The Thesis Committee for Leah Gage Huling

Certifies that this is the approved version of the following thesis:

Streamflow and precipitation data assimilation into the National Water Model: an investigative study for statewide flood forecasting

APPROVED BY

SUPERVISING COMMITTEE:

Paola Passalacqua, Supervisor

David Maidment

Streamflow and precipitation data assimilation into the National Water Model: an investigative study for statewide flood forecasting

by

Leah Gage Huling

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ENGINEERING

The University of Texas at Austin

August 2020

Acknowledgments

First and foremost, I would like to thank my advisor, Paola Passalacqua, for her constant guidance and support throughout the last two years. She is the best advisor and mentor I could ask for, and her creative vision was always there for me when I stalled. Dr. Passalacqua has fostered an incredible research team that I feel grateful to have been a part of. I am also grateful to David Maidment for his enthusiasm and expert knowledge. He is a driving force who fosters inspiration in those around him and never wavers in his pursuit of groundbreaking research. Sincere gratitude to Dirk Schwanenberg for taking me under his wing during my internship at KISTERS headquarters in Aachen, Germany. He was an invaluable guide in the field of data assimilation and in the region of western Germany.

Thanks to the flood research team at the Center for Water and the Environment (CWE). I would like to especially thank Tim Whiteaker for his expertise in all things coding and David Arctur for being a quick email away with questions about data archives, sensor availability, and really anything else.

Thanks to all current and former members in the PassaH20 research group for all of the memories and advice. I absolutely could not have done this without you. A special thanks to my parents and sisters for the nonstop support and encouragement. You all are the best support system anyone could ask for. A final thanks to Ty Genard for always being there for me when grad school felt overwhelming and I needed a calm voice to steady me.

Lastly, I would like to dedicate this thesis to my dad, who has been my inspiration from day one. From high school science fairs to my master's thesis, you are my number one motivator and my rock.

This thesis was financially supported by TxDOT project number 5-9054-01 *Streamflow Measurement at TxDOT Bridges* (Maidment et al., 2019). My summer internship at KISTERS A.G. in Aachen, Germany was sponsored by CWE and by KISTERS.

Streamflow and precipitation data assimilation into the National Water Model: an investigative study for statewide flood forecasting

Leah Gage Huling, M.S.E.

The University of Texas at Austin, 2020

Supervisor: Paola Passalacqua

Flooding is a dangerous natural disaster that poses risk to property and personal safety world-wide. The state of Texas is especially vulnerable to floods; a region known as "Flash Flood Alley" runs through central Texas, and the coastal region is prone to severe tropical storms and hurricanes. Flood warning systems exist in large cities throughout Texas, but there is no coordinated state-wide flood warning network. In this study, we investigate the feasibility of creating a locally intelligent, state-wide flood forecasting system by using local observational streamflow data to better inform a national forecasting model. A method is developed to integrate streamflow sensors and precipitation products into short-range, 18-hour National Water Model (NWM) forecasts through data assimilation (DA). Four-dimensional variational data assimilation is coupled with a mass-conservative Muskingum-Cunge flow routing scheme to propagate streamflow corrections upstream and downstream from the sensor locations. The model is applied to a rural study area in the Texas Hill Country, and the streamflow DA model creates improved forecasts products at the sensor location and over 15 miles upstream and downstream of the sensor. Proof of concept results from a simplified surface runoff model using precipitation corrections indicate improved streamflow profiles at the

upstream location. With further validation and development, there is real potential in assimilating local data into the NWM to create a statewide flood forecasting network.

Keywords: flooding, streamflow data assimilation, variational data assimilation, Muskingum-Cunge, flood warning system

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Motivation

Flooding is often considered the most devastating natural disaster in the world. It is the deadliest severe weather hazard in the United States, with a 10-yr average of over 90 fatalities a year (NWS, 2018). Floods are also the costliest natural disaster in the United States, causing almost $5 billion in damages annually over the last 40 years, even excluding damage caused by tropical storms/cyclones (National Centers for Environmental Information, 2020). Fatalities and damage costs due to flooding are exacerbated and continue to increase as extreme rainfall events become more intense and frequent due to climate change (Boucher et al., 2013). Floods exist in different forms, such as pluvial flash floods due to intense rainfall events, prolonged coastal floods caused by tropical storms and hurricanes, and a combination of both, often referred to as compound flooding (Wahl, Jain, Bender, Meyers, & Luther, 2015). The state of Texas is both flood- and hurricane-prone, making it exceptionally susceptible to all types of flooding.

Large urban metroplexes in Texas have developed early flood warning systems to alert their citizens of dangerous severe weather events. Examples include the Flood Early Warning System (FEWS) in the city of Austin, TX and the Harris County flood warning system (HCFWS) for Houston, TX and the surrounding area. FEWS encompasses 130 water level gages, 15 automatic low water crossing lights or barricades, and predictive modeling and mapping (City of Austin, 2020).

The HCFWS has an even larger network of 177 water and stream level gages (Harris County Flood Control District, 2020). Preventative measures like these early warning systems are critical in reducing flood-related deaths. Most flood fatalities occur on roads, where people are unaware of the life-threatening danger of driving into flooded water and are unable to escape from their cars (Jonkman & Kelman, 2005; Sharif, Jackson, Hossain, & Zane, 2015).

Although robust flood monitoring and early warning systems exist in large cities, more rural areas and towns do not have such sophisticated flood preparedness measures in place, leaving citizens and response teams vulnerable to major storm events. In regions with ungaged rivers and no monitoring systems in place, emergency services often rely on emergency calls of rising water levels and submerged roads for road closures and evacuation plans. Depending on such reactionary measures can leave rural populations especially vulnerable to flash floods. For example, in the Texas Hill Country flood in September 2018, the Llano River suddenly swelled to 40 feet after 10 inches of rain fell in less than 24 hours (McNabb, 2018). There were two fatalities associated with the disaster.

Despite the seeming necessity of more proactive measures against flooding, there are several obstacles preventing small communities from implementing flood warning systems. Extensive gage networks are expensive to install and to maintain. Small communities may be more likely to install a few critically located gages, but a single gage can only provide a single pinpoint of data in an entire stream network. Forecasting and hydrologic models can use gage data to assign a wider significance to limited observations, but developing advanced hydrologic

models to forecast floods is often data intensive and requires significant expertise. Small, rural communities rarely have the resources available to support such measures.

There is a disconnect between people living in rural areas struggling to build flood warning systems from scratch and existing well-developed forecasting models. One such existing model is the National Water Model (NWM). The NWM is a hydrologic modelling framework that simulates observed and forecasted streamflow over the entire continental United States (CONUS) and Hawaii. The NWM ingests a variety of data ranging from radar-gauge observed precipitation to numerical weather prediction, which it uses to create informed forecasts extending up to 30 days in advance. This study proposes that the NWM creates a unique opportunity for a statewide implementation of cost-effective flood warning systems.

In this thesis, we evaluate the viability of combining observational data from streamflow gages with the National Water Model through data assimilation (DA) to bridge the gap between national flood forecasting and local emergency response. This study first seeks out to investigate the capability of applying NWM forecast outputs and local sensor observations to an external data assimilation model in real-time. The study will then evaluate the performance of the DA model with two configurations: (1) with only streamflow data assimilating into the model, and (2) with streamflow and precipitation observations informing the model.

1.2 Research Questions

This thesis examines the feasibility of utilizing the existing National Water Model forecasting framework to improve local streamflow forecasts and to eventually inform locally intelligent flood warning systems. Three research questions are addressed:

1. *1. Can an external modeling framework extract forecast data from the National Water Model and execute a data assimilation scheme using local streamflow data to create a more locally informed forecast? To what extent does this operation improve the forecasts?*

   We expect that the model chosen for this study, four-dimensional variational data assimilation, will efficiently combine the NWM forecasts and gage-observed streamflow into a single, improved flow profile. The improved streamflow profile can be propagated upstream and downstream of the streamflow sensor location.

2. *Does adding precipitation corrections into the data assimilation model improve the forecast accuracy? To what extent?*

   By adding the precipitation corrections, we will create a more complete hydrologic system within the model. The configuration incorporates surface runoff contributions to the channel in the form of lateral flow. We expect implementing precipitation corrections will improve the model performance, but to a less dramatic extent than the streamflow assimilation.

3. *Does this method of combining local gage-observed data and the National*

Water Model show potential for a state-wide flood warning system?

We expect this thesis to highlight both the strengths and weaknesses of utilizing a national forecasting framework to inform locally significant events. It is the first step in a larger effort to use the National Water Model as a tool for local flood warning systems across the state of Texas. We expect the study to provide promising results that will inform future endeavors.

1.3 Background

1.3.1 National Water Model

In August 2016, the National Oceanic and Atmospheric Administration (NOAA) established the National Water Model. The NWM is a hydrologic modelling framework that simulates observed and forecasted streamflow in over 2.7 million stream reaches over the entire continental United States and Hawaii (Office of Water Prediction, 2020). The National Water Model is comprised of four primary forecast configurations: analysis/assimilation, short-range, medium-range, and long-range forecasts. The analysis/assimilation and short-range products are produced hourly and are the focus of this work. The medium- and long-range products are ensemble forecasts extending up to 10 and 30 days in advance, respectively, and are created every 3 and 6 hours, respectively. The medium- and long-range forecasts were not chosen for this study for several reasons, including: (1) the larger temporal resolution of 3-6 hours does not suit the rapidly developing nature of flash floods, (2) forecasts with several days of lead time are extremely difficult to predict and typically lose accuracy as forecast windows extend, and (3) real-time data assimilation is better suited to describe immediate conditions or those in the near future.

The core of the NWM framework is the National Center for Atmospheric Research (NCAR)-supported community Weather Research and Forecasting Hydrologic model (WRF-Hydro). WRF-Hydro ingests forcing from a variety of sources including precipitation data from the Multi-Radar/Muti-Sensor precipitation system and Stage IV Multisensor Precipitation Estimator, and forecast data from High Resolution Rapid Refresh, Rapid Refresh, and Global Forecasting System and Climate Forecast System Numerical Weather Prediction. WRF-Hydro simulates land surface processes with the Noah-MP Land Surface Model (Niu, 2011). Separate water routing modules perform diffusive wave surface routing and saturated subsurface flow routing on a 250 m grid, and Muskingum-Cunge channel routing down National Hydrography Dataset stream reaches (Gochis et al., 2018).

For the National Water Model, WRF-Hydro employs a simple nudging data assimilation implementation to correct modeled stream flows to observed values at United States Geological Survey (USGS) gages. Nudging allows the NWM to improve model simulation and forecast initial conditions. This DA method is computationally inexpensive because it enacts a simple assimilation scheme. It inserts an observed state into the model (with some uncertainty), and the corrections are then propagated downstream within the stream network. Nudging works well close to the points of observation but can cause upstream errors. In order to avoid discontinuities, WRF-Hydro implements a limited temporal-interpolation approach to smooth the nudge in time and space. Essentially, the nudging loses power as it propagates downstream and into the forecast itself. The reader can explore the model further in the WRF-Hydro Technical Description (Gochis et al., 2018).

1.3.2 Hydrological and hydraulic routing models

A fundamental step in the development of flow routing models is the estimation of travel time and attenuation for flood waves in river channels (Heatherman, 2012). Streamflow routing models vary greatly in complexity and level of mathematical sophistication. The two primary categories of conceptual flow routing methods are hydrologic or "lumped" flow models and hydraulic or "distributed" models. The simpler hydrologic model solves the continuity equation using a relationship between storage and flow (Ionescu & Gogoaşe Nistoran, 2019). In hydrologic models, flow routing for a finite stream reach is determined by solving directly for the outflow as a function of inflow, with all geomorphological and hydraulic properties of the stream reach lumped into model parameters (Todini, 2007). Perhaps the best-known hydrologic routing model is the Muskingum method, introduced by McCarthy to manage the Muskingum River basin in Ohio (McCarthy, 1938). The Muskingum method is mass conservative, but its simplified nature makes it challenging to achieve accuracy over large flow regimes, and sharp gradients in the inflow may lead to false oscillations in the solution and negative flows in the rising limb of a hydrograph (Perumal, 1992; Perumal & Bhabagrahi, 2008).

Hydraulic routing models, on the other hand, are more flexible than their hydrologic counterparts. Hydraulic, or distributed, models calculate flow as a function of time and space throughout the system and are governed by both the continuity equation and the momentum conservation equations, i.e. the partial differential Saint Venant's equations (Ionescu & Gogoaşe Nistoran, 2019). In 1969, Cunge extended the Muskingum method to time-variable parameters whose values

could be determined as a function of a reference discharge and other parameters such as channel geometry. Cunge's time-variant model is often designated as a "semi-distributed" routing model and is commonly referred to as Muskingum-Cunge (MC).

The MC method has been widely and successfully used for discharge routing, but there are a couple of well-known shortcomings in the routing model. One such weakness is the one-dimensional nature of Muskingum-Cunge, which prevents the model from capturing the dynamics of backwater effects (Shastry et al., 2019). Backwater effects are a significant element of inland flow dynamics in coastal areas. This concept will be explored further in Section 3.2.

Another issue with Muskingum-Cunge, as several authors have pointed out, is that the approach displays a mass balance error that can reach values of 8 to 10% (Perumal & Bhabagrahi, 2008; Tang, Knight, & Samuels, 1999). Todini (2007) asserted the mass balance error exists because the original Muskingum derivation assumes constant parameters, while the Cunge alteration uses time-variant parameters. Todini proposed an iterative approach for adapting the MC model to be mass conservative. Schwanenberg and Alvarado Montero (2016) then reformulated the ideas from Todini (2007) into a numerical implementation that is mass-conservative and suppresses numerical oscillations, overshooting, and negative flows.

The National Water Model's modeling system implements a standard Muskingum-Cunge method of hydrograph routing with time varying parameter estimates (Gochis

et al., 2018). This study seeks to mimic the model framework from the NWM, but we did not want to ignore the great improvements that have been made in the field of flow routing models. To that point, this study implements the mass-conservative, iterative Muskingum-Cunge adapted routing model presented by Schwanenberg and Alvarado Montero (2016). This model will hereafter be referred to as the Muskingum-Cunge-Todini (MCT) routing model.

Chapter 2: Literature Review

2.1 National Water Model Performance

There have been a few studies that have evaluated the performance of the National Water Model since its release in 2016. Lin, Hopper, Yang, Lenz, and Zeitler (2018) studied the hydrometeorological factors constraining flood prediction skill of a physically based model resembling the NWM over the Texas Hill Country flood events in 2015. The study found that the best flood predictive skill of the NWM tends to be observed in regions experiencing rapid rises of floodwater from causative rainfall, with the model suffering at gages with less rapid flood responses and those with human-altered flows.

Austin-Petersen (2018) analyzed the accuracy of NWM analysis/assimilation and short-range forecasts during the Texas Hill Country floods of 2018 in the Llano River basin in central Texas as a measure of closeness to USGS stream gages. In gaged river basins, Austin-Petersen noted that the NWM performed well for the analysis/assimilation product, but the short-range forecast sometimes continuously predicted decreasing streamflow despite rising flood waters. He attributed this behavior to incorrect precipitation predictions and resulting rainfall-runoff mechanisms. The study indicated that the NWM performed poorly in river basins without USGS gages.

Viterbo et al. (2020) performed an error analysis on the National Water Model performance during a flash food event in Ellicott, Maryland in 2018. The study focused on understanding and assigning uncertainties within the NWM short-range forecast. They investigated the precipitation forecast product High Resolution Rapid Refresh and the corresponding hydrologic modeling framework response to understand how errors propagate through the forecast chain. Viterbo et al. (2020) found that a slight spatial displacement of a rainfall event, due to errors in meteorological inputs, can vastly effect the final streamflow forecasts produced by the National Water Model. They also concluded that these meteorological inaccuracies are compounded in large watersheds, because the location of intense rainfall events within a large watershed greatly affects the time and magnitude of peak flows.

2.2 Data Assimilation Models

Data Assimilation (DA) involves combining observations with "prior knowledge" such as numerical models to obtain an estimate of the true state of a system and the associated uncertainty of that estimate (Nychka & Anderson, 2010). The mathematical discipline of data assimilation was initially developed in the field of Numerical Weather Prediction (Daley, 1991), but its applications today are far-reaching. One example is how the new age of increasing availability of remote-sensing data has led to an expansion of data assimilation studies using satellite data (Montero et al., 2016). Interesting remote-sensing data assimilation studies include utilizing satellite data to inform a global groundwater storage time series for drought monitoring (Daley, 1991) and cloud analysis to predict tornadic thunderstorms (Hu, Xue, & Brewster, 2006).

In the realm of flow forecasting, numerous types of data assimilation methods are constantly developed to create more comprehensive and true-to-life numerical models. Examples include the assimilation of potential evaporation data (Seo, Cajina, Corby, & Howieson, 2009), of soil moisture and snow depth (Lu, Crow, Zhu, Yu, & Sun, 2015), and of streamflow data (Lee et al., 2012).

Here, we look at two primary classes of data assimilation models in hydrology—sequential and variational data assimilation. Sequential methods rely on the propagation of model updates from one time to the next in a recursive manner in order to improve the state estimate (Mclaughlin, 2002). A common principle in sequential DA is filtering, which attempts to sequentially obtain the posterior distribution of the state at the current time based on all observations collected so far (Katzfuss, Stroud, & Wikle, 2016). Kalman (1960) introduced a classic filtering model to work with the entire distribution of the state explicitly, which is now known as the Kalman filter (KF). The Kalman filter has served as a cornerstone model in the field of data assimilation. There have been adaptations of the KF; the most common are the extended Kalman filter and the widely applied Ensemble Kalman Filter (EnKF) (Evensen, 1994).

The second class of data assimilation, variational data assimilation, uses optimization algorithms to minimize an objective function (Reichle, 2008). This objective function is defined in terms of a trade-off between the amount of noise introduced into the model and the distance between simulated and observed variables (Alvarado-Montero, Schwanenberg, Krahe, Helmke, & Klein, 2017).

Recent studies using variational data assimilation to improve streamflow prediction include: operational streamflow forecasting for gauged headwater basins by assimilating streamflow, precipitation and potential evaporation data (Seo et al., 2009) and operational hydrologic forecasting in real-world experiments by assimilating streamflow data into soil moisture accounting and flow routing models (Lee et al., 2012). Lee et al. (2012) justify their decision to use a variational approach over the classic EnKF sequential model by explaining that EnKF is optimum only if the observation equation is linear; the linearity constraint can easily be violated when assimilating streamflow in large storm surges. The variational approach used in this study to assimilate streamflow into a hydrologic model is 4-dimensional variational (4D-Var) data assimilation and is described in Montero et al. (2016).

Variational methods depend on adjoint models which compute the sensitivity of the model output to each of the inputs and states of the model (Seo, Koren, & Cajina, 2003). The adjoint model is frequently seen as a significant drawback for variational methods because it adds an additional level of mathematical complexity to the model. Adjoint models will be discussed in further detail in Section 4.3.

Chapter 3: Study area and data sources

In this chapter we describe the selection process for the study area and storm events analyzed in this study. We then explore the types of data applied to the data assimilation model and the retrieval process used to gain access to the datasets. All data used in this study are open-source and available for public use. See Appendix A for online access to all data sources.

3.1 Texas Department of Transportation streamflow sensors

In 2017, the Center for Water and the Environment of the University of Texas at Austin partnered with the Texas Department of Transportation (TxDOT) to install a transect of 20 radar streamflow sensors on Interstate Highway 10 (I-10) bridges (Figure 3.1). The project's goals were, firstly, to examine the feasibility of a implementing a robust, statewide streamflow measurement program on TxDOT bridges and, secondly, to explore the potential of utilizing data from TxDOT gages and the National Water Model to more accurately depict and forecast flow conditions on the bridge and transportation network (Maidment et al., 2019). The project was titled and will be hereafter referred to as *Streamflow Measurement at TxDOT Bridges*. This thesis is an extension of the statistical work (Task 5) of *Streamflow Measurement at TxDOT Bridges*.

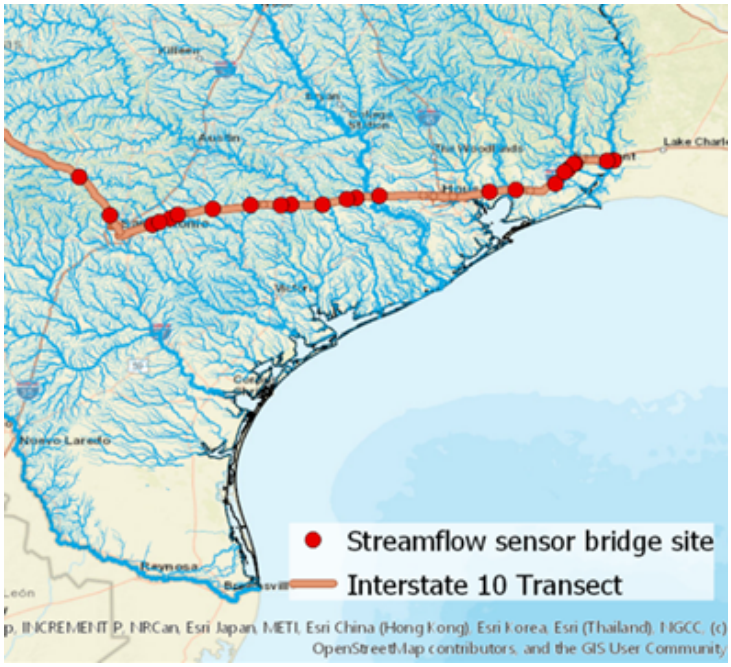

Figure 3.1: Transect of TxDOT streamflow radar sensors on I-10.

Each radar sensor on the I-10 transect measures the water elevation and velocity every 15 minutes. These data, coupled with detailed cross-sections of the streambeds, generate the channel streamflow. The TxDOT sensor streamflow data is separate from the USGS gage network and can therefore be treated as independent from the National Water Model.

The streamflow and stage data for the TxDOT streamflow sensors are processed and stored in a cloud data system maintained by KISTERS, a global software solutions and technology firm. The data can be viewed in html format or downloaded as Microsoft Excel and comma-separated values (CSV) files.

3.2 Study Area

The study area for this thesis needed to be selected from the 20 TxDOT streamflow sensors on the I-10 transect. Several of the sensors are placed in the far eastern region of Texas, near Houston and Beaumont, TX. This region is extremely flat and low-lying, with elevations near sea level, and the sensors are placed over streams or bayous close to the Gulf of Mexico. Because of these factors, there are considerable backwater effects and tidal action in these streams. The interactions between the inland flows and the coastal tides can create negative velocities in the stream, which is then transformed into negative flow rates. For example, the most eastern TxDOT sensor over the Sabine River exhibits extremely active tidal action (Figure 3.2).

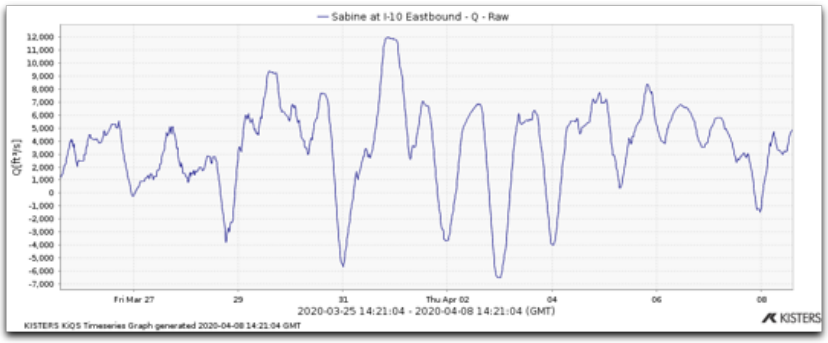

Figure 3.2: Sample hydrograph at the Sabine River TxDOT sensor demonstrating the effects of backwater and tidal action.

Unfortunately, the routing scheme used in this study cannot account for backwater effects, so these locations are not ideal for study area selection. Another factor in selecting the study area was data availability. Several of the TxDOT sensors were not fully installed and calibrated until the latter half of 2019; these sensors therefore did not record the larger storms that were captured by the se-

lected TxDOT gage described below.

The streamflow sensor we chose to analyze for this study is located in Kendall County in the Texas Hill Country and within "Flash Flood Alley" (Figure 3.3(a)).

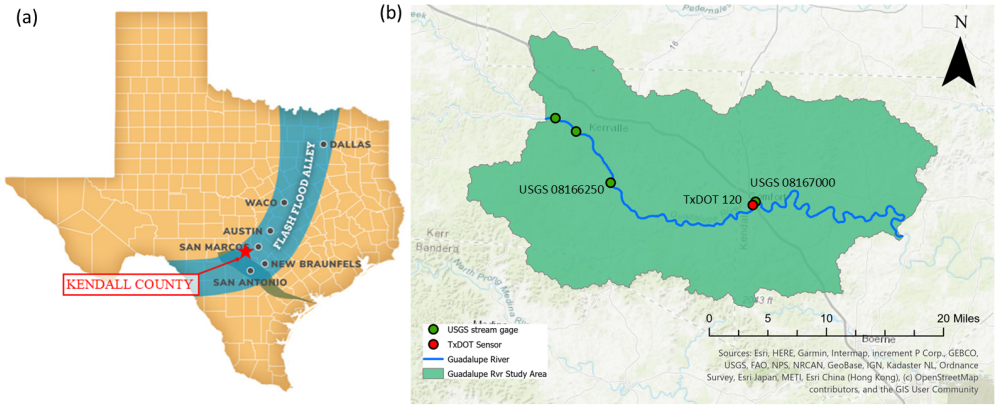

Figure 3.3: (a) Location of Guadalupe River study area relative to "Flash Flood Alley" (b) Relevant USGS and TxDOT streamflow sensors in Guadalupe River study area.

The TxDOT sensor is located at the intersection of the Guadalupe River and Interstate Highway 10. The study area then extends upstream and downstream from the sensor 50 and 30 miles, respectively. Previous work developed in *Streamflow Measurement at TxDOT Bridges* indicated that this was a conservative range for streamflow propagation (Maidment et al., 2019). We then widened the boundary to include the channel heads of this section of the Guadalupe River.

In this study area, there are four active USGS stream gages on the Guadalupe River (Figure 3.3(b)). The closest USGS gage upstream of the TxDOT gage is approximately 35 river miles away (USGS gage 08166250). Although the presence of an upstream USGS gage could affect the NWM forecasts, we considered the

span of 35 miles large enough for the influence of the nudging to phase out. The TxDOT gage itself is collocated with USGS gage 08167000. The data from USGS gage 08167000 tell a story of historical flooding. In May 2016, the stage height surged to almost 28 ft; the National Weather Service flood stage at this location is 21 ft. The USGS gage also recorded a peak flow of 61,200 cfs and a stage height of over 25 ft during the 2018 Texas Hill Country floods.

In order to circumvent the influence of the collocated USGS gage on the NWM products we substituted the NWM simulated streamflow with the sum of the two branching stream segments directly upstream. A brief exploration into the NWM forecast data confirmed the validity of this assumption.

3.3 Storm Events

The TxDOT streamflow radar sensor was installed and calibrated on November 14, 2018. In the period since installation, this region has experienced no large flooding events. For this study, we focus on the two storm events with the greatest streamflows recorded by the TxDOT streamflow sensor and one storm event with a smaller maximum flow (Figure 3.4).

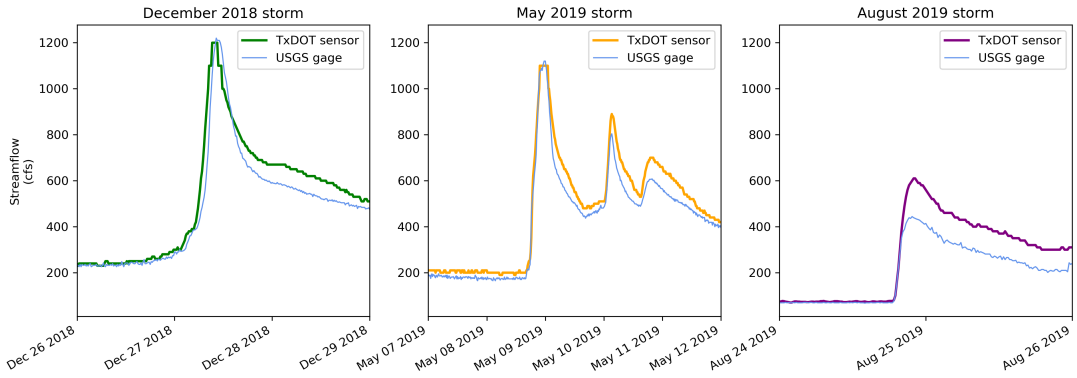

Figure 3.4: (Left) Single peak storm event in December 2018 containing the highest recorded flow on the TxDOT sensor. (Center) Complex multi-peak storm event in May 2019 spanning several days. (Right) Short, relatively small storm event in August 2019. USGS gage 08167000 streamflow is displayed as the blue series.

The first storm was a quick, single-peak storm event in December 2018, which recorded the maximum flow rate of 1,200 cfs for the TxDOT gage (Figure 3.4 (left)). The second storm occurred in May 2019, with a similar maximum streamflow as the December 2018 storm, but with a longer duration, spanning 3-4 days (Figure 3.4 (center)). The third and last storm event occurred a couple months later in August 2019, with a significantly lower peak flow of 660 cfs (Figure 3.4 (right)).

The TxDOT gage strongly agrees with the collocated USGS gage 08167000 during the three selected storm events (Figure 3.4). Leading up to the storm surges, the baseline streamflow is equivalent between the two gages. In the rising limb of the storms and during the high streamflow periods in the two greater storm events (December 2018 and May 2019) the TxDOT gage reports almost identical flow rates to the USGS gage. The only periods in which there is remarkable difference between the two streamflow sensors are the receding limbs of the storm events

and in the smaller peak flows. The most significant difference is during the small storm event in August 2019, where the TxDOT sensor recorded a peak streamflow approximately 200 cfs larger than the USGS gage. In a direct comparison of the stage height between the two gages, the water levels are evenly matched throughout all three storms (Figure 3.5).

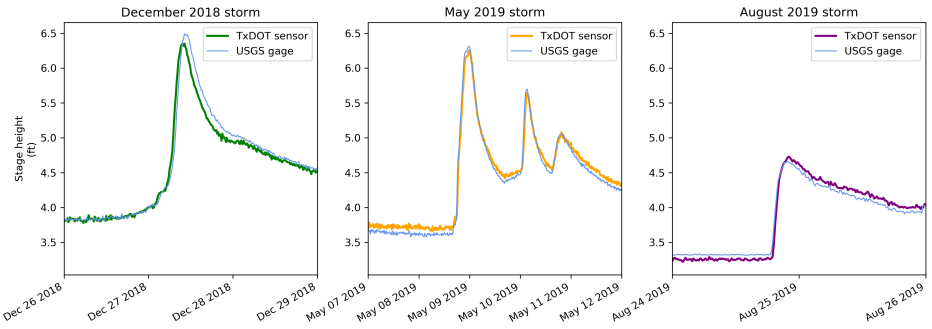

Figure 3.5: Comparison of stage values between the Guadalupe River TxDOT sensor and USGS gage 08167000 for the three chosen storm events.

The extremely similar stage values indicate that the differences in streamflow in the smaller storm and in the receding limbs of the larger storms are due to an error in velocity calibration. Overall, the comparison between the TxDOT sensor and USGS gage is very positive; the closely matched flow rates during high flows indicate that the TxDOT sensors can be relied upon for large flooding events.

3.4 National Hydrography Dataset

The National Hydrography Dataset Plus Version 2 (NHDPlusV2) is the hydrologic framework in which streamflow predictions from the National Water Model are based. The NHDPlusV2 is a comprehensive geospatial dataset of vector-based surface water features such as rivers and lakes (McKay et al., 2019). The NHD-

PlusV2 has a reach addressing system in which water features are broken into smaller reaches and given a unique ComID. Each ComID designates a singular stream reach, and each ComID has associated latitude/longitude coordinates and a corresponding contributing area known as its "catchment" (Figure 3.6).

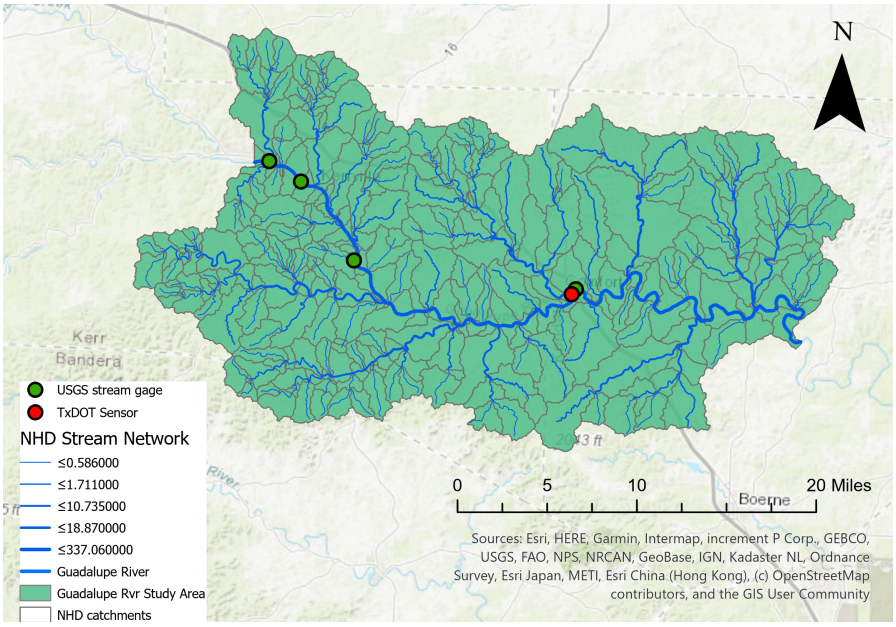

Figure 3.6: National Hydrography Dataset Plus Version 2 flowlines and catchments for Guadalupe River study area.

3.5 National Water Model forecasts

NOAA maintains a two-day rolling archive of National Water Model forecasts. This study required NWM data extending much further back than two days at the time of data retrieval. Fortunately, Google hosts an extensive archive on its Google Cloud Platform. The Google archive was created in October 2018 and is continuously updated as more NWM products are released. Each NWM product is available as a NetCDF file via manual selection on the online archive and via application programmers interface (API) calls, sometimes known as URL requests.

Although the analysis for this study is focused on a tiny fraction of the total stream reaches in the U.S., the NetCDF files in the archive are only available as bulk download of the entire CONUS. For this work, subsets of analysis/assimilation data and short-range products were collected and archived during the selected storm periods.

The two output types utilized in this study are (1) the point-type files for stream routing and reservoir tables (in this thesis referred to as channel files), and (2) 1-km gridded forcing files with the precipitation forcing data. For more information on the methodologies used to process the large NetCDF files, the reader can refer to Appendix B.

3.6 Multi-Radar Multi-Sensor precipitation data

The Multi-Radar Multi-Sensor (MRMS) system was recently implemented at the National Centers for Environmental Prediction (NCEP). MRMS combines data from a vast array of resources including multiple radars, satellites, surface observations, upper air observations, rain gauges and numerical weather prediction models to produce a suite of quantitative precipitation estimation (QPE) products (Zhang et al., 2016). MRMS provides radar-based QPE, gage-based QPE, and local gage bias-corrected radar QPE. Each product has 1-km spatial resolution and temporal resolution ranges from 1 to 72 hours. A previous study investigated the accuracy and predictive skill of precipitation products coupled with the NWM and found that the local gage bias-corrected radar QPE performs best overall (Lin et al., 2018). Viterbo et al. (2020) also used the bias-corrected QPE from MRMS as

the observational dataset to evaluate the NWM.

NCEP maintains a rolling one-week archive of real-time operational MRMS products, but the storms in this study extend past the one-week window. Iowa State University hosts an archive of several precipitation datasets in the Iowa Environmental Mesonet, including a selection of MRMS precipitation products. The archive allows for the user to navigate to a date of interest and download the target QPE product as a bulk file covering the continental United States. MRMS products are downloaded in a compressed format and, similar to the Google Cloud Platform for the NWM, can be retrieved via API calls.

The MRMS QPE files are downloaded as GRIB2 files, the World Meterological Organization's standard for distributing gridded data, in a highly compressed format in an unusual projection coordinate system. The process developed in this study to access the MRMS precipitation data involves a GRIB2 decoder program published by the NWS and additional spatial reference system manipulation. For information on the methodology used to process MRMS files, the reader can refer to Appendix C.

Chapter 4: Methodology

4.1 Channel flow routing (theory)

The National Water Model uses a straightforward variable parameter Muskingum-Cunge channel routing scheme. This routing method is based on the core continuity equation that relates input, output and storage:

$$\frac{dS}{dt} = I(t) - Q(t) \tag{4.1}$$

where *I(t)* is input, *Q(t)* is output, and *S(t)* is storage. In this situation, the input is known yet both output and storage are unknown. The Muskingum-Cunge scheme approaches this issue by framing storage in terms of 'wedges' (Figure 4.1).

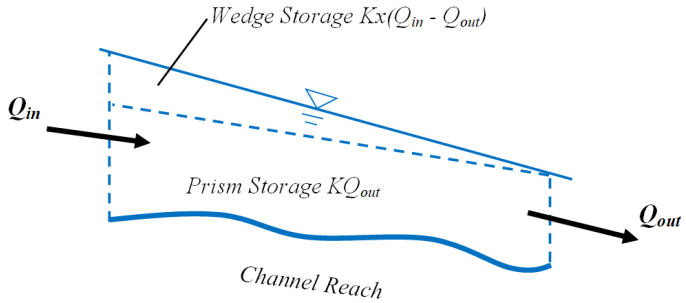

Figure 4.1: Diagram of the classic semi-distributed hydrologic routing scheme Muskingum-Cunge. Diagram adapted from Yoo et al. (2017).

The shape and size of the wedge depend on two parameters—flood wave travel time, K, and weighting factor, x (also referred to as ε). The two parameters

can be characterized as:

$$K = \Delta x / c \tag{4.2}$$

$$\varepsilon = 0.5(1 - \frac{Q}{BS_o c \Delta x}) \tag{4.3}$$

where Δx is the channel length, c is the kinematic (gravity-driven) wave celerity, Q is discharge, B is channel top width, and S_o is longitudinal channel slope. Given the dependence on discharge, the two parameters cannot be statically calculated for a given stream reach and are instead time-variable. A more detailed description of the mechanics of Muskingum-Cunge is available in Cunge (1969).

The routing model for this study is Muskingum-Cunge-Todini (MCT), which is a mass-conservative, iterative form of the Muskingum-Cunge routing model. To illustrate how the MCT scheme works we will return to Equation 4.1 and transform the equation to a discretized, integrated form:

$$\frac{S^k(I^k, Q^k, p) - S^{k-1}(I^{k-1}, Q^{k-1}, p)}{\Delta t} - I^{k-1/2} + Q^{k-1/2} = 0 \tag{4.4}$$

In Equation 4.4, *k-1* and k are time steps of a generalized flow routing model and *k-1/2* describes the flux in and out of the control volume between time steps. For simplicity, we assume that storage can be expressed as a function of inflow (I), outflow (Q), and some parameters (p). We can further discretize Equation 4.4 by introducing the theta-method to express the intermediate flux terms $I^{k-1/2}$

and $Q^{k-1/2}$ as variables of time steps *k,k-1* by:

$$I^{k-1/2} = (1-\theta_I)I^{k-1} + \theta_I I^k \tag{4.5}$$

$$Q^{k-1/2} = (1-\theta_Q)Q^{k-1} + \theta_Q Q^k \tag{4.6}$$

where $\theta_{I,Q}$ serve as time-weighting factors between the old and new time steps. Using this substitution, we can now formulate:

$$\begin{aligned} F(I^{k-1,k}, Q^{k-1,k}) = &\frac{S^k(I^k, Q^k, p) - S^{k-1}(I^{k-1}, Q^{k-1}, p)}{\Delta t} \\ &-(1-\theta_I)I^{k-1} - \theta_I I^k + (1-\theta_Q)Q^{k-1} + \theta_Q Q^k \end{aligned} \tag{4.7}$$

where F is an implicit function representing the mass error in the reservoir. Since the inflow is known at both time steps *k-1,k* and the outflow is available at time step *k-1*, the only unknown time-dependent variable in Equation 4.7 is the outflow Q^k at the new time step *k*. In a generic setup, the solution to determine the unknown Q^k can be achieved by a Newton–Raphson iterative procedure, reading:

$$Q^k_{n+1} = Q^k_n - \frac{F(Q^k_n)}{F'(Q^k_n)} \tag{4.8}$$

where n is the iteration index. The solution for Q^k is reached when the function F, or the mass balance error, is close enough to zero. We can apply this mass-

conservative formulation to the Muskingum-Cunge model to receive:

$$F(Q^k) = \frac{[K\varepsilon I + K(1-\varepsilon)Q]^k - [K\varepsilon I - kK(1-\varepsilon)Q]^{k-1}}{\Delta t} \\ -0.5(I^{k-1} - Q^{k-1}) - 0.5(I^k - Q^k) \tag{4.9}$$

$$F'(Q^k) = \frac{[(\delta K/\delta Q(1-\varepsilon) - K\delta\varepsilon/\delta Q)Q + K(1-\varepsilon)]^k}{\Delta t} + 0.5 \tag{4.10}$$

In the above formulation, the time-weighting coefficients $\theta_{I,Q}$ are equal to 0.5. By implementing $\theta_{I,Q}$ values of 0.5, the flow approximations are second order in time, creating a much more accurate flow routing method. Muskingum parameters K and ε at time step k change in the iteration history of Equation 4.8 because of its dependency on Q_n^k. In the original Muskingum–Cunge method, the parameters are evaluated at Q^{k-1} only, which corresponds to a single iteration step of the Newton–Raphson approach with an initial guess of Q^k.

4.2 Streamflow Routing Implementation

In this section we describe the hands-on implementation of the MCT routing scheme explained above. There are two types of routing used in this study—routing schemes with only channel streamflow and routing schemes with channel and lateral streamflow. Both processes are described in the following sections.

4.2.1 Channel streamflow routing (implementation)

In order to create an operational MCT routing network within the Guadalupe River study area, there are two types of required inputs—channel characteristics and flow rates. The National Water Model has a database of channel characteristics such as slope, bed width, and length for every ComID in the NHDPlusV2 stream network. The parameters are stored in a NetCDF file named "RouteLink_NHDPLUS.nc" (McKay et al., 2019). These data provide information to generate (1) synthetic rating curves which relate stage values to a correlating flow rate and (2) Muskingum's parameters K and ε by applying Equations 4.2 and 4.3. Using this information, we can create a table for each ComID in the study area, tabulating discharge, stage value, and Muskingum's K and ε, as a function of increasing stage values. The prepared tables allow for quick iterations in the MCT.

Next, there are two types of streamflow inputs required to initialize the MCT routing model. The first streamflow input is referred to as the "initial conditions" (Figure 4.2 (left)) or $Q_{start,i}$ where i denotes the stream segment. The initial conditions are essentially a snapshot of the discharge in every branch of the main channel at time = 0. The second streamflow input is referred to as "inflow" (Figure 4.2 (right)) or Q_{in}. The inflow is the channel streamflow at the farthest upstream branch for all time steps in the MCT routing model.

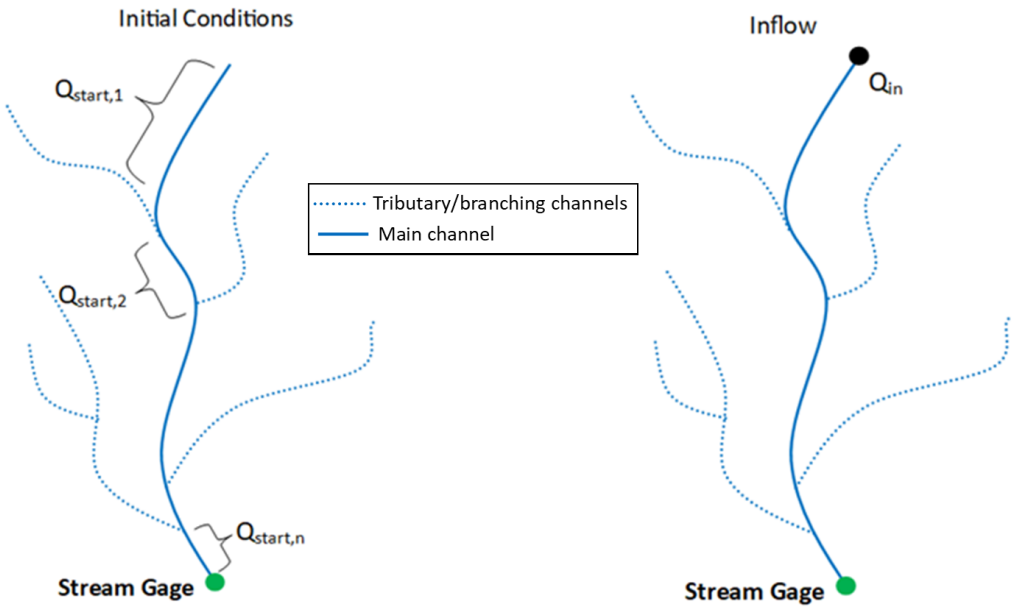

Figure 4.2: Diagrams for the two inputs into the channel flow routing model: initial conditions (left) and inflows (right).

With all inputs at hand, the Muskingum-Cunge-Todini routing model can now be initiated. At the first time step, all streamflow in the channel is known ($Q_{start,i}$). Then for all subsequent time steps, the initial flows, $Q_{start,i}$, are propagated downstream and more streamflow enters the system at the upstream boundary as inflow (Q_{in}).

4.2.2 Channel and lateral streamflow routing (implementation)

In real-world scenarios rivers swell and flood waves propagate, but these actions are more complex than a simple function of increased upstream flow. During storm events, precipitation falls on the terrain surrounding the channel network, and a fraction of that precipitation becomes surface runoff. The overland flow

can then drain into the main channel, forming a streamflow contribution called "lateral" flow. In Section 4.5, we further discuss the mechanisms that determine what fraction of precipitation is transformed into surface runoff. In the MCT routing model, the lateral flow is treated as an additional input into each ComID-designated stream segment and is merged into the channel flow.

In the Muskingum-Cunge-Todini routing and 4-dimensional variation data assimilation (4D-Var DA) coupled model, the lateral flow inputs are not treated like the main channel streamflow. They are not iterated upon, but they are accounted for in the overall water budget. By including lateral flows, we can account for the rainfall-runoff phenomenon along the main river channel and thus form a more realistic flood wave event.

4.3 Streamflow data assimilation (theory)

In order to improve the National Water Model short-range forecasts, the TxDOT streamflow data are assimilated using 4D-Var data assimilation coupled with the flow routing scheme described in the previous section.

Unlike the "nudging" data assimilation model currently implemented in the NWM, 4D-Var allows for corrections in the streamflow both upstream and downstream from the point of observation at the sensor location (Figure 4.3).

Figure 4.3: Simplified diagram to illustrate how the 4-dimensional variational data assimilation model operates.

In the 4D-Var DA model, observed streamflow is accepted at a downstream location, notated "sensor location" (Figure 4.3). An upstream inflow is then iteratively altered to suit the downstream flow profile. Essentially the model works by ingesting a downstream hydrograph and asking itself what could have occurred upstream to create this streamflow profile. It forms that upstream hydrograph and routes it downstream to the sensor using the Muskingum-Cunge-Todini routing scheme.

As stated earlier, variational methods use optimization algorithms to minimize an objective function (Seo et al., 2003). The objective function is a summation of weighted squared distances between model simulation series and the observa-

tion series for the duration of the analysis (Chao & Lang-Ping Chang, 1992). The objective function for the 4D-Var DA model for assimilating streamflow reads:

$$\min_{\Delta Q_{DA}} f = \sum_{k=-N+1}^{0} [w_{QI}(QI_{DA,k} - QI_{NWM,k})^2 + w_{QO}(QO_{obs,k} - QO_{DA,k})^2] \quad (4.11)$$

where k is the time step index, and w_{QI} and w_{QO} are experimentally determined weighting coefficients. Equation 4.11 can be split into two terms: (1) the squared difference at the inflow location between the inflow series created by the DA model ($QI_{DA,k}$) and the original inflow from the NWM ($QI_{NWM,k}$), and (2) the squared difference at the streamflow sensor between the observed streamflow ($QO_{obs,k}$) and the streamflow simulated from the DA model ($QO_{DA,k}$). Simply put, the objective function is a trade-off between the amount of upstream flow alteration introduced into the model and the distance between simulated and observed variables.

The minimization problem described in Equation 4.11 theoretically has an infinite number of solutions, so we need a method to optimize the solution. One approach is to compute the derivatives of the objective function with respect to the noise terms to enable the use of gradient-based optimizers (Alvarado-Montero et al., 2017). However, calculating numerical differentiation for each possible iteration of noise introduced can quickly become computationally cumbersome for large optimization problems and can introduce truncation errors (Alvarado-Montero et al., 2017). Instead, we can use adjoint models. Adjoint models calculate the sensitivity of the model to changes in the inputs and model states through algorithmic differentiation in reverse mode (Griewank & Walther, 2008). This process essentially traces first-order derivatives backwards in time through the simulation

model, allowing for a single run to evaluate all gradients (Seo et al., 2003). The general adjoint equation for the 4D-Var model in this study reads:

$$\sum_{k=-N+1}^{0} [2w_{QI}(QI_{DA,k} - QI_{NWM,k}) - 2w_{QO}(QO_{obs,k} - QO_{DA,k})] \tag{4.12}$$

The data assimilation methodology for National Water Model short-range forecasts hinges on the determination of a look-back period, sometimes also referred to as the assimilation window. The look-back period is the portion of the forecast that is dedicated to data assimilation (Figure 4.4).

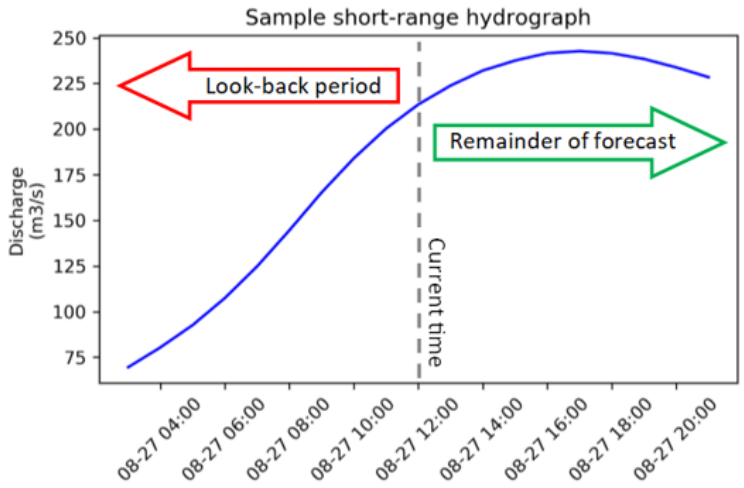

Figure 4.4: Diagram to illustrate how the National Water Model short-range forecasts are divided as a function of the lookback window length.

In order to significantly alter the flow profiles, some part of the 18-hour forecast time-series needs to have already happened. For example, say it is 11:00 AM on 8-27-19, as depicted in Figure 4.4, and a town forecasting center would like to run the data assimilation model. They decided they need a forecast window of 9 hours, so they dedicate the first 9 hours to assimilation and the last 9 hours to forecast propagation. To do this, the model would retrieve the short-range forecast

initiated at 8-29-19 2:00. It could then run the data assimilation model for the last 9 hours of observed discharge from the gage, then propagate the altered flow for the remainder of the forecast (9 hours).

4.4 Streamflow data assimilation (implementation)

The procedure for initiating the data assimilation model is similar to the methods described in Section 4.2.1 for streamflow routing without DA. The model requires the same two inputs of (1) initial conditions and (2) inflow series at upstream location, but the DA model also requires an additional third input of the observed streamflow series from the TxDOT sensor. Once the inputs are ready, the 4D-Var data assimilation model runs for a pre-determined lookback period. The model solves the optimization problem of the most feasible upstream hydrograph to create the downstream observation. The new data-assimilated series at the sensor location can then be inserted into the entire 18-hour short-range forecast and routed downstream using the MCT routing scheme.

By the nature of data assimilation models, the 4D-Var method only improves data during periods in which observations are available. Once that period ends, the improved series quickly reverts to the raw National Water Model forecast series. This can be improved with coupled forecast modeling, but it is not within the scope of this study.

4.5 Streamflow and precipitation data assimilation

As discussed in Section 4.2.2, the addition of precipitation corrections into the data assimilation process adds a layer of complexity and creates a more realistic model. When precipitation falls to Earth's surface, there are a myriad of physical and meteorological characteristics that affect runoff. These factors include, but are not limited to: land use, soil type, soil saturation levels, slope of the land, distribution of rainfall over the drainage basin, the size of the drainage basin, and direction of storm movement (USGS, 2020). The National Water Model uses the Noah-MP land-surface model (Niu, 2011) to model these complex interactions. However, the motivation behind this study is to create a simplified data assimilation model that can be implemented in any small community without extensive hydrologic knowledge and modeling experience. To that end, we investigated the viability of using a simple rational method approach to transform precipitation events to equivalent surface runoff. The rational method is used to estimate flow rate for a given precipitation event, which is computed using:

$$Q = CIA \tag{4.13}$$

where Q is the equivalent flow rate, C is the runoff coefficient, I is the rainfall intensity, and A is the drainage area (Dhakal et al., 2011). The C coefficient ranges from 0 for a completely pervious surface to 1 for a completely impervious surface. The exact interpretation of the C coefficient varies throughout the literature, but it is generally defined as the portion of rainfall that becomes direct runoff during an event (Merz, Blöschl, & Parajka, 2006).

In this study, we characterize the C coefficient using a land cover raster, which can be obtained at a 30-meter resolution from the 2016 National Land Cover Dataset (NLCD). The relationship between land cover class and C values has been investigated in previous studies, and the studies conservatively determined and applied C values corresponding to NLCD classes (Greer, Wilbanks, Clifton, Wilson, & Graettinger, 2018; Young, McEnroe, & Rome, 2009). Table 4.1 shows the equivalent C values for discrete land cover classes.

Table 4.1: Land cover classes and equivalent C values

NLCD classes and Rational Method Coefficients

NLCD Value	Land Cover Class	C Value
11	Open water	1
21	Developed, open space	0.8
22	Developed, low intensity	0.85
23	Developed, medium intensity	0.9
24	Developed, high intensity	0.95
31	Barren land	0.6
41	Deciduous forest	0.3
42	Evergreen forest	0.2
43	Mixed forest	0.25
52	Shrub/scrub	0.7
71	Herbaceous	0.65
81	Hay/pasture	0.4
82	Cultivated crops	0.45
90	Woody wetlands	0.9
95	Emergent herbaceous wetlands	0.95

Using Table 4.1 and the catchment polygons from NHDPlusV2, we can

estimate the composite C value for a given ComID drainage area using:

$$C = \frac{\sum_{j=1}^{N}(C_j A_j)}{\sum_{j=1}^{N} A_j} \tag{4.14}$$

where A_j is the area for land cover j and n is the number of distinct land cover categories. The NLCD land cover raster is clipped to the catchments along the main channel of the Guadalupe River (Figure 4.5), and a composite C value is computed for each catchment.

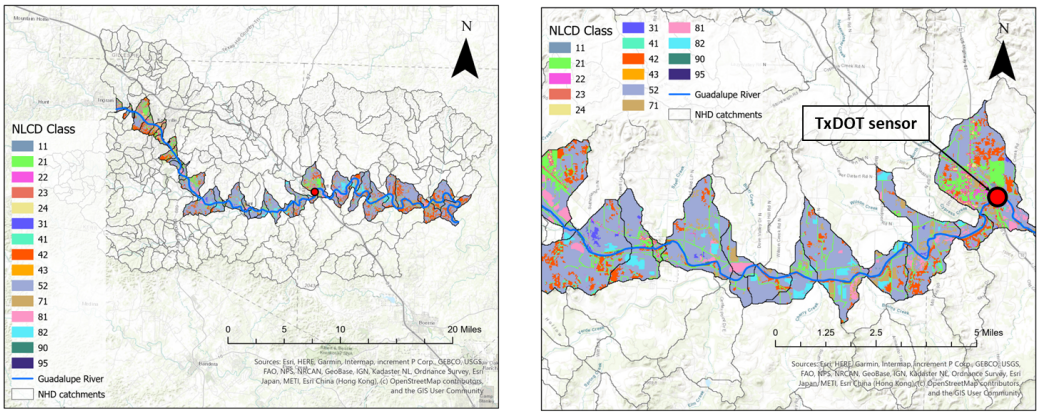

Figure 4.5: National Land Cover Dataset raster clipped to the NHD catchments along the entire Guadalupe River main channel in the study area (left) and directly upstream of the TxDOT sensor (right).

Using the composite C coefficients for the NHD catchments, we can calculate an adjusted lateral flow value for each ComID by applying:

$$q_{lat,corrected} = q_{lat,NWM} + C(I_{MRMS} - I_{NWM})A_{catchment} \tag{4.15}$$

where $q_{lat,NWM}$ is the lateral flow product from the NWM provided in the channel output file (derived from LSM terrain routing), I_{MRMS} is the rainfall intensity

from the MRMS gage-adjusted QPE, I_{NWM} is the rainfall intensity from the NWM gridded forcing file, and $A_{catchment}$ is the area of the individual NHD catchment. The corrected lateral flow can now be inserted into the routing/data assimilation run as indicated in Section 4.2.2.

4.6 Data assimilation evaluation metrics

Two statistical metrics are used to evaluate the data assimilation model performance at the sensor location. Table 4.2 summarizes the statistical indicators used in the study.

Table 4.2: Performance metrics for quantitative analysis

Statistical indicators and relevant information

Abbreviation	Statistical Indicator	Value Range	Performance Classification	Reference
R^2	Coefficient of determination	$0.7 < R^2 < 1$ $0.6 < R^2 < 0.7$ $0.5 < R^2 < 0.6$ $R^2 < 0.5$	Very Good Good Satisfactory Unsatisfactory	(Moriasi et al., 2007)
NSE	Nash-Sutcliffe Efficiency coefficient	$0.75 < \text{NSE} < 1$ $0.65 < \text{NSE} < 0.75$ $0.5 < \text{NSE} < 0.65$ $0.4 < \text{NSE} < 0.5$ $\text{NSE} < 0.4$	Very Good Good Satisfactory Acceptable Unsatisfactory	(Moriasi et al., 2007) (Boskidis et al., 2012)

The Nash-Sutcliffe Efficiency (NSE) coefficient is given by the formula:

$$NSE = 1 - \frac{\sum_{i=1}^{n}(Q_{obs} - Q_{sim})^2}{\sum_{i=1}^{n}(Q_{obs} - Q_{obs,avg})^2} \tag{4.16}$$

where n represents the number of observations in the study, i indicates the time series of the observed and simulated pairs, Q_{obs} and Q_{sim} stand for the observed and simulated streamflow, respectively, and $Q_{obs,avg}$ is the average observed discharge

by the TxDOT sensor during the storm event. NSE is a widely used statistical indicator that measures the predictive power of hydrologic models. Its value ranges from $-\infty$ to 1, where 1 indicates a perfect model and negative values indicate that the observed mean is a better predictor than the model.

The coefficient of determination (R^2) is represented by the equation:

$$R^2 = \frac{\sum_{i=1}^{n}[(Q_{obs} - Q_{obs,avg})(Q_{sim} - Q_{sim,avg})]^2}{\sqrt{\sum_{i=1}^{n}(Q_{obs} - Q_{obs,avg})^2}\sqrt{\sum_{i=1}^{n}(Q_{sim} - Q_{sim,avg})^2}} \tag{4.17}$$

where the notation is identical to that in Equation 4.16. The coefficient of determination provides a measure of how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model (Steel & Torrie, 1960).

Chapter 5: Analysis Results and Discussion

The analysis is divided into three main sections. First, in Section 5.1, we explore how the National Water Model performed during the three flood events before any external assimilation. Next, in Sections 5.2 and 5.3, we quantitatively and qualitatively evaluate the streamflow-only data assimilation model, and finally in Section 5.4, we evaluate the streamflow + precipitation assimilation model.

5.1 Performance of NWM without Data Assimilation

In order to assign importance to the data assimilation model in this study, we need to evaluate how the National Water Model performed without external interference from the TxDOT sensor observational forcing. In this section, we qualitatively compare the sequence of short-range forecasts and the analysis/assimilation time series to the observed discharge recorded by the TxDOT streamflow sensor for each storm event.

5.1.1 December 2018 storm event

Throughout the duration of the December 2018 storm event, the accuracy of the Nation Water Model short-range products varies widely (Figure 5.1).

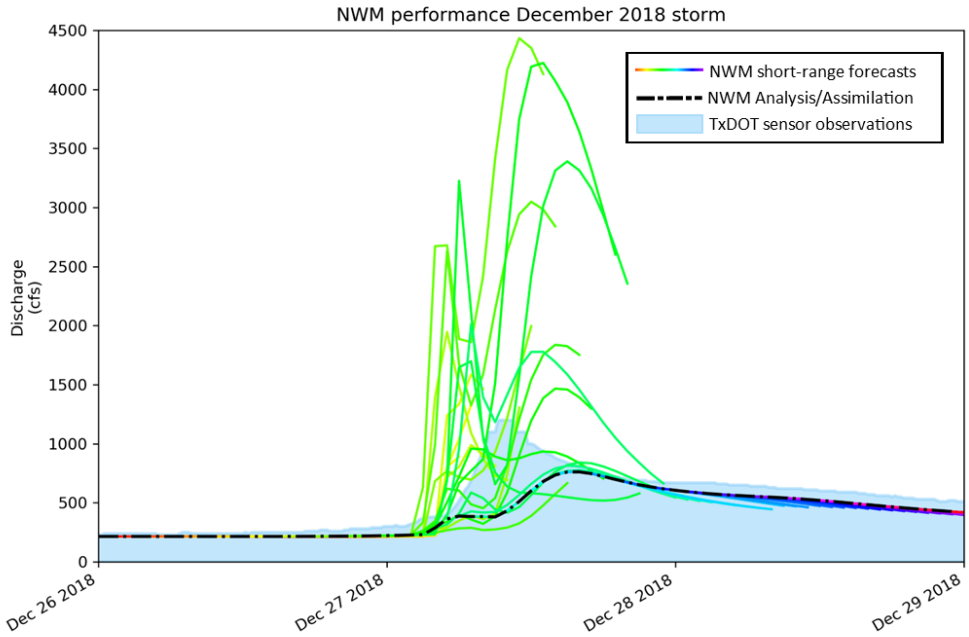

Figure 5.1: Performance of the National Water Model short-range and analysis/assimilation forecast products during the December 2018 storm event.

The short-range forecasts with long lead times before the storm event do fairly well. They are indicated by the lighter yellow series in Figure 5.1. Most forecasts predict the timing of the streamflow surge, which is indicated by the lack of short-range series remaining at the initial baseline flow. As the storm draws nearer, however, the forecasts begin to greatly overestimate the peak flows of the flood event. Several forecasts predict peak flows two to a little over three times higher than the observed maximum discharge (1,200 cfs). The short-range forecasts during this period are largely variable as well. Some forecasts are actually under-predicting the storm magnitude and some are more noticeably over-predicting. The NWM then recovers in the falling limb of the storm, agreeing well with the timing and the magnitude of the receding flow rate.

The analysis/assimilation series creates a hydrograph of a two-peak storm, with the true streamflow peak occurring in between them. It largely misses the high flows that occur during that peak event; the maximum flow rate for the analysis/assimilation product is slightly over half of the true peak flow.

5.1.2 May 2019 storm event

Leading up to the May 2019 storm event, the National Water Model exhibits significant variability in its prediction of the storm surge (Figure 5.2).

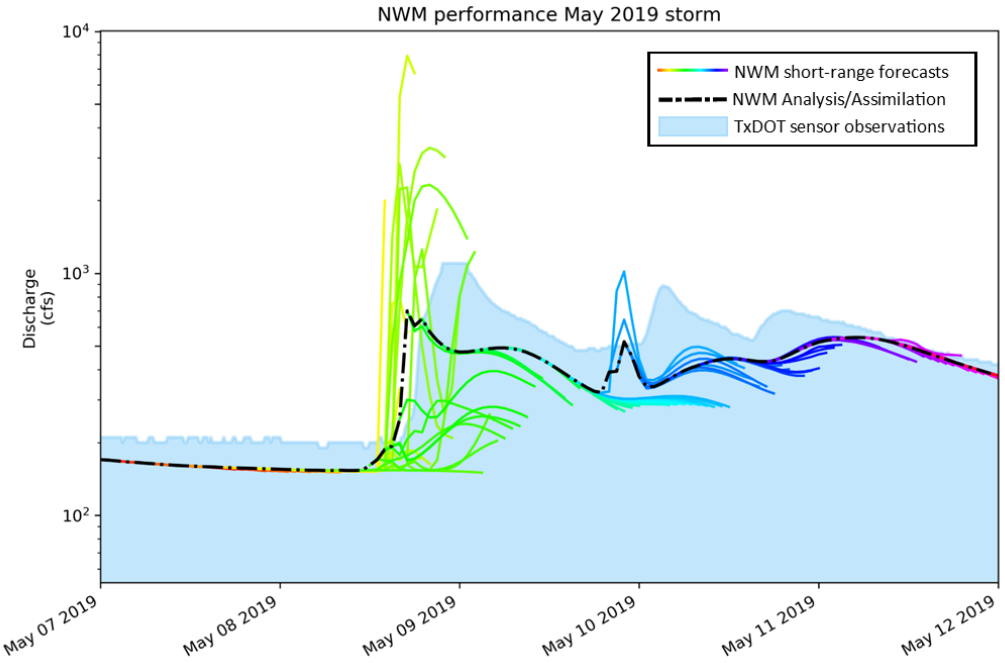

Figure 5.2: Performance of the National Water Model short-range and analysis/assimilation forecast products during the May 2019 storm event. The y-axis is log-scale for clarity.

The short-range forecasts produced in the pre-storm period both vastly

overestimate and underestimate the magnitude of the storm event, depending on the individual forecast, than was observed by the TxDOT streamflow gage. The forecasts in the rising limb of the storm largely underestimate the event as well. Unlike the two other storms analyzed in this study, the May 2019 storm is a multi-peak, several-day event. After the first, and largest, rising limb passes, the National Water Model recovers well. Although slightly offset in its timing and magnitude of the following peak flows, the NWM short-range forecasts during this period generally forecast the shape of the flow event.

The analysis/assimilation time series for the May 2019 storm performs well. It effectively estimates the shape of the observed hydrograph, but the streamflow is approximately 75% of the observed discharge at the TxDOT gage.

5.1.3 August 2019 storm event

For the August 2019 storm, most of the short-range National Water Model forecasts fall within the general magnitude of the surging streamflow (Figure 5.3).

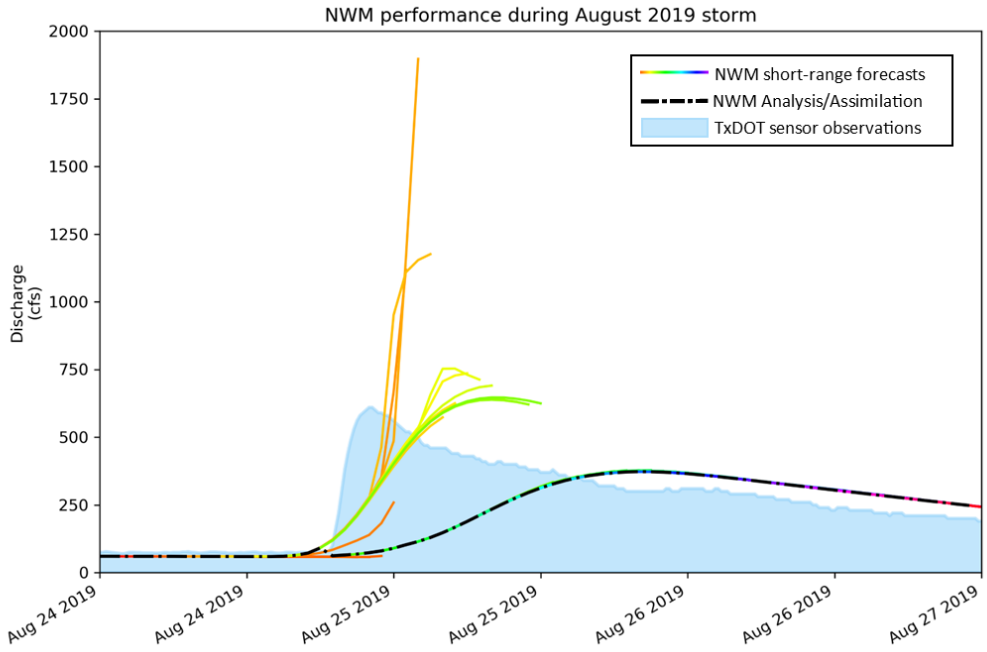

Figure 5.3: Performance of the National Water Model short-range and analysis/assimilation forecast products during the August 2019 storm event.

In the pre-storm and rising limb period, most of the short-range forecasts predict the rise in streamflow reasonably accurately (yellow to light green time series in Figure 5.3). The magnitude is close to the true peak discharge (660 cfs), but the streamflow surge in these forecasts takes longer to develop than the quick, two- to three-hour period observed in the TxDOT sensor. During this period, there are a few outlier short-range forecasts that predict significantly greater flow in the last couple hours of the 18-hour forecast. This behavior, also seen in the other two storms, is likely attributed to incorrect meteorological forecasts several hours in the future.

The short-range forecasts identically follow the analysis/assimilation series in the falling limb of the storm event. The NWM estimates the increased flow to

be much slower and much more delayed than the observed streamflow.

5.2 Quantitative analysis and discussion of streamflow data assimilation

The quantitative analysis is broken into two sections. First, the results from the quantitative analysis are presented and explained in Section 5.2.1. Then, key takeaways from the analysis are discussed in Section 5.2.2.

5.2.1 Quantitative streamflow data assimilation results

As mentioned in Section 4.6, the performance metrics used in this study are the Nash-Sutcliffe Efficiency coefficient (NSE) and the coefficient of determination (R^2). The data assimilation model was run for each storm given three different lookback periods of 6, 9, and 12 hours. The NSE and R^2 values are evaluated for each data assimilation run and then plotted against the hydrograph for the relevant storm event (Figure 5.4). The DA series evaluated in Figure 5.4 only span the lookback period itself, not the entire 18-hour forecast. The full short-range forecast series are presented later in Figure 5.5.

In this section, we also compare the performance of the un-assimilated NWM forecasts to the data assimilation model results. The NSE values are calculated between the un-assimilated NWM forecasts and the TxDOT sensor observations, both for the first 12 hours of the forecast (Figure 5.4) and for the entire short-range period (Figure 5.5). The un-assimilated R^2 performance scores are not

included to preserve the clarity of the figures. The R^2 values for the DA lookback series rapidly change from time step to the next, making several series difficult to depict. Additionally, the bulk of the following analysis is focused on the NSE results because they are more discerning in their evaluation of the model.

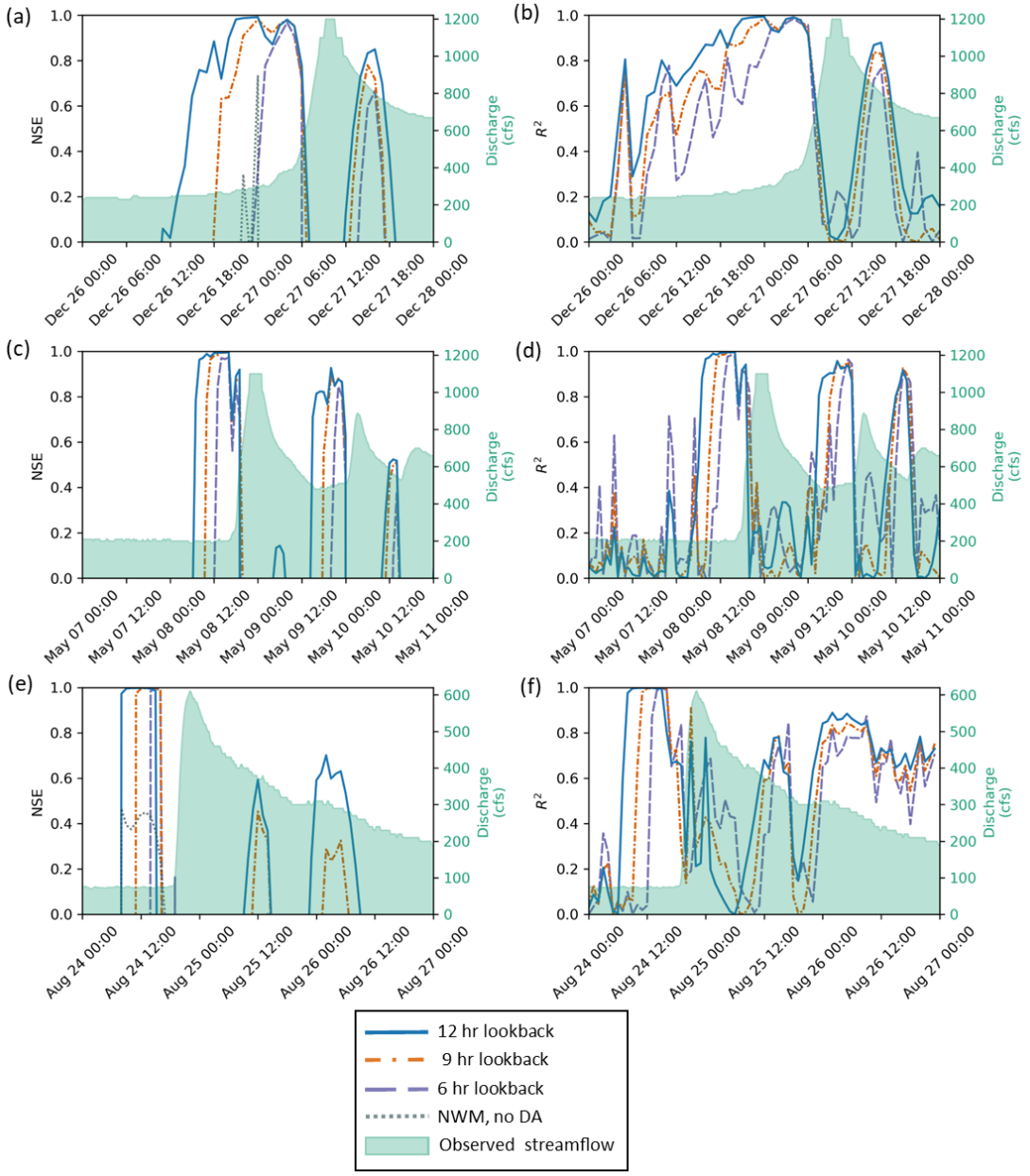

Figure 5.4: Data assimilation evaluation for all storms with respect to NSE (left) and R^2 (right). The duration of the model runs span the look back periods of 6, 9, 12 hours.

During the period leading up to the December 2018 storm surge, the DA model performs well across all lookback periods, which is indicated by NSE values rapidly increasing to the 0.8 to 1 range (Figure 5.4 (a)). These high scores are con-

sidered to be within the "very good" range for evaluating hydrologic simulations (Table 4.2). The un-assimilated NWM forecasts do not perform well in this period; the NSE values are very low with only two visible peaks above zero (Figure 5.4 (a)). The 12-hour lookback DA model results outperform the simulations with shorter lookback periods. The NSE values from the 12-hour lookback simulations are in the "very good" range 12 hours before the surge in streamflow, which is approximately 3 and 6 hours before the the 9-hour and 6-hour lookback period simulation results, respectively. The model runs with longer lookback windows regularly receive better statistical scores because they have more informed data points in the 18-hour forecast, which in turn produces much more closely matched series overall.

After the pre-storm period in the December storm event, the DA results from all three lookback periods rapidly deteriorate. The NSE values plunge below zero during the second half of the rising limb and peaking stage of the storm (Figure 5.4(a)). Negative NSE values indicate that the observed mean is a better indicator than the model. The magnitude of a negative NSE value does not convey significance, so results in the negative NSE range are not differentiated from each other in this study. The phenomenon of why the NSE values are negative for forecasts initiated during periods of abruptly increasing streamflow is investigated further in Section 5.2.2. After the peak streamflow has passed, the NSE values re-surge to the 0.75 to 1 range, with the 12-hour lookback DA model once again maintaining high NSE values for the longest period.

The DA model results from the December 2018 storm event are similar when measured by the coefficient of determination (Figure 5.4(b)). In the pre-storm pe-

riod, the R^2 values gradually increase from the "acceptable" and "satisfactory" value ranges (Table 4.2) to the "very good" range of 0.85 to 1. All three data assimilation model results maintain extremely high coefficient of determination scores for more than 6 hours in the end of the pre-storm period. However, once the storm surge begins, the model performance of all three simulations drops to unsatisfactory levels. The R^2 values again increase to good levels after the surge, much like the NSE scores. Throughout the entire storm duration, the periods of time in which the R^2 values are at excellent levels are much longer than their NSE counterparts. This pattern persists in the next two storms as well, indicating that this is a byproduct of the statistical indicators themselves.

The performance results from the multi-peak, prolonged storm event in May 2019 (Figure 5.4(c) and 5.4(d)) echo the patterns we observed in the December 2018 storm. The data assimilation model once again does well in the pre-storm period with NSE and R^2 values at excellent levels just before the first storm surge. Once the streamflow begins rapidly increasing, the results fall to unsatisfactory levels. The model then recovers during the receding limb of the storm event. The time periods in which the DA model performs well are narrower in the more complex May storm event, but this cycle is evident in all three peaks. The unassimilated NWM forecasts do not predict the observed streamflow adequately; the NSE values are negative for the entire May 2019 storm event.

The August 2019 storm event exhibits similar behavior to the first two storms (Figure 5.4(e) and 5.4(f)). The performance values repeat the same cycle of excellent performance in the pre-storm period, followed by unsatisfactory scores

during the period of increasing flows, which is in turn followed by good results in the receding limb. The un-assimilated forecasts are "adequate" (Table 4.2) for a few hours in the pre-storm period and are not seen again. The DA model results in the receding limb are lower than the December 2018 and May 2019 storms.

Next, the performance metrics are applied to the entire short-range forecasts (Figure 5.5).

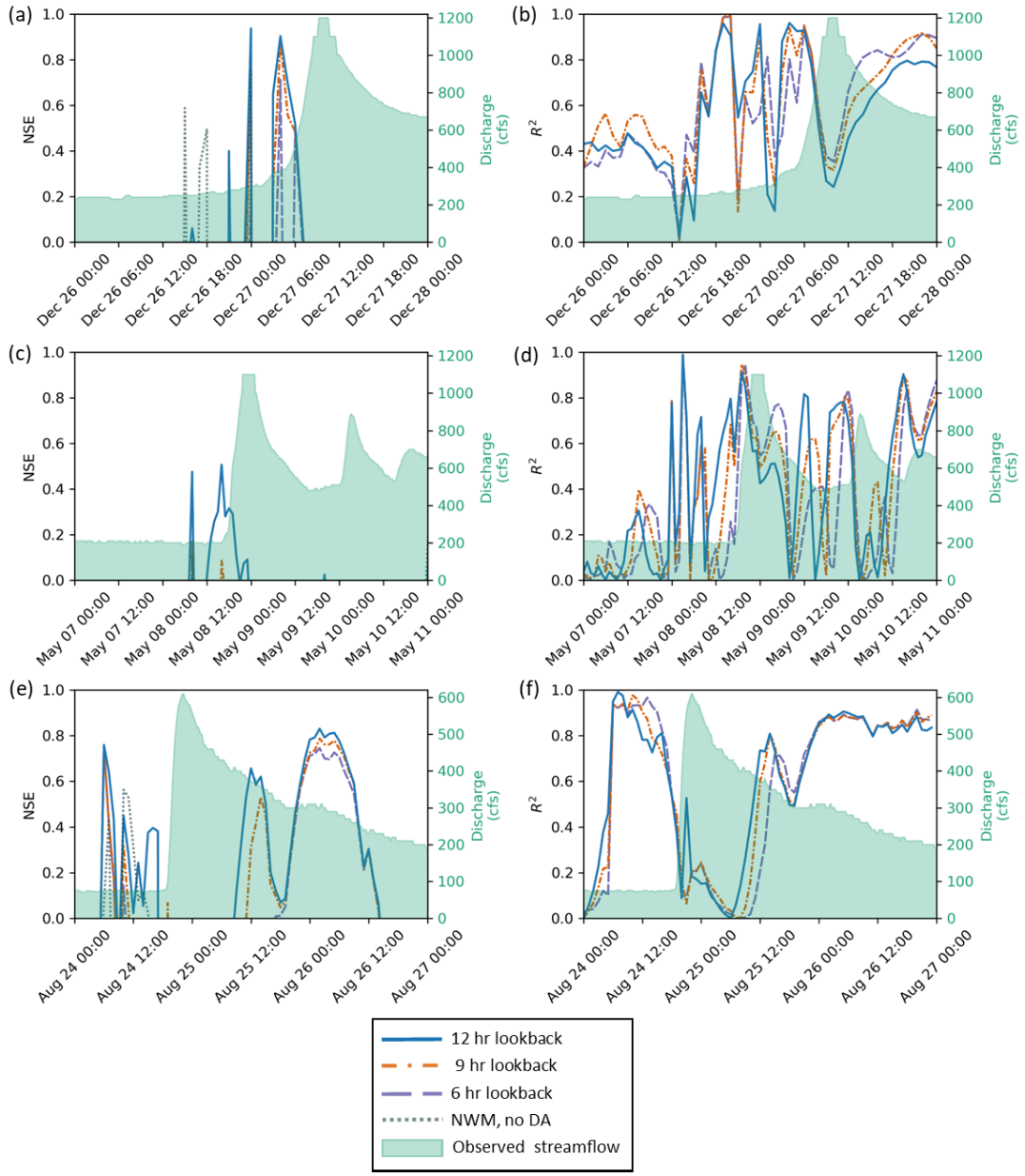

Figure 5.5: Data assimilation evaluation for all storms with respect to NSE (left) and R^2 (right). The model runs span the entire short-range 18-hour forecast.

Here, the remaining (un-assimilated) sections of the 18-hour forecasts are incorporated into the performance metrics. This marks the departure from observation-improved streamflow to the integration of unaltered National Water Model fore-

casts. Therefore, the information gathered from the statistical indicators (Figure 5.5) is an evaluation of the National Water Model as a method to attach forecasting capacity to the assimilated streamflow, rather than an evaluation of the data assimilation model itself.

Across all storms and lookback windows, the agreement between the data assimilation model results and the TxDOT observations decreases substantially, indicated by a significant decline in the statistical indicator scores. The December 2018 and May 2019 storms particularly display significant reduction in performance scores. The May 2019 storm event only has a few individual results from the 12-hour lookback model to rise above the unsatisfactory levels (Figure 5.5(c)). The coefficient of determination is not as harshly affected; there are regions in every storm across all three lookback periods that have R^2 values in the "good" to "very good" range.

Overall, the data assimilated results outperform the raw, un-assimilated National Water Model forecasts. The NSE scores in the pre-storm period are greater for the DA model excepting a few early spikes in the December 2018 storm and one spike in the August 2019 storm. However, the DA model results no longer have a dominance over the un-assimilated NWM series. This change in behavior suggests that the errors in the remainder of the forecast (Figure 5.4 vs. Figure 5.5) are significant.

5.2.2 Quantitative streamflow data assimilation discussion

A positive result from the lookback-only analysis (Figure 5.4) is the excellent performance of the DA model in the pre-storm period before the streamflow surges hit the TxDOT gage. In all three storms, the performance of the 12-hour lookback model is "very good" for at least 12 hours before the increase in flow. This period marks the most critical time for flood streamflow forecasting. Another critical period during a flood event is the receding limb of a storm surge. During this time, emergency responders want to know when the flood waters will recede and when roads will be accessible. The high performance values in this region are also an encouraging result.

The significant decrease in performance scores from the assimilation period (Figure 5.4) to the full short-range forecast (Figure 5.5) indicates that the NWM forecasts from the three storms do not agree with what was ultimately observed. The combination of a well-matched forecast with the data assimilation window is critical in creating an operational and reliable flooding warning system. If the National Water Model is not performing well in what it sees as ungaged watersheds, there may be a need for more involved methods for transitioning from the assimilation window to the forecast period.

One option to improve the forecast period is to implement a temporal-interpolation method between the simulated series and the NWM forecast at the last point of the assimilation window. This method would create interpolated values between the last observations and the future forecasted streamflow, thus

enabling the observations to carry weight beyond the assimilation window. This operation is similar to how the nudging method works in the National Water Model forecasts.

The poor performance scores in both the lookback window-only (Figure 5.4) and full short-range forecasts (Figure 5.5) during periods of rapidly increasing streamflow prompt a need for further investigation. Although forecast warning systems emphasize the need for accurate forecasts in the pre-storm period, the climbing limb is still extremely important in more prolonged flooding events. We have included a sample data assimilation model run of the full short-range forecast at the peak of the December 2018 storm event (Figure 5.6). The NSE values for the three simulated look-back periods and for the un-assimulated NWM forecast are all negative.

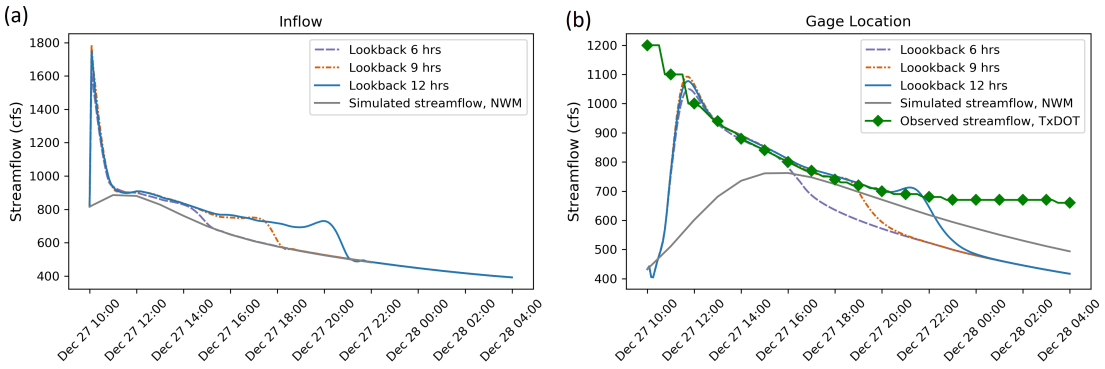

Figure 5.6: Data assimilation model run for the peak flow of the December 2018 storm event with low NSE and R^2 scores.

Initially, it may seem unclear why the forecast received such low NSE scores. Quickly after the model is initialized, the simulated streamflows match the ob-

served streamflow and remain close to the observations until the lookback periods expire. The simulated streamflows then rejoin the forecast in the upstream location (Figure 5.6(a)), and the flow is propagated downstream, which is lower than the observed streamflow. This is the expected behavior for the 4D-Var DA model if the NWM is under-forecasting.

Upon closer inspection, we see the first data point of the observed streamflow is almost three times the value of the the NWM estimation. It is known that the NSE metric can be sensitive to outliers, making this initial difference very powerful in the overall performance score. The gap between the observed and simulated series quickly closes, but once the data assimilation periods are over, the flows again diverge. The simulated forecasts return to falsely under-forecasted streamflows. Therefore, although the DA model did not fail to properly assimilate the observed values, the NSE evaluations produce unsatisfactory scores.

Another interesting result in this analysis is the unsatisfactory performance scores in the pre-storm period for the full short-range forecast model (Figure 5.5) relative to the lookback-only model (Figure 5.4). These drastically lower scores are a direct reflection of the short-range forecasts (Figures 5.1, 5.2, and 5.3) that significantly overestimate, and sometimes underestimate, the magnitude of the storm surge. If the NWM predicts extremely high flows in the remainder of the forecast (last 6 to 12 hours, depending on the lookback period), the high flows are directly absorbed into the DA series. The large disparity between simulated and observed series would return poor NSE values.

A possible method to improve these forecasts in the pre-storm period is take several previous forecasts into account. The sequence of forecasts frequently oscillates between vastly overestimating and underestimating the storm magnitude, reflecting large uncertainty in the forecast. Optimally combining some previous number of forecasts could create a more stable, time-averaged forecast series.

5.3 Upstream and downstream propagation of streamflow alteration

Once the data assimilation model has created improved streamflow at the inflow location and at the sensor location, the altered flow profiles can be propagated downstream (Figure 5.7).

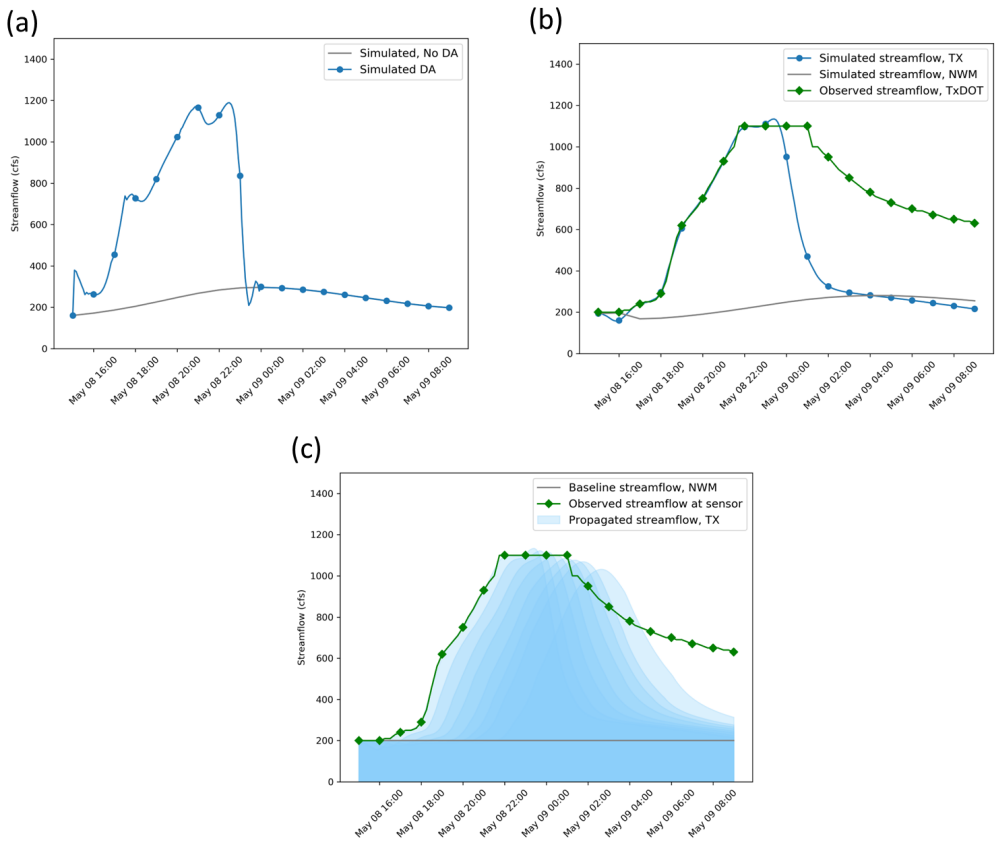

Figure 5.7: Sample data assimilation results from upstream and downstream flow propagation. (a) Hydrograph 10 miles upstream; (b) Hydrograph at the sensor location; and (c) Hydrograph of each downstream reach, extending 18 miles downstream.

The power of using 4-D variational data assimilation rather than the traditional sequential or nudging DA models is evident in its far-reaching influence both upstream and downstream (Figure 5.7). With one point of observation, we can estimate the flows of a 28 mile stretch of the Guadalupe River. The quality of the propagated flow estimates relies heavily on the length of the data assimilation window. If the window is too small, it cannot capture a meaningful change in the volume of water to be propagated. The propagation of streamflow-only assimilation also loses power as the distance increases, because it neglects contribution

from branching channels and lateral flows from surface runoff.

The propagation region of 28 miles presented here is somewhat arbitrary; the power of the propagation could be even larger. We do not have validation methods available to us at this point to confirm the accuracy of the upstream/-downstream hydrographs. There are no other streamflow gages within a reasonable propagation zone. The closest gage upstream is 35 miles and there are no downstream gages close to the study area. We can, however, improve the model by incorporating corrections in precipitation estimates over the study area to model contributions to the channel via surface runoff. This concept is explored in the next section.

5.4 Analysis of streamflow and precipitation data assimilation

As explained in Section 4.5, we incorporate precipitation corrections into the data assimilation model to allow for lateral flow contributions to the primary channel streamflow through rainfall-runoff interactions. Heavy rainfall over the study area can significantly contribute to the rising water levels in flash floods. For example, Viterbo et al. (2020) found that a slightly displaced rainfall event over a small watershed was likely a key reason why the National Water Model underestimated the magnitude of the flood event in Ellicott, Maryland in 2018.

Unluckily for flood analysis, but luckily for the citizens of Kendall county, there have been no significant flood events in the Guadalupe River study area since the TxDOT streamflow sensor was installed in November 2018. The precipitation

observed at the TxDOT sensor for its two largest recorded storm events are still very low (Figure 5.8).

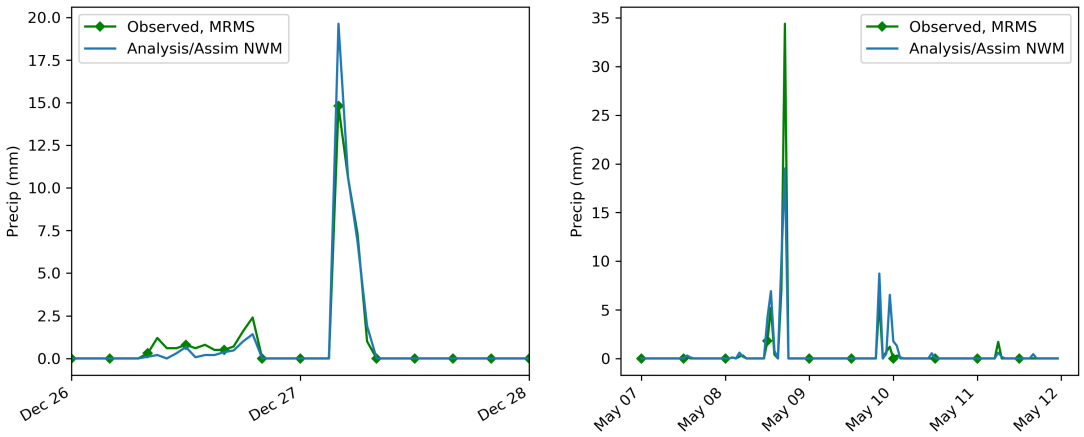

Figure 5.8: Hourly cumulative precipitation from National Water Model analysis/assimilation forcings and Multi-Radar Multi-Sensor gage-corrected QPE during the December 2018 and May 2019 storm events.

The hourly intensity in either storm event does not surpass 3.5 cm/hr. Because of the low rainfall rates, improvement to the data assimilation model through precipitation corrections will be limited. In order to produce proof of concept results for precipitation contributions to the data assimilation model, we chose to analyze a short-range NWM forecast with the most significant difference between precipitation observed and precipitation forecasted along the Guadalupe River main channel. The forecast initialized on May 8, 2019 at 10:00 fits that parameter. The NWM forecast indicated very little precipitation and lateral flows during the following 18 hours. The magnitude of the difference between forecasted and observed precipitation varies across the Guadalupe River study area (Figure 5.9).

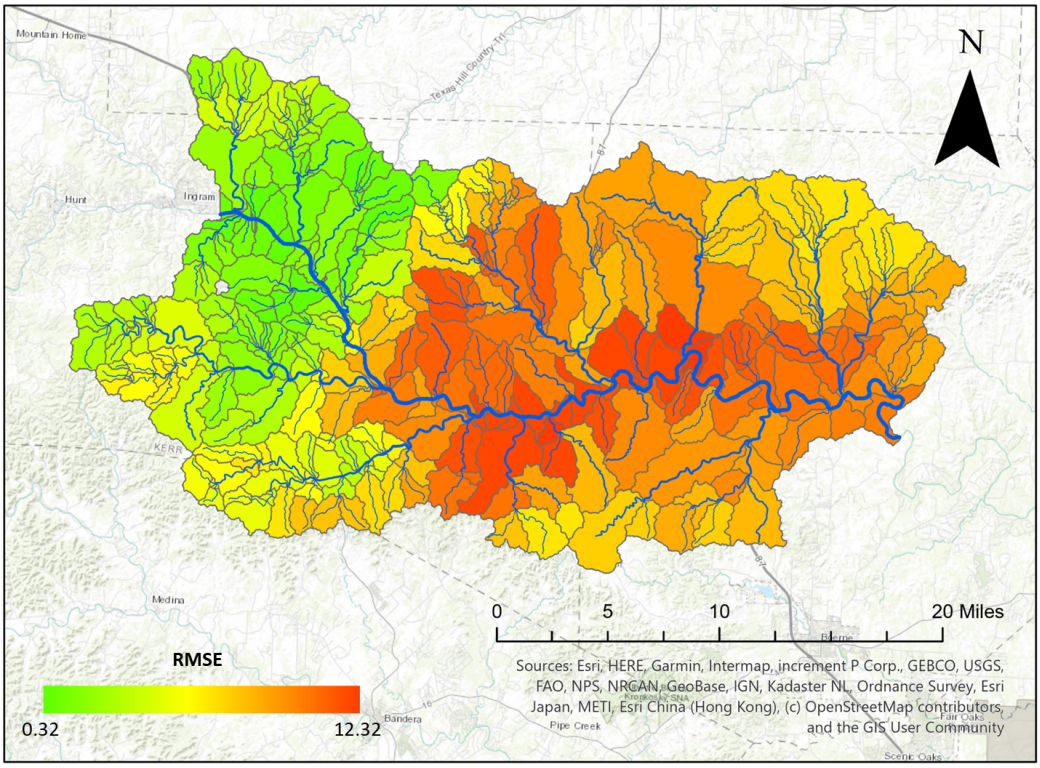

Figure 5.9: Error estimation (RSME) between forecasted and observed precipitation data over the entire Guadalupe study area for the May 8, 10:00 short-range forecast.

The precipitation forecast most closely matches the observed rainfall in the northwest region of the study area (Figure 5.9). The greatest RMSE, and therefore the greatest discrepancy between the NWM forecast and the observations, are centered near the main channel of the Guadalupe River near the streamflow sensor location.

In order to allow for more lateral flow contributions, the inflow location in the precipitation correction model is 16 miles upstream of the sensor (Figure 5.10(a)).

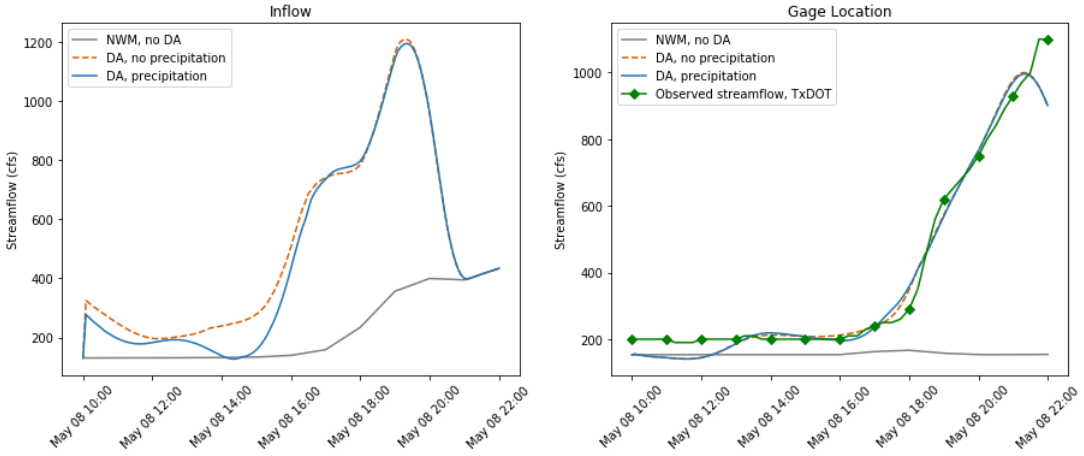

Figure 5.10: Streamflow assimilation and lateral runoff contribution results on a sample short-range forecast with a 9-hour lookback window during the May 2019 storm event.

There is a difference in the upstream flow profile when the lateral flow is accounted for (Figure 5.10(a)). The initial jump in streamflow is noticeably smaller, and the periods of recorded rainfall (Figure 5.8(a)) allow for reduced flows upstream. The streamflow profile downstream at the gage location is extremely similar between the precipitation-corrected and streamflow DA-only models, which is expected.

These positive changes in the upstream flow profile are presumed to be more influential in storm events with higher precipitation values. This correction is important, because upstream improvement is a major advantage of the 4-dimensional variational data assimilation model over sequential DA model alternatives.

Chapter 6: Conclusions

In this study, we demonstrated that a 4-dimensional variational data assimilation model improved streamflow estimation and forecasting in the Guadalupe River study area. The improved streamflow was propagated over 15 miles upstream and downstream. We also found that there are significant errors and uncertainty in National Water Model forecasts, so we proposed additional methods to advance the current modelling framework. The incorporation of precipitation correction into the model created improved streamflow profiles in the upstream channel reaches. We believe this study is a significant first step towards a NWM-based statewide flood forecasting system.

6.1 Research Question 1 Answered

Can an external modeling framework extract forecast data from the National Water Model and execute a data assimilation scheme using local streamflow data to create a more locally informed forecast? To what extent does this improve the forecasts?

In this study, we were able to establish a procedure for extracting NWM forecasts and feeding the data to an external 4D variational data assimilation model. The 4D-Var DA model was coupled with a mass-conservative Muskingum-Cunge routing scheme to assimilate data from a TxDOT streamflow sensor on the Guadalupe River. The DA/routing model is extremely computationally efficient,

requiring less than 1/100th of a second to complete a model run. The high model efficiency is ideal for real-time flood forcasting.

The DA model was evaluated at three different lookback windows—6, 9 and 12 hours. The un-assimilated NWM forecasts were evaluated for what could be considered the 12 hour lookback period. Two statistical indicators, the Nash-Sutcliffe Efficiency coefficient (NSE) and the coefficient of determination (R^2), were used to evaluate the performance of the data assimilation model and National Water Model forecasts. Across the board, the data assimilated simulations significantly out-performed the un-assimilated series. This is confirmation that the data assimilation model creates a more locally informed time series throughout the lookback period.

The simulated lookback window data scored "very good" values for the performance metrics, especially R^2, in the pre-storm and falling limb regions of the storm events. Excellent performance in these periods of the storm is an encouraging result from a flood forecasting and emergency management perspective. The longer the assimilation time, the longer the model maintained high scores in the performance metrics. The statistical indicator values dropped to unsatisfactory levels during the climbing limb and peak flow regions of the storm. We attribute the low performance scores during this period to the tendency of the NWM to under-estimate forecasts during this time and to the high emphasis the performance metrics place on outliers.

The assimilated lookback windows were re-integrated with the remainder

of the short-range 18-hour forecasts and routed downstream. In these model runs, the statistical indicator scores declined across all lookback window lengths and all storms. The pre-storm period, previously highly scored, saw a remarkable decrease in NSE value. This was contributed to the fact that many of the NWM forecasts for this period are highly variable and frequently highly overestimate the observed values.

Two approaches were proposed to help resolve the issues in the current data assimilation model. The first is to implement a temporal-interpolation method to allow the observations to carry significance beyond the assimilation window. The second is to optimally combine several forecasts to create a more stable, time-averaged forecast period.

The improved streamflow from the data assimilation model could be propagated upstream and downstream of the streamflow sensor location. In the study, the propagation zone extended up to 16 miles upstream and 18 miles downstream. The propagation zone could be extended farther, but it is assumed that the confidence in the streamflow profile decreases as the distance from the streamflow sensor increases. Additional streamflow sensors upstream and downstream are necessary to validate the propagated flows.

6.2 Research Question 2 Answered

Does adding precipitation corrections into the data assimilation model improve the forecast accuracy? To what extent?

A simplified surface runoff model was created to convert rainfall to equivalent lateral flow using the rational method and land cover data. The storms recorded by the Guadalupe River sensor were relatively small storm events with little input from precipitation over the study area. Because of this limitation, a proof of concept model run was created to illustrate the potential benefits of including precipitation corrections into the 4D variational data assimilation model.

The results showed that the profile at the upstream inflow location was noticeably improved. The initial increase of streamflow was decreased. There were also smaller channel flows throughout the assimilation window where the observed precipitation levels were elevated. The precipitation corrections were substantial, but not to the extent of the of the streamflow data assimilation. In order to more thoroughly evaluate the value of adding precipitation corrections, the model needs to be tested on bigger storm events with high precipitation intensity over the study area.

6.3 Research Question 3 Answered

Does this method of combining local gage-observed data and the National Water Model show potential for a state-wide flood warning system?

The results from this study show that there is considerable potential in assimilating local streamflow into National Water Model forecasts as a method to build a state-wide flood warning system in under-served rural areas. The methodol-

ogy creates highly correlated simulated storm forecasts to the observed streamflow in the period before the storm surge hits.

The variable nature of the full short-range NWM forecasts in the periods leading up to the storm events indicate that there is a need for more involved forecast modeling. More accurate forecast periods are crucial in developing an operational and reliable flooding warning system. Two options were proposed in this thesis for improving the forecasts. The first suggestion is to implement a temporal-interpolation scheme to phase observational data into the forecast period. Another strategy is to develop a method to optimally combine successive short-range forecasts into time-averaged, more stabilized forecast products. Further investigation of larger storm events across several study areas is necessary to get a more complete assessment of the model.

Chapter 7: Future Work

The methodology developed in this study is a promising start to an external modeling framework. The 4-dimensional data assimilation model is built to optimally combine local streamflow data into the well-established National Water Model streamflow forecasting model. In this thesis, there were two suggested improvements that could be made to the existing model framework. These improvements would allow for a more informed transition from the data assimilation period to the forecast period in the short-range forecasts.

An exciting opportunity of improvement, beyond what has already been mentioned, is to extend the region of influence wider than the existing single-channel framework. Currently, the streamflow assimilation only works on the channel on which the streamflow sensor is located. The NWM, meanwhile, has a cohesive routing and assimilation structure embedded into the complex, branching stream network that spans the United States. The next step is to incorporate the entire branching stream network into the 4D variational data assimilation model. Branching streams can contribute substantial streamflow to a primary river channel, especially during flash floods when minor channels in small watersheds can quickly turn into high velocity, high risk flooded channels. At this point, we have achieved an operational branched Muskingum-Cunge-Todini routing scheme but have not been able to develop the data assimilation model for the branching network. This is the obvious next step.

Another promising area for model development is in the flow routing scheme. The one-direction nature of Muskingum-Cunge severely limits the capacity of the 4-dimensional data assimilation model in coastal areas. Backwater and tidal interactions are significant factors in compound flooding events when both precipitation and tropical storms/hurricanes contribute to dangerous flooding conditions. A more flexible two-dimensional routing scheme would greatly improve the modelling framework and expand the regions of Texas where it can be applied.

Appendices

A Online data resources

Texas Department of Transportation sensor data

In Section 4.5 of the *Streamflow Measurement at TxDOT Bridges* technical report, there is a detailed outline for how to access the TxDOT sensor data in the KISTERS Big Data system (Maidment et al., 2019). We refer the reader to this resource for a thorough understanding of how the KISTERS database operates.

For a cursory inspection of the TxDOT sensor data, the following URL lists the available time series data for each sensor in the I-10 transect: `https://nwm.kisters.de/KiWIS/KiWIS?service=kisters&type=queryServices&datasource=0&request=getStationList&format=html&site_id=30284&addlinks=true`.
The interested reader can select the timeseries list for the desired sensor location and follow the guidelines in Maidment et al. (2019) from there.

National Water Model forecast data in the Google Cloud Platform archive

The Google Cloud Platform archive landing page can be found at: `https://console.cloud.google.com/storage/browser/national-water-model`

From there, the reader can navigate to the date and specific forecast of interest. For example, the short-range forecasts for May 1st, 2020 are located at: `https://console.cloud.google.com/storage/browser/national-water-model/nwm.20200501/short_range/`

Iowa Environmental Mesonet MRMS precipitation data archive

The entire IEM archive contains many datasets outside the MRMS system and the overall archive landing page is: `https://mtarchive.geol.iastate.edu/`.

From there, the reader can navigate to the year, month, and day of interest. The Multi-Radar Multi-Sensor data is housed in the "mrms" folder, and in that directory, you can choose from the suite of MRMS radar/precipitation products. For example, the 1-hr gage-corrected precipitation data used in this study for May 1st, 2020 are located at: `https://mtarchive.geol.iastate.edu/2020/05/01/mrms/ncep/GaugeCorr_QPE_01H/`

B Data processing for National Water Model outputs

Channel file data processing

The conversion from raw short range and analysis/assimilation channel NetCDF files covering CONUS to a single CSV file for only the stream reaches in the study area is contained in the following script.

```
# -*- coding: utf-8 -*-
"""
@author: leahh
"""

from netCDF4 import Dataset
from datetime import datetime,timedelta
import pandas as pd
import numpy as np
import xarray as xr
import os

#function to generate the NetCDF indices for your comIDs
def get_id_indices(find_ids, all_ids):
    if len(find_ids) and type(find_ids[0]) is str:
        find_ids = [int(x) for x in find_ids]
    if type(find_ids) != 'numpy.ndarray':
        find_ids = np.array(find_ids)
    if type(all_ids) != 'numpy.ndarray':
        all_ids = np.array(all_ids)
    sorted_index = all_ids.argsort()
```

```
22      sorted_nc_comids = all_ids[sorted_index]
23      found_index_sorted = np.searchsorted(sorted_nc_comids,
        find_ids)
24      index = sorted_index[found_index_sorted]
25      return index
26
27 df = pd.read_csv('GuadBranchingMC.csv') #file with Musk-cunge data
28 COMIDs = df['COMID'].tolist() #grab comIDs
29
30 #sample netcdf file
31 samplefc = Dataset(r'C:\Users\leahh\Box\Research\NWMdata\
        SR_channel_rt\June2019_storm\
        shortrange_channel_rt_2019_06_17_t08.nc','r')
32 allCOMID = samplefc.variables['feature_id']
33 allCOMID_arr = allCOMID[:]
34 index_sel = get_id_indices(COMIDs, allCOMID_arr)
35
36 #path to SR forecasts
37 path = 'C:/Users/leahh/Box/Research/NWMdata/SR_channel_rt/
        Aug2019_storm'
38
39 #path to analysis/assim forecasts
40 path_AnA = 'C:/Users/leahh/Box/Research/NWMdata/AnA_channel_rt/
        Aug2019_storm'
41
42 #output path for csv file
43 comb_path = 'C:/Users/leahh/Box/Research/NWMdata/CombinedCSVs/
        Aug2019_storm'
44
45 fnames = [i for i in os.listdir(path) if os.path.isfile(os.path.
```

```
    join(path,i)) and \
46          'shortrange_channel_rt' in i]
47 nwmvars = ['streamflow','qSfcLatRunoff','qBucket','nudge']
48 shortnames = ['flow','lat','gwBucket','nudge']
49 for file in fnames:
50     date_time_str = file[22:-3] #isolate datetime code from end of
       filename
51     t0 = datetime.strptime(date_time_str, '%Y_%m_%d_t%H')
52
53     AnAfn = 'AnA_channel_rt'+date_time_str+'.nc'
54     AnAfile = os.path.join(path_AnA,AnAfn)
55
56     timelist=[]
57     for i in range(18):
58         t = t0 + timedelta(hours=i+1)
59         timelist.append(t)
60
61     SR_data = xr.open_dataset(os.path.join(path,file), chunks = {'
      time': 18})
62     AnA_data = xr.open_dataset(AnAfile)
63     for v,sn in zip(nwmvars,shortnames):
64         SR_array = SR_data[v][:,index_sel].data.compute()
65         var_df = pd.DataFrame(data = SR_array, index=timelist,
      columns=COMIDs)
66
67         AnA_array = AnA_data[v][index_sel].values
68         var_df.loc[t0] = AnA_array  # adding a row
69         var_df = var_df.sort_index()
70
71         combined_fn= 'channel_rt'+date_time_str+'
```

```
    _GuadBranchingatI10_'+sn+'.csv'
72          export_csv = var_df.to_csv(os.path.join(comb_path,
    combined_fn))
73
74
75      print('finished coversion for ', date_time_str)
```

Forcing file data processing

The conversion from raw short range and analysis/assimilation forcing NetCDF files covering CONUS to a single CSV file for only the stream reaches in the study area is contained in the following script.

```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Thu Dec 11:22:31 2019
4
5  @author: leahh
6  """
7
8  from datetime import datetime,timedelta
9  import pandas as pd
10 import numpy as np
11 import xarray as xr
12 import os
13
14 MCdata = pd.read_csv('GuadBranchingMC_xycoords.csv') #import Musk-
       cunge data
15 x_coords = MCdata['POINT_X'].tolist() #grab lat/lons for every
```

```
    COMID in the SA
y_coords = MCdata['POINT_Y'].tolist()
COMIDs = MCdata['COMID'].tolist() #grab comid lists

# path to short range files
forcing_path = r'C:\Users\leahh\Box\Research\NWMdata\SR_forcing\
    Aug2019_storm'

# path to analysis/assim files
path_AnA = r'C:/Users/leahh/Box/Research/NWMdata/AnA_forcing/
    Aug2019_storm'

#output path
comb_path = r'C:/Users/leahh/Box/Research/NWMdata/CombinedCSVs/
    Aug2019_storm'

#grab forcing file names
fnames = [i for i in os.listdir(forcing_path) if os.path.isfile(os
    .path.join(forcing_path,i)) and \
          'shortrange_forcing' in i]

for file in fnames[18:]:
    date_time_str = file[19:-12] #isolate datetime code from end
    of filename
    AnAfn = 'AnA_forcing'+date_time_str+'.nc'

    t0 = datetime.strptime(date_time_str, '%Y_%m_%d_t%H')
    timelist=[]
    timelist.append(t0)
    for i in range(18):
```

```
40          t = t0 + timedelta(hours=i+1)
41          timelist.append(t)
42
43      dataNWM = xr.open_dataset(os.path.join(forcing_path,file))
44      AnA_data = xr.open_dataset(os.path.join(path_AnA,AnAfn))
45
46      precip_arr = np.zeros([len(x_coords),19])
47      for i in range(len(x_coords)):
48          ana_precip=AnA_data.RAINRATE.sel(x=x_coords[i],y=y_coords[
    i],method='nearest').values
49          ana_prec = float(ana_precip)
50          precip_arr[i,0] = ana_prec
51          precip = dataNWM.RAINRATE.sel(x=x_coords[i],y=y_coords[i],
    method='nearest').dropna(dim='time').values
52          precip_arr[i,1:20] = precip
53      df = pd.DataFrame(data = precip_arr,columns = timelist,index =
     COMIDs)
54      dfT = df.T
55
56      combined_fn= 'forcing_'+date_time_str+'_GuadBranching_atI10.
    csv'
57      export_csv = dfT.to_csv(os.path.join(comb_path,combined_fn))
```

C Data processing for Multi-Radar Multi-sensor precipitation files

The data processing for the MRMS files was a more involved process. The procedure is as follows:

1. Retrieve the highly compressed GRIB2 files from the Iowa State mesonet archive.

2. Create a batch file with the following script to convert the compressed GRIB2 files to NetCDF files. The batch script utilizes the command prompt package Degrib from the NWS to handle GRIB2 data.

```
# -*- coding: utf-8 -*-
"""
@author: leahh
"""
from datetime import datetime,timedelta
import numpy as np
import os

def MRMSfilenames(begin_date,number_of_days):
    #INPUTS same as first function in
   DataRetrieval_MRMS_iastate
    #OUTPUTS: (1) Retrieval_names: nams of the .grib2.gz
   files created in previous script
    #         (2) extracted_names: intermediate names created
    by degrib (standardized, do not touch)
    #         (3) nc_names: filenames for .nc final products
    #         (4) bat_name: name of the batch file you are
   making

```

```
16      t1 = datetime.strptime(begin_date, '%Y%m%d')
17      datestoget = [t1+n*timedelta(days=1) for n in range(
        number_of_days)]
18
19      bat_name = 'GZtoNC_'+datetime.strftime(datestoget[0], '%b
        %d')+'to'+datetime.strftime(datestoget[-1], '%b%d%Y')
20      # this is the name for the batch file you are creating
21      print(bat_name)
22
23      retrieval_names = np.empty([len(datestoget),24], dtype =
        object)
24      extracted_names = np.empty([len(datestoget),24], dtype =
        object)
25      nc_names = np.empty([len(datestoget),24], dtype = object)
26
27      for index,date in enumerate(datestoget):
28          datestring = 'MRMS_GaugeCorr_QPE_'+datetime.strftime(
            date, '%Y_%m_%d')
29          extract_datestring = 'MRMS_GaugeCorr_QPE_01H_00.00_'+
            datetime.strftime(date, '%Y%m%d')
30          namelist=[]
31          extract_namelist=[]
32          nc_namelist=[]
33          for hour in range(24):
34              hourstring = '_t{:02d}'.format(hour)+'.grib2.gz'
35              extract_hourstring = '-{:02d}'.format(hour)+'
                0000.grib2'
36
37              filename = datestring+hourstring
38              extractedname = extract_datestring+
```

```
    extract_hourstring
39              ncfilename = filename[:-8]+'nc'
40
41              namelist.append(filename)
42              extract_namelist.append(extractedname)
43              nc_namelist.append(ncfilename)
44
45          retrieval_names[index,:] = namelist
46          extracted_names[index,:] = extract_namelist
47          nc_names[index,:] = nc_namelist
48      return(retrieval_names,extracted_names,nc_names,bat_name)
49
50 #execute the function above:
51 files,extracted_names,nc_fns,bat_fn = MRMSfilenames('20181226
    ',4)
52
53 ## the process below creates the lines needed to access
    tcldegrib tool and decompress and convert files from .
    grib2.gz to .nc
54 ## uses the outputs from the above function
55
56 #path for location of MRMS .grib2.gz files
57 tmpdir = r'C:\Users\leahh\Box\Research\PrecipModels\MRMSfiles
    \MRMS_retrieved'
58
59 #batch filename with extension
60 #NOTE: for a trial run you can attach '.txt' instead so you
    can see if the lines are written correctly
61 bat_file = bat_fn+'.bat'
62
```

```
#batch path+name
dl_file = os.path.join(tmpdir, bat_file)
print(dl_file)

#operation to actually write the content of the .bat file
with open(dl_file, 'w') as f:
    #beginning lines to setup the environment you need
    #root is the location of Anaconda3 installation
    #batfile is the path to the anaconda batch script "activate"
    #finally call 'root' and 'batfile'
    f.write('@echo off \nset root='+'"'+r'C:\\Users\\leahh\\Anaconda3'+'"'+'\n'+'set batfile='+'"'+r'C:\\Users\\leahh\\Anaconda3\\Scripts\\activate.bat'+'"'+'\n'+'call %batfile% %root% \n')

    num_days = files.shape[0]
    num_hrs = files.shape[1]

    for day in range(num_days):
        for hour in range(num_hrs):
            file = files[day,hour]
            extract_file = extracted_names[day,hour]
            line = r'C:\Users\leahh\Anaconda3\Scripts\patool extract '+file
            newfn = file[:-8]+'nc'
            line2 = r'C:\ndfd\degrib\bin\degrib ' + '"'+extract_file+'"' + ' -C -msg 1 -NetCDF 1 -out ' + '"'+newfn+ '"' + ' -namePath ' +'"'+ r'C:\Users\leahh\Box\Research\PrecipModels\MRMSfiles\MRMS_retrieved'+'"'
```

```
            f.write(line +'\n')
            f.write(line2+'\n')
```

3. Run the batch script in a command prompt.

4. Run the following python script to grab the precipitation data from the latitude/longitude coordinates of your study area stream reaches from the output projection coordinate system of the NetCDF file.

```
# -*- coding: utf-8 -*-
"""
@author: leahh
"""

from netCDF4 import Dataset
import numpy as np
import xarray as xr
import pyproj
import pandas as pd
import os
from datetime import datetime,timedelta

#Import latitude and longitude point data in negative degrees
    for longitude

MC_df = pd.read_csv(r'C:\Users\leahh\Box\Research\NWMdata\
   GuadBranchingMC.csv')
lats = MC_df['Latitude'].tolist()
lons = MC_df['Longitude'].tolist()
COMIDs = MC_df['COMID']

```

```
#Get the projection function that converts the latitude, longitude coordinates to the projection cell number
PCS2=pyproj.Proj("+proj=eqc +lat_1=20 +lat_2=55 +lon_1=-60 +lon_2=-130 +x_0=0 +y_0=0 +a=6378160.0 +b=6356684.713804713 +units=m")

files_path = r'C:\Users\leahh\Box\Research\PrecipModels\MRMSfiles\MRMS_retrieved'
fnames = [i for i in os.listdir(files_path) if os.path.isfile(os.path.join(files_path,i)) and \
        'MRMS_GaugeCorr_QPE_2019_08' in i and '.nc' in i]

long_max = 299.994998; lat_max=54.995000
precip_arr = np.zeros([len(lats),len(fnames)])
timelist = []
for i,file in enumerate(fnames):
    date_time_str = file[19:-3] #isolate datetime code from end of filename
    t = datetime.strptime(date_time_str, '%Y_%m_%d_t%H')
    timelist.append(t)

    precipdata = xr.open_dataset(os.path.join(files_path,file))
    varlist = precipdata.variables

    for j in range(len(lats)):
        x_init,y_init = PCS2(long_max-lons[j],lat_max-lats[j])
        x_meter =x_init/1113.119; y_meter = y_init/1113.119
        x_cell = 7000-x_meter; y_cell = 3500-y_meter
```

```
43
44          #select the precipitation value corresponding to the
    nearest cell
45          precip_pt = precipdata[precipvar].sel(XCells=x_cell,
    YCells=y_cell,method='nearest').values
46          precip_arr[j,i] = precip_pt
47 df = pd.DataFrame(data = precip_arr,columns = timelist,index
    = COMIDs)
48 dfT = df.T
```

D Data Assimilation + Performance Metrics script

```
# -*- coding: utf-8 -*-
"""
Created on Wed Apr 22 14:11:04 2020

@author: leahh
"""

import numpy as np
import pandas as pd
from datetime import datetime, timedelta
from netCDF4 import Dataset
from matplotlib import pyplot as plt
import time
import muskingum
import muskingum_util_edited_dirk
import Hydro_PerfMetricFuncts
from matplotlib import dates as mdates
import Preprocessing
from scipy.optimize import minimize
from scipy.optimize import check_grad
import os

#link to NWM AnA + SR csv files for the given storm
#storm_path = r'C:/Users/leahh/Box/Research/NWMdata/CombinedCSVs/
    Aug2019_storm/sub_flows'
storm_path = r'C:\Users\leahh\Box\Research\NWMdata\CombinedCSVs\
    Branching_channel_rt\Sub_flows'

```

```
#list of NWM fc files
fnames = [i for i in os.listdir(storm_path) if os.path.isfile(os.
    path.join(storm_path,i)) and \
          'channel_rt' in i]

filenum = len(fnames)

#use a smaller length if you are having stability issues to divide
     long channels
critical_length = 20000
epsilon_limit = 0.2
baseline_flow = 0
theta = 0.5
lbwindows = [6,9,12] #number of hours for the lookback windows

#initialize the performance metric matrices
NSE_lb = np.zeros([filenum,3])
NSE_SR = np.zeros([filenum,3])
MBE_lb = np.zeros([filenum,3])
MBE_SR = np.zeros([filenum,3])
CC_lb = np.zeros([filenum,3])
CC_SR = np.zeros([filenum,3])
NWM_eval = np.zeros([filenum,3])
timestamps = []

#loop through every forecast
for fnum,file in enumerate(fnames):
    date_dt_str = file[10:24]   #isolate datetime in filename
    date_dt = datetime.strptime(date_dt_str, '%Y_%m_%d_t%H')
    timestamps.append(date_dt)
```

```
55      for i in range(len(lbwindows)):
56          lb_time = lbwindows[i] # select lookback period (hrs)
57
58          #get the streamflow data from the csv files
59          flows = pd.read_csv(os.path.join(storm_path,file))
60          flows.rename(columns = {'Unnamed: 0':'time'}, inplace =
    True)
61
62          flows['time'] = pd.to_datetime(flows['time'])
63          flows.set_index('time', inplace=True)
64          flows = flows/0.028316847   #convert from m3/s to cfs
65
66          #resample to 5 minute intervals (same in NWM WRF-Hydro)
67          flows = flows.resample('300S').interpolate(method='linear'
    )
68
69          #generate Muskingum Cunge parameters from separate script
70          MCdf = Preprocessing.MC_data_structure(r'C:\Users\leahh\
    Box\Research\NWMdata\GuadFinalMC.csv',critical_length)
71
72          #read observational data from TxDOT sensor
73          txdot = pd.read_csv(r'C:\Users\leahh\Box\Research\Sensor
    Data\KISTERS_txdot\Guadelupe_May2019_storm_UTC_507to512.csv',
    delimiter=';',header=None)
74          txdot.columns = ['time','Q_cfs']
75          txdot['time'] = pd.to_datetime(txdot['time'])
76          txdot.set_index('time', inplace=True)
77          txdot = txdot.resample('300S').interpolate(method='linear'
    )
78
```

```
79        txsensorCOM = 3589508 #COMID of sensor location
80        Qout_init = flows[str(txsensorCOM)]
81
82        sensor_ind = 33 #index within the system of sensor
    location
83        up_ind = 26 #index within the system of upstream location
84
85        MC_in = MCdf['COMID'][up_ind] #location of Qin upstream
86
87        Qin_orig = flows[str(MC_in)] #unassimilated upstream
    series
88        MCdf_upstream = MCdf.iloc[up_ind:sensor_ind].copy()
89
90        #export MC data
91        export_csv = MCdf_upstream.to_csv('GuadMC_ratingcurve_inp.
    csv', index = None, header=True) #used for rating curve
    function
92
93        lengths = MCdf_upstream['Length (m)']
94        len_of_channel=sum(lengths)*0.621371*10**-3 #converting
    length of channel to miles
95        #print('length of channel with improved flow: ',
    len_of_channel, 'miles')
96
97        COMIDs = MCdf_upstream['COMID'] #COMID list in numerical
    form
98        COMIDstr = [str(i) for i in COMIDs] #COMID list as strings
99
100       t0 = flows.index[0] #beginning of fc and assimilation
    window
```

```
101         tf_sel = t0 + timedelta(hours=lb_time) #end of
    assimilation window
102         tf = flows.index[-1] #end of forecast for the USGS
    selection
103
104         flows_DA = flows[(flows.index >= t0) & (flows.index <=
    tf_sel)] #the portion of the fc dedicated to DA
105         flows_DA_up = flows_DA[COMIDstr] #selection from the flows
     df for the selected COMIDs
106         flows_up = flows[COMIDstr] #NWM simulated flow for the
    selected COMIDs for the entire fc
107         flows_sensor = flows[str(txsensorCOM)].tolist()
108
109         txdot_DA = txdot[(txdot.index >= t0) & (txdot.index <=
    tf_sel)] #portion of USGS observations dedicated to DA
110         txdot_up =  txdot[(txdot.index >= t0) & (txdot.index <= tf
    )] #portion of USGS observations during the entire 18 hour fc
111
112         #calculate performance metrics for unassimilated NWM data
113         if i == 0:
114             NWM_eval[fnum,0] =  Hydro_PerfMetricFuncts.NSE(
    flows_up[str(txsensorCOM)],txdot_up['Q_cfs'].tolist())
115             NWM_eval[fnum,1] = Hydro_PerfMetricFuncts.MBE(flows_up
    [str(txsensorCOM)],txdot_up['Q_cfs'].tolist())
116             CC_NWM = np.corrcoef(flows_up[str(txsensorCOM)].tolist
    (),txdot_up['Q_cfs'].tolist())[0,1]
117             NWM_eval[fnum,2] = float(CC_NWM)
118
119         rc = muskingum_util_edited_dirk.rating(len(MCdf_upstream),
    'GuadMC_ratingcurve_inp.csv')
```

```
120         rc[:,3,:] = np.minimum(rc[:,3,:], epsilon_limit)
121         ntime = len(flows_DA_up)
122         dt = 300 # timestep in measure of seconds
123         nbranch = len(MCdf_upstream)
124         indx = muskingum.get_indx()
125         nseries = max([max(indx['inp'].values()), max(indx['state'
    ].values()), max(indx['out'].values())])+1
126         value = np.zeros((ntime, nbranch, nseries), dtype= np.
    double)
127         adjoint = np.zeros((ntime, nbranch, nseries), dtype= np.
    double)
128 
129         startflows = flows_DA_up.iloc[0].copy()
130 
131         #set the inflow to the value array
132         value[:,0,indx['inp']['QBC']] = flows_DA_up[COMIDstr[0]]
133 
134         #Set the initial flows
135         value[0,:,indx['state']['QI']] = startflows
136         value[0,:,indx['state']['QO']] = startflows
137 
138         indx_par_int = muskingum.get_indx_par_int()
139         npar_int = max(indx_par_int.values())+1
140         par_int = np.zeros((nbranch, npar_int), dtype= np.int)
141         par_int[:,indx_par_int['INDX_DOWN']] = np.arange(1,nbranch
    +1, dtype=np.int)
142         par_int[-1,indx_par_int['INDX_DOWN']] = -1
143         par_int[:,indx_par_int['MODE']] = 3
144 
145         indx_par_dbl = muskingum.get_indx_par_dbl()
```

```
        npar_dbl = max(indx_par_dbl.values())+1
        par_dbl = np.zeros((nbranch, npar_dbl), dtype= np.double)
        par_dbl[:,indx_par_dbl['K']] = 15000.0
        par_dbl[:,indx_par_dbl['M']] = 1.0
        par_dbl[:,indx_par_dbl['EPS']] = 0.25
        par_dbl[:,indx_par_dbl['THETA']] = theta

        # model simulation
        start = time.time()
        muskingum.simulate(value, par_int, par_dbl, rc, dt)

        # adjoint mode
        start = time.time()
        muskingum.adjoint(value, adjoint, par_int, par_dbl, rc, dt
    )

        Qobs = txdot_DA['Q_cfs'].tolist()
        QI = flows_DA_up[str(MC_in)]

        w_QI = 0.1
        w_QO = 1.0

        def cost_function(QI_DA):
            global value

            # simulation
            value[:,0,indx['inp']['QBC']] = QI_DA
            muskingum.simulate(value, par_int, par_dbl, rc, dt)

            # cost function
```

```
175             f = 0.0;
176             for it in range(ntime):
177                 f += w_QO * (Qobs[it] - value[it,nbranch-1,indx['
    state']['QO']])**2
178                 f += w_QI * (value[it,0,indx['inp']['QBC']] - QI[
    it])**2
179             return f
180 
181         def cost_function_derivative(QI):
182 
183             global value
184             global adjoint
185 
186             # simulation
187             value[:,0,indx['inp']['QBC']] = QI
188             muskingum.simulate(value, par_int, par_dbl, rc, dt)
189 
190             # initialization of adjoint
191             adjoint = np.zeros((ntime, nbranch, nseries), dtype=
    np.double)
192 
193             # cost function derivative
194             for it in range(ntime):
195                 adjoint[it,nbranch-1,indx['state']['QO']] = w_QO *
     -2.0*(Qobs[it]-value[it,nbranch-1,indx['state']['QO']])
196                 adjoint[it,0,indx['inp']['QBC']] = w_QI * 2.0*(
    value[it,0,indx['inp']['QBC']] - QI[it])
197 
198             # adjoint
199             muskingum.adjoint(value, adjoint, par_int, par_dbl, rc
```

```
, dt)
200
201             return adjoint[:,0,indx['inp']['QBC']]
202
203
204         f = cost_function(QI)
205         ###print('cost function value f = ' + str(f))
206
207         QI_DA = minimize(cost_function, QI, method='BFGS', jac=
    cost_function_derivative,
208                          options={'gtol': 1e-6, 'maxiter': 500, '
    disp': True})
209         ###print('cost function value f = ' + str(QI_DA.fun))
210
211         Qsensor_DA = value[:,nbranch-1,indx['state']['QO']].copy()
212         Qin_DA = QI_DA.x
213         Qsensor_NWM = flows_DA_up[str(txsensorCOM)]
214
215         graphname = 'Guad_DA_'+date_dt_str+'lb'+str(lb_time)+'hour
    .png'
216
217         ###print('accuracy of derivative')
218         ###print(check_grad(cost_function,
    cost_function_derivative, QI, epsilon=1e-4))
219
220         NSE_lb[fnum,i] = Hydro_PerfMetricFuncts.NSE(Qsensor_DA,
    Qobs)
221         MBE_lb[fnum,i] = Hydro_PerfMetricFuncts.MBE(Qsensor_DA,
    Qobs)
222         CC_1 = np.corrcoef(Qsensor_DA,Qobs)[0,1]
```

```
223         CC_lb[fnum,i] = float(CC_1)
224         ### Moving on to complete SR forecast!!
225 
226         flows_up[str(MC_in)][0:len(Qin_DA)]=Qin_DA
227         Qin_new = flows_up[str(MC_in)]
228 
229         ntime = len(flows_up)
230         dt = 300 # timestep in measure of seconds
231         nbranch = len(MCdf_upstream)
232 
233         rc = muskingum_util_edited_dirk.rating(len(MCdf_upstream),
    'GuadMC_ratingcurve_inp.csv')
234         rc[:,3,:] = np.minimum(rc[:,3,:], epsilon_limit)
235 
236         indx = muskingum.get_indx()
237         nseries = max([max(indx['inp'].values()), max(indx['state'
    ].values()), max(indx['out'].values())])+1
238         value = np.zeros((ntime, nbranch, nseries), dtype= np.
    double)
239         adjoint = np.zeros((ntime, nbranch, nseries), dtype= np.
    double)
240 
241         startflows = flows_up.iloc[0].copy()
242 
243         #set the inputs to the value array
244         value[:,0,indx['inp']['QBC']] = flows_up[COMIDstr[0]]
245         value[0,:,indx['state']['QI']] = startflows
246         value[0,:,indx['state']['QO']] = startflows
247 
248         indx_par_int = muskingum.get_indx_par_int()
```

```
249         npar_int = max(indx_par_int.values())+1
250         par_int = np.zeros((nbranch, npar_int), dtype= np.int)
251         par_int[:,indx_par_int['INDX_DOWN']] = np.arange(1,nbranch
    +1, dtype=np.int)
252         par_int[-1,indx_par_int['INDX_DOWN']] = -1
253         par_int[:,indx_par_int['MODE']] = 3
254 
255         indx_par_dbl = muskingum.get_indx_par_dbl()
256         npar_dbl = max(indx_par_dbl.values())+1
257         par_dbl = np.zeros((nbranch, npar_dbl), dtype= np.double)
258         par_dbl[:,indx_par_dbl['K']] = 15000.0
259         par_dbl[:,indx_par_dbl['M']] = 1.0
260         par_dbl[:,indx_par_dbl['EPS']] = 0.25
261         par_dbl[:,indx_par_dbl['THETA']] = theta
262 
263         # model simulation
264         muskingum.simulate(value, par_int, par_dbl, rc, dt)
265 
266         # adjoint mode
267         muskingum.adjoint(value, adjoint, par_int, par_dbl, rc, dt
    )
268 
269         Qsensor_DA_fullSR = value[:,nbranch-1,indx['state']['QO']]
270         #print('length = ',len(Qsensor_DA_fullSR))
271         NSE_SR[fnum,i] = Hydro_PerfMetricFuncts.NSE(
    Qsensor_DA_fullSR,txdot_up['Q_cfs'].tolist())
272         MBE_SR[fnum,i] = Hydro_PerfMetricFuncts.MBE(
    Qsensor_DA_fullSR,txdot_up['Q_cfs'].tolist())
273         CC = np.corrcoef(Qsensor_DA_fullSR.tolist(),txdot_up['
    Q_cfs'].tolist())[0, 1]
```

```
274         CC_SR[fnum,i] = float(CC)
275     #CC_SR[i] = Hydro_PerfMetricFuncts.CCsquared(Qsensor_DA_fullSR
    ,txdot_up['Q_cfs'].tolist())
276
277 print(NSE_SR)
278 dataseries = np.hstack((NSE_lb, MBE_lb, CC_lb, NSE_SR, MBE_SR,
    CC_SR, NWM_eval))
279 colname = ['NSE_lb6','NSE_lb9','NSE_lb12','MBE_lb6','MBE_lb9','
    MBE_lb12',\
280     'CC_lb6','CC_lb9','CC_lb12','NSE_SR6','NSE_SR9','NSE_SR12',\
281     'MBE_SR6','MBE_SR9','MBE_SR12','CC_SR6','CC_SR9','CC_SR12','
    NWM_NSE','NWM_MBE','NWM_CC']
282 perfdf =pd.DataFrame(data=dataseries,columns=colname,index=
    timestamps)
283 exp_csv = perfdf.to_csv(r'C:\Users\leahh\Box\Research\Thesis_Work\
    PerformanceFigures\PerformanceMetricsMayStorm.csv')
284 plt.plot(flows_up.index,txdot_up['Q_cfs'].tolist(),label='observed
    ')
285 plt.plot(flows_up.index,flows_sensor,label='NWMorig')
286 plt.plot(flows_up.index,value[:,nbranch-1,indx['state']['QO']],
    label='simulated')
287 plt.show()
```

References

Alvarado-Montero, R., Schwanenberg, D., Krahe, P., Helmke, P., & Klein, B. (2017). Multi-Parametric Variational Data Assimilation for Hydrological Forecasting. *Advances in Water Resources*, *110*(October), 182–192. Retrieved from `http://dx.doi.org/10.1016/j.advwatres.2017.09.026` doi: 10.1016/j.advwatres.2017.09.026

Austin-Petersen, A. T. (2018). *Error Assessment of National Water Model Analysis & Assimilation and Short-range Forecasts* (Unpublished doctoral dissertation). University of Texas.

Boucher, O., Randall, D., , Artaxo, P., Bretherton, C., Feingold, G., ... Zhang, X. Y. (2013). Clouds and aerosols. In P. M. Stocker, T. F. and Qin, D. and Plattner, G.-K. and Tignor, M. and Allen, S. K. and Doschung, J. and Nauels, A. and Xia, Y. and Bex, V. and Midgley (Ed.), *Climate change 2013 the physical science basis: Working group i contribution to the fifth assessment report of the intergovernmental panel on climate change* (Vol. 9781107057, pp. 571–658). Cambridge, UK: Cambridge University Press. doi: 10.1017/CBO9781107415324.016

Chao, W. C., & Lang-Ping Chang. (1992). Development of a four-dimensional variational analysis system using the adjoint method at GLA. Part 1: dy-

namics. *Monthly Weather Review*, *120*(8), 1661–1673. doi: 10.1175/1520-0493(1992)120⟨1661:doafdv⟩2.0.co;2

City of Austin. (2020). *Flood Early Warning System.* Retrieved 2020-03-06, from `https://www.austintexas.gov/department/flood-early-warning-system`

Cunge, J. A. (1969). On The Subject Of A Flood Propagation Computation Method (Muskingum Method). *Journal of Hydraulic Research*, *7*(2), 205–230. doi: 10.1080/00221686909500264

Daley, R. (1991). *Atmospheric data analysis*. Cambridge University Press, UK.

Dhakal, N., Fang, X., Cleveland, T. G., Thompson, D. B., Asquith, W. H., & Marzen, L. J. (2011). Estimation of volumetric runoff coefficients for texas watersheds using land-use and rainfall-runoff data. *Journal of Irrigation and Drainage Engineering*, *138*(1), 43–54. doi: 10.1061/(ASCE)IR.1943-4774.0000368

Evensen, G. (1994). Sequential Data Assimilation With a Nonlinear Quasi-Geostrophic Model Using Monte Carlo Methods to Forecast Error Statistics. *Journal of Geophysical Research*, *99*, 10143–10162.

Gochis, D., Barlage, M., Dugger, A., Fitzgerald, K., Karsten, L., Mcallister, M., ... Yu, W. (2018). WRF-Hydro Technical Description, (version 5.0). *NCAR Technical Note*, *(Version 5*, 107 pp. Retrieved from `https://ral.ucar.edu/sites/default/files/public/WRFHydroV5TechnicalDescription.pdf`

Greer, A. D., Wilbanks, Z. B., Clifton, L. D., Wilson, B., & Graettinger, A. J. (2018). GIS-Enabled Culvert Design: A Case Study in Tuscaloosa, Alabama. *Advances in Civil Engineering*, *2018*. doi: 10.1155/2018/4648134

Griewank, A., & Walther, A. (2008). *Evaluating Derivatives Principles and Techniques of Algorithmic Differentiation* (2nd ed.). Philadelphia, Pa: Society for Industrial and Applied Mathematics SIAM.

Harris County Flood Control District. (2020). *Harris County FWS.* Retrieved 2020-04-20, from `https://www.harriscountyfws.org/About`

Heatherman, W. J. (2012). Flood Routing on Small Streams : a Review of Muskingum-Cunge , Cascading Reservoirs , and Full Dynamic Solutions Flood Routing on Small Streams : a Review of Muskingum-Cunge , Cascading Reservoirs ,.

Hu, M., Xue, M., & Brewster, K. (2006). 3DVAR and cloud analysis with WSR-88D Level-II data for the prediction of the Fort Worth, Texas, tornadic thunderstorms. Part I: Cloud analysis and its impact. *Monthly Weather Review*, *134*, 675–698.

Ionescu, C. S., & Gogoaşe Nistoran, D. E. (2019). Influence of reservoir shape upon the choice of Hydraulic vs. Hydrologic reservoir routing method. *E3S Web of Conferences*, *85*, 1–8. doi: 10.1051/e3sconf/20198507001

Jonkman, S. N., & Kelman, I. (2005). An Analysis of the Causes and Circumstances of Flood Disaster Deaths. *Disasters*, *29*(1), 75–97. doi: 10.1111/j.0361-3666.2005.00275.x

Kalman, R. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, *82*, 35–45.

Katzfuss, M., Stroud, J. R., & Wikle, C. K. (2016). Understanding the Ensemble Kalman Filter. *The American Statistician*, *70*(4), 350–357.

Lee, H., Seo, D. J., Liu, Y., Koren, V., McKee, P., & Corby, R. (2012). Variational assimilation of streamflow into operational distributed hydrologic models:

Effect of spatiotemporal scale of adjustment. *Hydrology and Earth System Sciences*, *16*(7), 2233–2251. doi: 10.5194/hess-16-2233-2012

Lin, P., Hopper, L. J., Yang, Z. L., Lenz, M., & Zeitler, J. W. (2018). Insights into hydrometeorological factors constraining flood prediction skill during the May and October 2015 Texas Hill Country flood events. *Journal of Hydrometeorology*, *19*(8), 1339–1361. doi: 10.1175/JHM-D-18-0038.1

Lu, H., Crow, W. T., Zhu, Y., Yu, Z., & Sun, J. (2015). No Title. *EEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *8*(11), 5116–5129.

Maidment, D., Evans, H., Arctur, D., Passalacqua, P., Huling, L., Ables, M., ... McCall, A. (2019). *Streamflow Measurement at TxDOT Bridges: Final Report* (Technical Report No. 5-9054-01). Texas Department of Transportation. Retrieved from `https://library.ctr.utexas.edu/ctr-publications/5-9054-01-1.pdf`

McCarthy, G. T. (1938). The unit hydrograph and flood routing. *Proceedings of Conference of North Atlantic Division, US Army Corps of Engineers, 1938*, 608–609.

McKay, L., Bondelid, T., Dewald, T., Johnston, J., Moore, R., & Rea, A. (2019). NHDPlus Version 2: User Guide (Data Model Version 2.1). , 182. Retrieved from `https://s3.amazonaws.com/nhdplus/NHDPlusV21/Documentation/NHDPlusV2{_}User{_}Guide.pdf{%}0Aftp://ftp.horizon-systems.com/NHDplus/NHDPlusV21/Documentation/NHDPlusV2{_}User{_}Guide.pdf`

Mclaughlin, D. (2002). An integrated approach to hydrologic data assimilation: interpolation, smoothing, and filtering. *Advances in Water Resources*, *25*, 1275–1286.

McNabb, M. (2018, oct). Body Recovered After Hill Country Flooding & Latest Road Closures Info. *Texas Hill Country*. Retrieved from https://texashillcountry.com/body-recovered-flooding-latest-road-closures/{#}

Merz, R., Blöschl, G., & Parajka, J. (2006). Spatio-temporal variability of event runoff coefficients. *Journal of Hydrology*, *331*(3-4), 591–604. doi: 10.1016/j.jhydrol.2006.06.008

Montero, A. R., Schwanenberg, D., Krahe, P., Lisniak, D., Sensoy, A., Sorman, A. A., & Akkol, B. (2016). Moving horizon estimation for assimilating H-SAF remote sensing data into the HBV hydrological model. *Advances in Water Resources*, *92*, 248–257. doi: 10.1016/j.advwatres.2016.04.011

Moriasi, D., Arnold, J., Van Liew, M., Bingner, R., Harmel, R., & Veith, T. (2007). Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *ASABE*(50), 885–900.

National Centers for Environmental Information. (2020). *Billion-Dollar Weather and Climate Disasters.* Retrieved from https://www.ncdc.noaa.gov/billions/summary-stats doi: 10.25921/stkw-7w73

Niu, G.-Y. (2011). The community Noah land surface model with multiparameterization options (NoahMP): 1. Model description and evaluation with local-scale measurements. *Journal of Geophysical Research*, *116*. doi: 10.1029/2010JD015139

NWS. (2018). *Weather Related Fatality and Injury Statistics.* Retrieved from https://www.weather.gov/hazstat/

Nychka, D., & Anderson, J. L. (2010). Data Assimilation. In A. Gelfand, P. Diggle, P. Guttorp, & M. Fuentes (Eds.), *Handbook of spatial statistics* (pp. 477–

492). New York, NY: Chapman & Hall/CRC.

Office of Water Prediction. (2020). *The National Water Model.* Retrieved 2020-04-20, from http://water.noaa.gov/about/nwm

Perumal, M. (1992). The cause of negative initial outflow with the Muskingum method. *Hydrological Sciences Journal*, *37*(4), 391–401. doi: 10.1080/02626669209492603

Perumal, M., & Bhabagrahi, S. (2008). Volume Conservation Controversy of the Variable Parameter Muskingum–Cunge Method. *Journal of Hydrologic Engineering*, *134*(4), 475–485. doi: 10.1061/ΔASCEΘ0733-9429Δ2008Θ134:4Δ475

Reichle, R. H. (2008). Data assimilation methods in the Earth sciences. *Advances in Water Resources*, *31*(11), 1411–1418. Retrieved from http://dx.doi.org/10.1016/j.advwatres.2008.01.001 doi: 10.1016/j.advwatres.2008.01.001

Schwanenberg, D., & Alvarado Montero, R. (2016). Total variation diminishing and mass conservative implementation of hydrological flow routing. *Journal of Hydrology*, *539*, 188–195. Retrieved from http://dx.doi.org/10.1016/j.jhydrol.2016.05.007 doi: 10.1016/j.jhydrol.2016.05.007

Seo, D. J., Cajina, L., Corby, R., & Howieson, T. (2009). Automatic state updating for operational streamflow forecasting via variational data assimilation. *Journal of Hydrology*, *367*(3-4), 255–275. Retrieved from http://dx.doi.org/10.1016/j.jhydrol.2009.01.019 doi: 10.1016/j.jhydrol.2009.01.019

Seo, D. J., Koren, V., & Cajina, N. (2003). Real-time variational assimilation of hydrologic and hydrometeorological data into operational hydrologic forecasting. *Journal of Hydrometeorology*, *4*(3), 627–641. doi: 10.1175/

1525-7541(2003)004⟨0627:RVAOHA⟩2.0.CO;2

Sharif, H. O., Jackson, T. L., Hossain, M. M., & Zane, D. (2015). Analysis of flood fatalities in texas. *Natural Hazards Review*, *16*(1), 1–8. doi: 10.1061/(ASCE)NH.1527-6996.0000145

Shastry, A., Egbert, R., Aristizabal, F., Luo, C., Yu, C. W., & Praskievicz, S. (2019). Using Steady-State Backwater Analysis to Predict Inundated Area from National Water Model Streamflow Simulations. *Journal of the American Water Resources Association*, *55*(4), 940–951. doi: 10.1111/1752-1688.12785

Steel, R. G. D., & Torrie, J. H. (1960). *Principles and Procedures of Statistics with Special Reference to the Biological Sciences*. McGraw Hill.

Tang, B. X.-n., Knight, D. W., & Samuels, P. G. (1999). Volume Conservation in Variable Parameter Muskingum-Cunge Method. *Journal of Hydrologic Engineering*, *125*(June), 610–620.

Todini, E. (2007). A mass conservative and water storage consistent variable parameter Muskingum-Cunge approach. *Hydrology and Earth System Sciences*, 1645–1659.

USGS. (2020). *Surface Runoff and the Water Cycle.* Retrieved 2020-04-10, from `https://www.usgs.gov/special-topic/water-science-school/science/surface-runoff-and-water-cycle`

Viterbo, F., Mahoney, K., Read, L., Salas, F., Bates, B., Elliot, J., ... Cifelli, R. (2020). A Multiscale , Hydrometeorological Forecast Evaluation of National Water Model Forecasts of the May 2018 Ellicott City , Maryland Flood. *Journal of Hydrometeorology*, *21*(March 2018), 475–499. doi: 10.1175/JHM-D-19-0125.1

Wahl, T., Jain, S., Bender, J., Meyers, S. D., & Luther, M. E. (2015). Increasing risk of compound flooding from storm surge and rainfall for major US cities. *Nature Climate Change*, *5*(12), 1093–1097. doi: 10.1038/nclimate2736

Yoo, C., Lee, J., & Lee, M. (2017). Parameter estimation of the muskingum channel flood-routing model in ungauged channel reaches. *Journal of Hydrologic Engineering*, *22*, 05017005. doi: 10.1061/(ASCE)HE.1943-5584.0001507

Young, C. B., McEnroe, B. M., & Rome, A. C. (2009). Empirical Determination of Rational Method Runoff Coefficients. *Journal of Hydrologic Engineering*, *14*(12), 1283–1289. doi: 10.1061/(ASCE)HE.1943-5584.0000114

Zhang, J., Howard, K., Langston, C., Kaney, B., Qi, Y., Tang, L., ... Kitzmiller, D. (2016). Multi-radar multi-sensor (MRMS) quantitative precipitation estimation: Initial Operating Capabilities. *Bulletin of the American Meteorological Society*(April), 621–638. doi: 10.1175/BAMS-D-14-00174.1