establish the importance or non-importance of temporal correlation and correlations among errors in forcing in the generation of the ensembles is required. There are multiple possible reasons for why the PF Q did not add value beyond that of the EnKF SWE, including insufficient particle size, inadequate representation of the impact of lag between soil moisture and streamflow, and the narrow range of uncertainty in soil moisture compared to the distribution of SWE ensemble. It may also be due to the dominant predictor for streamflow being the amount of snow, and prior to the forecast date (April 1) there has been limited snowmelt. Pinpointing the exact reason requires further investigation.

Given the high computational demand by the large number of model realizations required for the Particle Filter, and given that it did not result in appreciable improvement in the forecasted streamflow, it would not be feasible to apply in operational forecasting. However, the coupled UEB+RDHM model introduced in this work is expected to serve as a first step for further investigation.

The Python Client Library used to access individual functions of HydroDS, introduced in Chapter 3, has interface definitions to each atomic HydroDS function. In using these, however, it was common to sequence these as components of a workflow. It may improve efficiency to have this sequencing implemented on the server. The work in Chapter 3 dealt with large basins such as the Colorado River Basin by breaking them down into CBRFC forecast watersheds and handling data processing for smaller,

individual watersheds. This was a design choice that worked well for this study. Future studies should address the alternative approach of processing river basins such as the Colorado Basin as a whole. Future work should also extend the services to provide inputs for unstructured grid models and models using HRUs or other equivalent tessellations of the landscape for HydroDS to support a wider range of hydrologic models. Developments should also consider a simplified way to extend the services through provision of a Software Development Kit (SDK).

The work in Chapter 4 demonstrates the importance of including IO operations within the parallelizable code section and using efficient IO handling methods together with distributed computing resources. Efficient IO scaling requires adopting file read/write strategies that take advantage of parallel file access or separate files for different processes. The approach used in the paper is a simple one, which involves reading and writing multiple arrays of data in one-step, and it could be limited by the availability of memory per core. There is a need for a more advanced IO strategy that also accounts for the available memory. Coupling the UEB snowmelt model to a runoff model would increase the arithmetic intensity because more computations would be done in a grid cell without significantly increasing the IO volume. Additional inter-process communication would be introduced but would likely be smaller than the added arithmetic operations. Therefore, it would be interesting to extend this study to examine if a better scalability may be achieved with the coupled model.

APPENDICES

Appendix: UEB Land Cover Variables Lookup Table based on NLCD

Description	NLCD Class	cc*	hcan*	LAI*	ycage*
Open Water	11	0	0	0	2
Perennial Ice/Snow	12	0	0	0	2
Developed, Open Space	21	0	0	0	2
Developed, Low	22	0	0	0	2
Developed, Medium Intensity	23	0	0	0	2
Developed High Intensity	24	0	0	0	2
Barren Land (Rock/Sand/Clay)	31	0	0	0	2
Deciduous Forest	41	0.5	8	1	2
Evergreen Forest	42	0.7	15	4.5	3
Mixed Forest	43	0.8	10	4	2
Dwarf Scrub	51	0	0	0	2
Shrub/Scrub	52	0.5	3	1	2
Grassland/Herbaceous	71	0	0	0	2
Sedge/Herbaceous	72	0	0	0	2
Lichens	73	0	0	0	2
Moss	74	0	0	0	2
Pasture/Hay	81	0	0	0	2
Cultivated Crops	82	0	0	0	2
Woody Wetlands	90	0.5	8	1	2
Emergent Herbaceous Wetlands	95	0	0	0	2

* cc = Fraction of canopy cover
 hcan = Canopy height
 LAI= Leaf area index
 ycage = Forest canopy structure flag

CURRICULUM VITAE**Tseganeh Zekiewos Gichamo****E-mail:zacctsega@gmail.com****EDUCATION**

PhD, Civil and Environmental Engineering, Utah State University, Logan, Utah. March 2019. Dissertation Title: Advancing Streamflow Forecasts through the Application of a physically based Energy Balance Snowmelt Model with Data Assimilation and Cyberinfrastructure Resources.

MSc, Water Science Engineering, Specialization Hydroinformatics, UNESCO-IHE Institute for Water Education, Delft, The Netherlands. 2008. Thesis: Embankment Breach Modeling using the HR BREACH Model.

BSc, Civil Engineering, Mekelle University, Mekelle, Ethiopia. 2003.

RESEARCH INTERESTS

- Surface water hydrology, open channel flow, environmental hydraulics.
- Data assimilation, snowmelt modeling, streamflow forecast.
- Hydroinformatics, Geographic Information Systems (GIS).
- Dam breach/break modeling, Flood forecasting.
- Numerical methods in environmental hydraulics and hydrology.

PUBLICATIONS

T. Z. Gichamo, A. Jonoski, I. Popescu, M. Morris, M. Hassan. (2013)
Embankment Failure Modeling using the HR BREACH Model.

Environmental Engineering and Management Journal, Vol.12, No. 5, 865-874.

T. Z. Gichamo, I. Popescu, A. Jonoski, D. Solomatine. (2012) River Cross Section Extraction from the ASTER Global DEM for Flood Modelling. Environmental Modelling & Software 31 (2012) 37-46, Elsevier; DOI:10.1016/j.envsoft.2011.12.003.

T. Z. Gichamo., A. Jonoski, I. Popescu. (2010) Web Based Demonstrator for the HR BREACH Model. Proc. 9th Intern. Conf. on Hydroinformatics, Tianjin, China, September 2010.

A. D. Santos Almoradie, A. Jonoski, Y. Xuan, T. Gichamo, D.P. Solomatine, J. De Ruiter. (2010) Web-based solutions for flood risk analysis, modelling and management. Proc. 9th Intern. Conf. on Hydroinformatics, Tianjin, China, September 2010.

WORK/RESEARCH EXPERIENCE

July 2008, July 2009, July 2010 Tutor, Integration of Models (UNESCO-IHE, Institute for Water Education, Delft, The Netherlands)

January 2009 – December 2010 Research Fellow (UNESCO-IHE, Institute for Water Education, Delft, The Netherlands)

June 2008 Code Developer (UNESCO-IHE, Institute for Water Education, Delft, The Netherlands)

October 2007 – December 2008 Research Fellow (UNESCO-IHE, Institute for Water Education, Delft, The Netherlands)

August 2005 – October 2006 Graduate Assistant (Technology Faculty,

Hawassa University, Awassa, Ethiopia)

June 2004 – August 2005 Junior Contract Engineer (Contract Implementation Section, Ethiopian Roads Authority, Addis Ababa, Ethiopia)

November 2003 – June 2004 Junior Structural Engineer (Design Department, Water Works Design and Supervision Enterprise, Addis Ababa, Ethiopia)

PROGRAMMING LANGUAGES

C++, Python, R, Pascal (Free Pascal or Borland Delphi Pascal), CUDA, MPI, FORTRAN

SOFTWARE

HEC-RAS, HEC-HMS, MikeSHE, Mike11, MODEFLOW, HR-BREACH, SWAT, UEB, GDAL, Arc-GIS, QGIS, Access, MySQL

REFERENCES

Dr. David G. Tarboton. Professor of Hydrology, Department of Civil and Environmental Engineering, Utah State University, Logan, UT 84321. E-mail: dtarb@usu.edu

Dr. Jeffery S. Horsburgh. Associate Professor, Department of Civil and Environmental Engineering, Utah State University, Logan, UT 84321. E-mail: jeff.horsburgh@usu.edu

Dr. David E. Rosenberg. Associate Professor, Department of Civil and Environmental Engineering, Utah State University, Logan, UT 84321. E-mail: david.rosenberg@usu.edu

Dr. Ioana I. Popescu. Associate professor of Hydroinformatics, IHE Delft
Institute for Water Education, PO Box 3015, 2601 DA Delft , The
Netherlands. E-mail: i.popescu@unesco-ihe.org

Dr. Solomon D. Seyoum. Lecturer in Drainage and Sewerage, IHE Delft
Institute for Water Education, PO Box 3015, 2601 DA Delft , The
Netherlands. E-mail: s.seyoum@unesco-ihe.org