12-2017

Advancing the Cyberinfrastructure for Integrated Water Resources Modeling

Caleb A. Buahin

Utah State University

ADVANCING THE CYBERINFRASTRUCTURE FOR INTEGRATED WATER RESOURCES MODELING

by

Caleb A. Buahin

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

of

DOCTOR OF PHILOSOPHY

in

Civil and Environmental Engineering

Approved:

Jeffery S. Horsburgh
Major Professor

David G. Tarboton
Committee Member

Bethany T. Neilson
Committee Member

Sarah E. Null
Committee Member

Steven J. Burian
Committee Member

Mark R. McLellan, Ph.D.
Vice President for Research and
Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2017

ABSTRACT

Advancing the Cyberinfrastructure for Integrated Water Resources Modeling

by

Caleb A. Buahin, Doctor of Philosophy

Utah State University, 2017

Major Professor: Dr. Jeffery S. Horsburgh
Department: Civil and Environmental Engineering

Water systems in the United States — especially those in the semi-arid western parts — are heavily human mediated because of the emphasis on managing the scarce water resources available and protecting life and property from water related hazards. While many hydrologic and hydraulic models have been developed for simulating individual components of this human-natural water system, there is rarely a single model that can be selected to investigate the increasingly multifaceted challenges we face. These challenges, including impacts of land use changes on water resources, floods, droughts, transport and fate of contaminants in the environment, etc., increasingly require integrating models that simulate different processes to make them more tractable. The research in this dissertation is an investigation of ways for achieving this model integration/coupling. An emphasis is placed on identifying those model integration strategies that enhance reproducibility of model results, flexibility of model selection and coupling configurations, and maintainability of model code bases. Using model integration application examples that investigate the interactions between the natural hydrologic system and engineered stormwater infrastructure in urban areas, evaluations were undertaken to: (1) identify and evaluate the suitability of different model integration approaches for different applications scenarios; (2) assess the numerical errors and computational efficiency of different model coupling configurations and approaches for improving model performance; (3) identify the most appropriate coupling configurations for different model integration scenarios; and (4) identify drawbacks in the design and data structures of the cyberinfrastructure for integrated water resources modeling and implement advancements to address them. Results from these evaluations are instructive for modelers in conducting model integration/coupling efficiently while ensuring that science underlying the coupling is correct. Additionally, new software is provided that provides domain specific data structures while allowing water resources modelers to take advantage of increasingly common, high performance, heterogeneous computing infrastructure.

(213 pages)

PUBLIC ABSTRACT

Advancing the Cyberinfrastructure for Integrated Water Resources Modeling

Caleb A. Buahin

Like other scientists, hydrologists encode mathematical formulations that simulate various hydrologic processes as computer programs so that problems with water resource management that would otherwise be manually intractable can be solved efficiently. These computer models are typically developed to answer specific questions within a specific study domain. For example, one computer model may be developed to solve for magnitudes of water flow and water levels in an aquifer while another may be developed to solve for magnitudes of water flow through a water distribution network of pipes and reservoirs. Interactions between different processes are often ignored or are approximated using overly simplistic assumptions. The increasing complexity of the water resources challenges society faces, including stresses from variable climate and land use change, means that some of these models need to be stitched together so that these challenges are not evaluated myopically from the perspective of a single research discipline or study domain. The research in this dissertation presents an investigation of the various approaches and technologies that can be used to support model integration. The research delves into some of the computational challenges associated with model integration and suggests approaches for dealing with these challenges. Finally, it advances new software that provides data structures that water resources modelers are more accustomed to and allows them to take advantage of advanced computing resources for efficient simulations.

ACKNOWLEDGMENTS

I dedicate this work to my wife, Cristalynn, and my children, Kyra and Dylan without whose love, support, and patience, I would not have been successful in this endeavor.

I owe a debt of gratitude to my advisor, Dr. Jeffery Horsburgh for his high academic standards and invaluable mentorship. I am forever grateful for his patience, encouragement, and the time he readily made available to me during my studies. I am very grateful to my committee members Dr. David Tarboton, Dr. Bethany Neilson, Dr. Sarah Null, and Dr. Steven Burian for their time, guidance, and counsel throughout my studies. I would also like to thank other colleagues and advisors who collaborated with me during my studies including: Dr. James Nelson, Dr. David Maidment, Dr. Venkatesh Merwade, Cassandra Fagan, Nikhil Sangwan, and Enjie Li.

I would like to express my gratitude to the Utah Water Research Laboratory and the Office of Research and Graduate Studies at Utah State University for the financial support they provided during my studies.

The research presented in this dissertation was primarily funded by the innovative Urban Transitions and Aridregion Hydro-Sustainability (iUTAH) project, a research effort sponsored by the National Science Foundation EPSCoR Grant IIA 1208732. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Caleb A. Buahin

CONTENTS

Page

CHAPTER

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Nash et al. (1990) foresaw that major problems involving the interaction of man with the hydrological environment on the global scale would increasingly require the attention of teams of scientists from many disciplines, including scientifically trained hydrologists. Despite their inconclusive level success in practice (Walther, 1987; Biswas, 2004; Kashyap, 2004; Medema et al., 2008), interdisciplinary research and integrated management frameworks that explore the complex dynamics and interactions between economic, social, engineered, and natural systems have been prescribed as the most effective way to develop sustainable natural resources (Argent, 2004). An important part of interdisciplinary research and management frameworks is the integration of mathematical models that simulate different processes to help explain the complex and often non-linear processes and feedbacks that characterize earth system dynamics. Scientists from different earth systems disciplines develop unique models to characterize dynamics within their areas of research specialty. Influences from other domains or processes are frequently approximated with simplified assumptions, simplified process formulations, or applied as external boundary conditions, without considering important feedbacks that may exist between processes or domains.

Within any earth systems science domain (e.g., hydrosphere, atmosphere), a model is typically developed for a particular purpose and has enough detail or complexity to assess the processes and scales of interest (Voinov and Shugart, 2013). This is true in the hydrology field, where model selection for a study requires a consideration of the model's ability simulate the dominant processes of interest at some desired scales to address specific study questions (Leavesley et al., 2002; Argent, 2004; Chowdhury and Eslamian, 2015). Available computing resources and model parsimony are also often considerations for model selection (Chowdhury and Eslamian, 2015; Leavesley et al., 2002). There is seldom a single model that can simulate all hydrologic processes at the disparate scales and complexities often desired for interdisciplinary or integrated water management studies. As an example, the United States Environmental Protection Agency's (USEPA) Storm Water Management Model (SWMM) is specialized for simulating urban water systems, which have networks of engineered structures including pipes, culverts, pumps, detention ponds, storage tanks, etc. However, SWMM's catchment runoff formulation is lumped, such that each sub-catchment is treated as a spatially homogenous system where internal dynamics are ignored and outputs are only provided at the outlet of a catchment. Therefore, SWMM is not ideal for applications that require explicitly modeling spatial scales smaller than the catchment scale.

Traditionally, model integration has been undertaken by extending existing models through the inclusion new process formulation code and compiling this aggregated code into a model to attain new functionality (Castronova and Goodall, 2013; Voinov and Shugart, 2013). An example of this approach, referred to as “tight integration/coupling” (Sui and Maggio, 1999), is the Coupled Ground-Water and Surface-Water Flow Model (GSFLOW) (Markstrom et al., 2008). GSFLOW is an integrated model that extends the widely-used U.S. Geological Survey (USGS) Modular Ground-Water Flow Model (MODFLOW) (McDonald and Harbaugh, 1984; Harbaugh,

2005) by incorporating a surface water hydrology component from the USGS Precipitation-Runoff Modeling System (PRMS) (Leavesley et al., 1983).

Data centric and scientific workflows in general have also been used for integrated modeling (Ludäscher et al., 2006; Miles, 2014). Workflows have emerged as a paradigm for representing and managing complex, distributed scientific computations by capturing the individual data transformations and analysis steps as well the mechanisms to carry them out in a distributed environment, furthering reproducible science (Gil et al., 2007). Workflows are ideal for automated acquisition and processing of large datasets in preparation for modeling. Workflows are also ideal for coupling models where feedbacks are limited/non-existent. This is because data exchanges within data centric workflows are often mediated by databases, which gives rise to latency costs when data needs to exchanged back and forth frequently between models. Another limitation of the data centric workflow approach is that workflows for model data preparation or analysis are sometimes tailored to specific models, which means that they cannot be easily reused with other models.

More recently, the component-based modeling or “loose integration/coupling” paradigm has been proposed (Moore and Tindall, 2005; Rizzoli et al., 2008) to generalize approaches used in these coupling exercises and provide modelers more flexibility to adapt and reuse existing model codes for different applications. Component-based modeling asserts that by breaking down complex modeling systems into independent, more functional, and more manageable units/components that can be coupled together in a generalized way, a more holistic modeling system can be obtained. With this approach to model development, a modeler can readily replace one component with another that provides a specific type of process representation in a “plug-and-play” fashion to promote experimentation (Peckham et al., 2013). Additionally, more process representations from different science disciplines can be incorporated into a modeling system. This is possible because developers are able focus on perfecting model components in their domains of specialty with the goal of eventually coupling them to models from other domains to explore feedbacks in more detail (Elag and Goodall, 2013). While the work reported here does not specifically evaluate the plug-and-play capabilities of component-based modeling systems, we sought to advance the cyberinfrastructure that would enable modelers to better take advantage of the plug-and-play capabilities.

While several component-based modeling frameworks and standards with varying degrees of complexity have been proposed for different earth systems and environmental modeling domains, including the Earth System Modeling Framework (ESMF) (Hill et al., 2004), Model Coupling Toolkit (MCT) (Warner et al., 2008), Community Surface Dynamics Modeling System (CSDMS) (Peckham et al., 2013), and the Object Modeling System (OMS) (David et al., 2002), in this dissertation the Open Modeling Interface (OpenMI) (Moore and Tindall, 2005; Gregersen et al., 2005; Gregersen et al., 2007) was adopted as a representative framework to test and advance the component-based modeling approach. OpenMI was selected because it was developed for water resources applications, it is well documented, it provides flexibility for how components must be developed, it shares similar data structures with many existing component-based modeling frameworks, and has been recently accepted as an Open Geospatial Consortium (OGC) standard (Vanecek and Moore, 2014).

Despite the advantages of loose coupling, the use of this model coupling or integration strategy can introduce some computational errors and performance penalties into a modeling system. However, these consequences have not been well studied or characterized, limiting our understanding of the potential costs and benefits of different coupling approaches. Yet, in order for model developers to select the most appropriate model integration or coupling approach needed for any study, these challenges need to be investigated and solutions found to address them. This imperative was basis for the research presented in this dissertation. To guide the research, the following objectives were identified. Each of the objectives enumerated is addressed within one or more chapters of this dissertation.

Objective 1: Identify and quantify the computational costs associated with the various model coupling strategies.

To supply the correct data from one model to another in a coupled modeling system, it is frequently necessary to perform data transformations (e.g., spatial or temporal re-gridding, sampling, or interpolation) that introduce performance costs. Additionally, mismatches in the programming languages used in models involved in a coupled modeling simulation can compound these costs because there is often an overhead when passing data from a piece of software written in one programming language to another written using a different language. It is important to quantify how these factors increase simulation times since different approaches for performing data transformations or passing data between components may determine the viability of long running simulations including those simulations that require running the same model multiple times with varied input variables (e.g., parameter estimation, uncertainty assessment, ensemble simulations, etc.). Work under this objective was aimed at quantifying the effects of different model coupling configurations within the loosely coupled, component-based modeling approach on computational times and examining the tradeoffs of different strategies for minimizing the computational penalties.

Objective 2: Implement strategies for handling coupled model boundary interactions and resolving spatial and temporal mismatches between coupled models.

Earth system scientists employ a variety of models that often operate on different spatial and temporal scales. In coupling these models, it is important to ensure that these spatial and temporal mismatches between models are resolved appropriately while ensuring that the mathematical formulations underpinning them remain correct. For example, in water resources models it is important that the relevant conservation laws are obeyed across the coupling boundaries between models. Using a tight coupling approach, it is often more straightforward to resolve spatial and temporal mismatches between two models, as linkages are explicitly handled in the model code (but are, thus, not reusable). In contrast, the loose coupling approach presumes that when two model components are coupled, either of them could be exchanged for an alternative model or process representation. Thus, resolving spatial and temporal mismatches at the boundaries between coupled models must be done in a generic and reusable way. Work under this objective focused on examining different coupling scenarios and developing and testing strategies for minimizing mass balance errors associated with boundary interactions between coupled models.

Objective 3: Advance component-based modeling frameworks to better support integrated water resources modeling applications.

It is important that frameworks selected for integrated water resources modeling applications provide applicable data structures and data exchange workflows. Additionally, like other earth systems modelers, hydrologic modelers would benefit from modeling frameworks that advance from the typical practice of running simulations on single desktop computers to high-performance computing infrastructure. Work under this objective was focused on demonstrating how the advances in component based modeling made as part of this research could be incorporated into an open source software solution for model coupling that can be used by the water resources modeling community, and, by extension, how they could be incorporated into other model coupling frameworks and software implementations.

In Chapter 2, a study representing a data centric workflow for model integration is presented that involves a flood inundation forecasting model for the Austin, Texas area. It was developed by coupling outputs from a weather forecast model to a flood inundation delineation model. This study mainly addresses Objective 2 by applying a downscaling technique to transform weather forecast datasets into runoff, which is subsequently transformed into inundation forecasts using a rating curve library. It also addresses Objective 1 to a lesser extent by identifying the computationally expensive areas of the model and implementing strategies to minimize them so that inundation forecasts can be produced in a timely fashion to mitigate loss of property and life.

In Chapter 3, the OpenMI component-based modeling interface definitions are employed in the coupling of several Environmental Protection Agency (EPA) Stormwater Management Models (SWMM) developed to represent different areas of the stormwater conveyance system of the City of Logan, Utah. These models were developed by successively decomposing a larger SWMM model into smaller spatial units and then recoupling them to determine the increases in simulation times and mass balance errors as the number of models involved in the coupling increases. This study addresses Objective 1 by identifying the bottlenecks that contribute to increased simulation times as the number of coupled models increase. Objective 2 is also addressed by devising approaches for coupling the temporal mismatches between the models while ensuring that the overall mass balance in the coupled system is minimized.

Chapter 4 addresses Objective 3 by adopting and demonstrating the HydroCouple component-based modeling framework software (Buahin and Horsburgh, 2016), which was developed by advancing OpenMI. HydroCouple provides better support for geospatial datasets than OpenMI and allows users to take advantage of high performance, heterogeneous computing resources for more efficient simulations. Chapter 4 extends the study presented in Chapter 3 by coupling a newly-developed, two-dimensional, distributed hydrologic model that simulates overland, riverine, and canal flows with a SWMM model that simulates flow in the one-dimensional engineered components of the City of Logan’s stormwater system (i.e., pipes, culverts, inlets, outfalls, etc.). It also illustrates how more efficient simulations can be undertaken by taking advantage of the parallelism that is afforded with increasingly common high-performance computing infrastructure.

References

Argent, R. M. (2004). “An overview of model integration for environmental applications—components, frameworks and semantics.” Environmental Modelling & Software, Concepts, Methods and Applications in Environmental Model Integration, 19(3), 219–234. DOI:10.1016/S1364-8152(03)00150-6

Biswas, A. K. (2004). “Integrated water resources management: a reassessment: a water forum contribution.” Water international, 29(2), 248–256. DOI:0.1080/02508060408691775

Buahin, C., and Horsburgh, J. (2016). “From OpenMI to HydroCouple: Advancing OpenMI to Support Experimental Simulations and Standard Geospatial Datasets.” International Congress on Environmental Modelling and Software, Toulouse, France.

Castronova, A. M., and Goodall, J. L. (2013). “Simulating watersheds using loosely integrated model components: Evaluation of computational scaling using OpenMI.” Environmental Modelling & Software, 39, 304–313. DOI:10.1016/j.envsoft.2012.01.020

Chowdhury, R. K., and Eslamian, S. (2015). “Climate Change and Hydrologic Modeling.” Handbook of Engineering Hydrology: Modeling, Climate Change, and Variability, CRC Press. DOI:10.1201/b16683-6

David, O., Markstrom, S., Rojas, K., Ahuja, L., and Schneider, I. (2002). “The Object Modeling System.” Agricultural System Models in Field Research and Technology Transfer, L. Ahuja, L. Ma, and T. Howell, eds., CRC Press.

Elag, M., and Goodall, J. L. (2013). “An ontology for component-based models of water resource systems.” Water Resources Research, 49(8), 5077–5091. DOI:10.1002/wrcr.20401

Gil, Y., Deelman, E., Ellisman, M., Fahringer, T., Fox, G., Gannon, D., Goble, C., Livny, M., Moreau, L., and Myers, J. (2007). “Examining the challenges of scientific workflows.” IEEE computer, 40(12), 26–34. DOI:10.1109/MC.2007.421

Gregersen, J. B., Gijsbers, P. J. A., Westen, S. J. P., and Blind, M. (2005). “OpenMI: the essential concepts and their implications for legacy software.” Advances in Geosciences, 4(4), 37–44.

Gregersen, J. B., P. J. A. Gijsbers, and S. J. P. Westen. (2007). “OpenMI: Open Modelling Interface.” Journal of Hydroinformatics, 9(3), 175. DOI:10.2166/hydro.2007.023

Harbaugh, A. W. (2005). MODFLOW-2005, the U.S. Geological Survey's Modular Ground Water Flow Model – The Groundwater Flow Process. U.S. Geological Survey Techniques and Methods, U.S. Geological Survey, Reston, Virginia.

Hill, C., DeLuca, C., Suarez, M., and Da Silva, A. (2004). "The architecture of the earth system modeling framework." Computing in Science & Engineering, 6(1), 18–28. DOI:10.1109/MCISE.2004.1255817

Kashyap, A. (2004). "Water governance: learning by developing adaptive capacity to incorporate climate variability and change." Water science and technology: a journal of the International Association on Water Pollution Research, 49(7), 141–146.

Leavesley, G. H., Lichty, R. W., Troutman, B. M., and Saindon, L. G. (1983). Precipitation-runoff modeling system: User's manual. US Geological Survey.

Leavesley, G. H., Markstrom, S. L., Restrepo, P. J., and Viger, R. J. (2002). "A modular approach to addressing model design, scale, and parameter estimation issues in distributed hydrological modelling." Hydrological Processes, 16(2), 173–187. DOI:10.1002/hyp.344

Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E. A., Tao, J., and Zhao, Y. (2006). "Scientific Workflow Management and the Kepler System: Research Articles." Concurrency and Computation: Practice and Experience, 18(10), 1039–1065. DOI:10.1002/cpe.v18:10

CHAPTER 2

PROBABILISTIC FLOOD INUNDATION FORECASTING USING RATING CURVE LIBRARIES[1]

Abstract

One approach for performing uncertainty assessment in flood inundation modeling is to use an ensemble of models with different conceptualizations, parameters, and initial and boundary conditions that capture the factors contributing to uncertainty. However, the high computational expense of many hydraulic models renders their use impractical for ensemble forecasting. To address this challenge, we developed a rating curve library method for flood inundation forecasting. This method involves pre-running a hydraulic model using multiple inflows and extracting rating curves, which prescribe a relation between streamflow and stage at various cross sections along a river reach. For a given streamflow, flood stage at each cross section is interpolated from the pre-computed rating curve library to delineate flood inundation depths and extents at a lower computational cost. In this article, we describe the workflow for our rating curve library method and the Rating Curve based Automatic Flood Forecasting (RCAFF) software that automates this workflow. We also investigate the feasibility of using this method to transform ensemble streamflow forecasts into local, probabilistic flood inundation

[1] Buahin, C.A., N. Sangwan, C. Fagan, D.R. Maidment, J.S. Horsburgh, E.J. Nelson, V. Merwade, and C. Rae, 2017. Probabilistic Flood Inundation Forecasting Using Rating Curve Libraries. JAWRA Journal of the American Water Resources Association 53:300–315.

Reproduced with permissions from the Journal of the American Water Resources Association delineations for the Onion and Shoal Creeks in Austin, Texas. While our results show water surface elevations from RCAFF are comparable to those from the hydraulic models, the ensemble streamflow forecasts used as inputs to RCAFF are the largest source of uncertainty in predicting observed floods.

2.1 Introduction

Floods present immense human and economic costs worldwide. In the United States (U.S.), inflation adjusted economic damages from floods were in excess of $500 billion for the 1980-2005 period (Lott and Ross, 2006). These damages are expected to increase in many areas around the world with land use and climate change being major drivers (Thieken et al., 2014). The Federal Emergency Management Agency's (FEMA) National Flood Insurance Program has been the primary vehicle through which the U.S. federal government encourages communities to adopt better floodplain management practices. As part of the National Flood Insurance Program, FEMA and partner communities develop Flood Insurance Studies, which include Flood Rate Insurance Maps. Flood Insurance Rate Maps depict the boundaries of areas that are most vulnerable to floods, the most common of which is the base-flood boundary – i.e., the inundation extents for the 100-year flood.

Although the National Flood Insurance Program has been instrumental in getting communities to adopt better floodplain management strategies and increasing flood insurance coverage, a single line is used to represent the flood inundation boundaries for each flood recurrence level in Flood Insurance Rate Maps. This single line does not convey the inherent uncertainties in the hydrologic and hydraulic analysis as well as the meteorological inputs used to derive them. Several researchers have recognized this shortcoming and have recommended approaches for evaluating and communicating these uncertainties to policy makers and the general public through probabilistic flood maps (Smemoe et al., 2007; Merwade et al., 2008; Jung and Merwade, 2012). Further, Flood Insurance Rate Maps provide static flood risk maps based on historical climate and hydrological conditions. They do not provide any insights on the potential impacts of future storms with sufficient lead times for targeted response planning.

An operational flood forecasting system that provides reliable flood warnings with adequate lead times will be invaluable to emergency responders and communities to help mitigate the impacts of flooding (Cloke and Pappenberger, 2009). Although there are several sources of uncertainty for hydrologic models used in flood forecasting systems, precipitation dynamics remain the largest source of uncertainty for streamflow predictions (Merwade et al., 2008). Most operational forecasting systems rely solely on weather observations or on short-term deterministic rainfall forecasts from numerical weather predictions of up to 2 days or less because the unpredictable degree of uncertainty at longer lead times renders the results unreliable and, therefore, not useful for decision-making (Demeritt et al., 2007; Thielen et al., 2009).

Increasingly, ensemble prediction systems are being used to extend flood warning lead times beyond the 48-hours typically provided by single deterministic numerical weather predictions (Thielen et al., 2009). Ensemble prediction systems are typically comprised of a Monte Carlo framework of numerical weather predictions, whose initial conditions are perturbed to generate multiple, equally likely weather realizations for the same location and time (Cloke and Pappenberger, 2009). The uncertainty surrounding these predictions can be cascaded downstream to other models when outputs from the ensemble numerical weather predictions are coupled to land surface, hydrologic, and hydraulic models.

Ensemble prediction systems such as The National Weather Service's Advanced Hydrologic Prediction Service and the European Center for Medium-Range Weather Forecasts (ECMWF) have relatively coarse spatial scale land surface or hydrological modeling components that predict runoff for catchments and streamflow for major rivers. Translating this coarse scale runoff and streamflow into local, actionable information such as inundation extents, depths, and velocities requires solving not only the mass conservation equations but also the momentum conservation equations – i.e., the Saint-Venant equations. Despite the myriad of methods proposed to solve the Saint-Venant equations, none have demonstrated stable, efficient dynamic simulations of unsteady flows in large river networks at the continental or even regional scales (Hodges, 2013). The successful application of microprocessor analysis used in the design of very-large-scale integration circuits by Liu and Hodges (2014) in solving the Saint-Venant equations is promising. Still, challenges related to characterizing river bathymetry, incorporation of reservoir operations, inclusion of hydraulic structures, prescribing boundary conditions, as well as the computational expense of running the large number of simulations required for ensemble forecasts over regional and continental scales remain. These challenges hinder the development of flood forecasting systems at the national or even regional scales capable of producing accurate and reliable forecasts of flood inundation with sufficient lead times for prompt dissemination of warnings and optimal allocation of resources for emergency response.

In this paper, we present a new method for probabilistic flood inundation forecasting that involves using an ensemble of streamflow forecasts to compute inundation depths and extents from rating curve libraries. This approach to flood inundation forecasting is computationally scalable and avoids the stability challenges and the complexity of prescribing boundary conditions for hydraulic models during forecasting. Scaling up to large river networks is achieved by parallelizing the inundation computations for each river reach in the network since the computations associated with each river reach is independent from others. Additionally, results from this rating curve library approach are comparable to the hydraulic models underlying the rating curve libraries. We assessed the feasibility of using this approach in an operational flood forecasting system by applying it to the Shoal and Onion Creeks in Austin, Texas. For our evaluation, we used streamflows produced by downscaling ECMWFs ensemble runoff forecasts produced roughly 2 days before the May 25th – 28th, 2015 flooding events that occurred in the Austin, Texas area. We also present the Rating Curve based Automatic Flood Forecasting (RCAFF) software that we have developed to automate the workflow associated with the new method.

2.2 Background

In the following sections, we present some background information on rating curves and their limitations. We also discuss the ECMWF system and their forecast products that we used in this study.

2.2.1 Rating Curves

A rating curve prescribes a one-to-one mathematical relation between stage and streamflow at a particular cross-section of a river reach. It is widely used by hydrologists to estimate streamflow from stage and vice versa. The wide adoption of rating curves is due to the fact that direct measurements of discharge in open channels is costly, time consuming, and sometimes impractical during floods (Braca, 2008). Modern approaches for developing rating curves used widely in the U.S. and worldwide have been spearheaded by the United States Geological Survey (USGS) and can be found in several technical reports, including: Corbett et al., (1943), Mitchell (1954), Dawdy (1961), Bailey and Ray (1967), and Sauer (2002). Despite frequently being treated as an empirical relation, there is a theoretical basis underpinning the use of rating curves that is often poorly understood or ignored leading to incorrect applications in practice.

A rating curve for a particular location is generated by making pairwise measurements of streamflow and stage that are then typically fitted to the power equation (Sauer, 2002):

$$Q = a + b(y + c)^d \tag{1}$$

where Q is streamflow, y is the gage height, and *a, b, c, and d* are constants. Equation 1 is valid under certain conditions of the Chézy formula (Equation 2) used to estimate average flow in an open channel (Schmidt and Yen, 2001).

$$Q = AK\sqrt{R \times S_f} \tag{2}$$

where *A* is the cross-sectional area of flow, *K is the* Chézy coefficient, *R* is the hydraulic radius, and S_f is the energy slope. The full unsteady gradually varied flow equation for the S_f is as follows:

$$S_f = S_o - \left[\frac{1}{gA}\frac{\partial Q}{\partial t} + \frac{1}{gA}\frac{\partial\left(\frac{Q^2}{A}\right)}{\partial x} + \frac{\partial y}{\partial x}\right] \tag{3}$$

where S_o is the channel bottom slope, *t* is time, *g* is acceleration due to gravity, and *x* is the longitudinal location along the channel.

Equations (2) and (3) have two implications for the requirement of one-to-one mapping of stage and streamflow for rating curves. The first is that for a rating curve to be valid, the relationship between stage and the cross-sectional area of flow must not change. In many rivers, scouring and deposition of sediment, seasonal vegetation growth within or along the banks of river channels, and ice formation within a river may alter the relationship between stage and cross-sectional area. When these conditions are present, part or all the flows of the pairwise streamflow-stage measurements of the rating curve may need to be shifted by a factor to account for the new conditions.

The second implication is that in order to have the single valued relationship between stage and streamflow required for rating curves, the energy slope must be constant. An assumption of steady uniform flow – i.e., negligible changes in streamflow with respect to time, streamflow with respect to the longitudinal direction along the channel as a result of changes in cross section, and depth with respect to the longitudinal direction along channel – satisfies this requirement. Changes in the value of the energy slope leads to a hysteresis effect known as a looped rating curve, where the same stage maps to different flows (Braca, 2008). In such rivers, the water-surface elevation tends to be higher on the falling limb of the hydrograph than on the rising limb at the same discharge (Lewis, 1998). Schmidt and Yen (2001) argue that the unsteadiness term (i.e., $\frac{\partial Q}{\partial t}$) in equation 2 will be negligibly small for nearly all commonly occurring gradually varied flow conditions compared to the other terms in equation 2. The pressure term (i.e., $\frac{\partial y}{\partial x}$) accounts for the backwater effects from tributaries, tidal estuaries, reservoirs, constrictions in the channel, and hydraulic structures (e.g., culverts, bridges, etc.). The convective term (i.e., $\frac{\partial\left(\frac{Q^2}{A}\right)}{\partial x}$) accounts for the translation, distortion, and attenuation of the flow hydrograph (Yen and Castro, 2000).

2.2.2 The European Center for Medium Range Forecasts

The European Center for Medium Range Weather Forecasts (ECMWF) is an independent intergovernmental organization comprised of several European countries that acts as a research institute as well as an operational forecasting service. ECMWF produces a wide variety of forecast products spanning the hydrology, meteorology, and oceanography fields. ECMWF, in conjunction with the Joint Research Center of the European Commission, has set up the Global Flood Awareness System, which is comprised of an integrated hydro-meteorological forecasting chain and a monitoring system that analyzes daily results and shows forecasted flood events on a dedicated web platform (Alfieri et al., 2013).

Underlying Global Flood Awareness System is ECMWF’s Integrated Forecasting System, which is comprised of a data assimilation system and a general circulation model. Outputs from the general circulation model component of the Integrated

Forecasting System include a 51-member ensemble medium range forecast with 50 perturbed members and one unperturbed deterministic control run. These ensemble members are produced at a horizontal resolution of ~32 km for 10 days, increasing to ~64 km from day 11 to 15 (Alfieri et al., 2013). The general circulation model produces output at a time step of 3 hours for the first 144 hours and at 6 hours for the rest of the duration of a simulation.

Within the ECMWF general circulation model, the ensemble numerical weather prediction simulations are coupled to the Hydrology Tiled ECMWF Scheme for Surface Exchanges over Land (H-TESSEL, Balsamo et al., 2009) land surface model. H-TESSEL computes the land surface response to atmospheric forcing; estimates the surface water and energy fluxes; and the temporal evolution of soil temperature, moisture content, and snowpack conditions (Alfieri et al., 2013). H-TESSEL also produces streamflow forecasts on a coarse gridded river network with a resolution of 0.1°. This gridded river network is generated by upscaling the vector river network from Hydrological Data and Maps Based on the Shuttle Elevation Derivatives at Multiple Scales (HydroSHEDS, Lehner et al., 2006) dataset (Alfieri et al., 2013). However, these coarse scale streamflow predictions do not provide the local scale forecasts that are needed for emergency response planning and were not used for this study. A need, therefore, exists to downscale H-TESSEL land surface model runoff into streamflow using more accurate and higher resolution river networks for routing.

2.3 Methods

The rating curve library method we developed for flood inundation forecasting involves running a series of steady state hydraulic model runs using a range of flows as inflow boundary conditions and extracting rating curves at various cross sections along river reaches into a library. The extracted rating curve at each cross section represents the relationship between stage and streamflow extracted from each of the steady state model runs. With this rating curve library, streamflow predictions from a land surface/hydrologic model can be readily used to interpolate inundation stage at each cross section. The water stage at each cross-section forms a three-dimensional line along the alignment of the cross-section. These lines were triangulated to create the water surface from which the river reach bathymetry is subtracted to estimate inundation depths and extents. For our study, we wanted to evaluate the feasibility of using this rating curve library method and ensemble streamflow forecasts derived by downscaling runoff from ECMWF’s H-TESSEL model to forecast flood inundation. The downscaling of the ECMWF-H-TESSEL runoff was accomplished by estimating the fraction of runoff generated for each catchment in the National Hydrography Dataset Plus (NHDPlus, McKay et al., 2012) dataset and routing this runoff through the NHDPlus river network using the Routing Application for Parallel Computation of Discharge (RAPID; David et al., 2011) model.

In the following sections, we describe the study area where we applied our rating curve library method and the downscaling process that was used to derive ensemble streamflows from the H-TESSEL runoff. We also describe the workflow and interpolation methods used in the RCAFF tool we developed to automate the rating curve library method. Finally, we describe the evaluations we performed to assess the performance of our rating curve library method.

2.3.1 Study Area

To test the viability of our rating curve library method for flood inundation forecasting, we applied the method to the Onion and Shoal Creeks located in Austin, Texas (Figure 2.1). We selected these watersheds for our assessment because of the availability of existing high quality hydraulic models and LiDAR data from the City of Austin. The hydraulic models were developed using the Hydrologic Engineering Centers River Analysis System (HEC-RAS; U.S. Army Corps of Engineers, 2015), which has traditionally been the widely-used model in FEMA Flood Insurance Studies. Additionally, these watersheds are located in an area that is prone to flash flooding, leading to the National Weather Service nicknaming it “Flash Flood Alley” (Lower Colorado River Authority, 2014). This vulnerability to flooding can be attributed to a combination of steep terrain, rocky soils, and intense precipitation events in the region (Colorado River Flood Guide, 2014). These factors can transform precipitation to overland flows, swelling rivers in very little time. These two watersheds were selected because they experienced significant flooding during the period that overlapped the ECMWF forecasts that were available for our study.

2.3.2 Downscaling ECMWF-HTESSEL Runoff through RAPID

To produce the local streamflow forecasts that are needed for detailed emergency response planning, Snow (2015) developed an automated workflow to downscale the

ECMWF-HTESSEL runoff into localized streamflow predictions. This was accomplished by routing the ECMWF-HTESSEL runoff through the RAPID model. The RAPID model uses a matrix based approach to efficiently solve the Muskingum (McCarthy, 1939) flood routing equations over large river networks. The computational speed-up required to run the model using high resolution river networks over large spatial scales is obtained by solving the resulting system of linear equations using a parallelized iterative solver (David et al., 2013).

Snow (2015) ran the RAPID model over the NHDPlus river network. The NHDPlus dataset incorporates features from the National Hydrography Dataset (NHD) at the medium scale resolution (1:100,000-scale), the National Elevation Dataset (NED), and the National Watershed Boundary (NWB) geospatial datasets. One of the major enhancements of NHDPlus over the original NHD dataset is that it provides the connectivity information between river reaches and their associated catchments. Each river reach and its contributing catchment is assigned the same unique identifier called the COMID. This COMID is also used to connect a river reach with is downstream reaches to create a river network.

The workflow for the ECMWF-RAPID downscaling process used by Snow (2015) proceeded by first deriving a matrix for the fraction of the area of each cell in the ECMWF gridded output that overlaps each of the NHDPlus catchments. For each time step, this matrix was used to transform the gridded ECMWF runoff into runoff for each of the NHDPlus catchments. This runoff was subsequently applied as lateral inflow into the NHDPlus reaches and then routed with the RAPID model. Outputs were produced at a 6-hour temporal resolution and saved in the NetCDF file format. The NetCDF output files were subsequently stored on an integrated Rule-Oriented Data System (iRODS; Moore, 2008) – a rule-oriented data-grid/data management middleware that gives individuals access to large amounts of geographically distributed data. iRODS presents a single virtual file system interface to a user for data that reside on distributed servers that are grouped into zones and run the iRODS server software.

2.3.3 The Rating Curve Based Automatic Flood Forecasting Tool

To streamline our rating curve library method, we developed the RCAFF tool to automate the workflow associated with the method. We developed RCAFF using the C# and python programming languages. RCAFF also uses several open source libraries for its geospatial computations. Figure 2.2 depicts the workflow used in the RCAFF tool.

Step 1 in the workflow is to provide RCAFF with a HEC-RAS model containing a series of steady state simulations (i.e., flow profiles) using inflow boundary conditions that span the range of flows anticipated on the reaches of interest. The inflow boundary conditions used to develop the rating curve library must be selected so that they can properly capture the evolution of water surfaces and inundation extents at each cross section. Using the HEC-RAS model supplied, RCAFF creates a library saved in an Extensible Markup Language (XML) file format that stores rating curves for each cross-section in the HEC-RAS model. Additionally, cutlines representing the plan view alignment for cross sections are saved as part of the rating curve library.

We selected the XML format because it is self-describing and is widely used for communicating data on the Internet. We extracted the rating curve using the Component

Object Model (COM) HECRASController library, which is included as part of the HEC-RAS installation package. This library allows modelers to access input and outputs of HEC-RAS models, change model execution modes, execute a model, and perform plotting operations. Details on how to use the application programming interface (API) for the HECRASController library are provided by Goodell (2014).

Step 2 in the RCAFF workflow involves linking the ECMWF-RAPID reaches from the NHDPlus dataset to the reaches in the rating curve library. RCAFF creates a comma separated values (CSV) mapping file with placeholders for the COMID unique identifier mapping values for each reach. This allows a modeler to assign each cross section in the rating curve library to its corresponding river reach in the NHDPlus network using the COMID unique identifier. This mapping file allows RCAFF to extract the flows to be used for the forecasting from the ECMWF-RAPID output. Additionally, a multiplier can be specified manually in the mapping file for each cross section to allow the flows extracted from the ECMWF-RAPID output to be scaled appropriately. Specifying the multiplier is necessary because the RAPID model produces streamflow forecasts at a single location at the downstream end of each reach. The multiplier for each cross section of a reach, therefore, allows a modeler to account for the decrease in streamflow as one moves upstream and less of the watershed area contributes to runoff. For our study, we estimated these multiplication factors for each cross section based on the fraction of the total catchment area contributing to runoff at that cross section. While the assignment of the cross-sections in the rating curve library to their respective reaches in the NHDPlus network and the prescription of flow multipliers for each cross-section was done manually for the application presented, it may be possible to automate these processes in a forecasting environment using many widely-used GIS software.

Step 3 in the RCAFF workflow involves combining the rating curve library and the COMID mapping file to create a forecast configuration XML file. Within the forecast file, a user must specify the digital elevation model (DEM) representing the river and floodplain bathymetry as well as the source of the streamflow forecasts that are going to be used for calculating flood inundation. RCAFF can automatically download forecasts from the integrated Rule-Oriented Data System (iRODS) as new forecasts become available using the Python-iRODSClient library. Alternatively, forecasts can be run with flow forecasts from locally available NetCDF files or time series data stored in a CSV file format.

The final step which is the forecasting process involves interpolating the water surface elevation at each cross-section for each river reach using the rating curve library and the streamflows retrieved using the COMID from the ECMWF-RAPID outputs at each time step. Next, RCAFF creates a water surface triangulation that is constrained to the cross-section cutlines using the water surface elevations interpolated for each cross-section. This triangulation of the water surface is conducted using the Triangle library (Shewchuk,1996). To speed up RCAFF’s calculations for each of the ensembles, an initial mapping is performed by assigning each pixel in the DEM representing the bathymetry of a reach to its overlapping triangle in the water surface triangulation if it exists. This mapping is used to find the overlapping water surface triangle for a pixel, which is then used to interpolate the water surface elevation for that pixel. The DEM value for the pixel is then subtracted from the interpolated water surface elevation to estimate the flood inundation depth. An inundation depth raster is calculated by repeating these steps over all the pixels in the DEM. This inundation depth raster is saved using the Geospatial Data Abstraction Library (GDAL; Warmerdam, 2008). RCAFF also produces an output raster showing the fraction of the ensembles that predicted inundation for each pixel. This raster can be interpreted as the probability of inundation if the assumption that all ensemble members are equally likely is valid.

In applying RCAFF to the Shoal and Onion Creeks, we ran their respective HEC-RAS model for 160 steady state profiles with streamflows ranging from 20 m^3/s to their 1% annual chance streamflow (i.e., ~ 703 m^3/s for Shoal Creek and ~3745 m^3/s for Onion Creek). We chose the streamflows so that the change in water surface elevations between each simulated streamflow was close to 0.5 m. The horizontal spatial resolutions of the DEMs we used for the inundation delineation were 0.3 m and 1.5 m for the Shoal Creek and Onion Creek watersheds respectively. These DEMs were developed by interpolating raw a LiDAR dataset with a vertical root-mean-square error of 10 cm supplied by the City of Austin.

2.3.4 Assessments Performed

To assess feasibility of using our rating curve library method for flood inundation forecasting, we performed three tests. For the first test, we compared streamflow from the ECMWF-RAPID ensemble forecasts to observed streamflow at the most downstream USGS gages (U.S. Geological Survey, National Water Information System. Accessed November 10, 2015, http://waterdata.usgs.gov/nwis. Unless otherwise noted all streamflow and stream stage data in this paper are from this source) for the two river reaches. This test was performed to get a sense for how well the ensemble streamflows from the ECMWF-RAPID downscaling process predicted the observed hydrographs for the storm event we considered.

In the second test, we compared the flood inundation depths and extents produced by RCAFF to those produced by the HEC-RAS model using the observed peak streamflows for the study period as inputs for both models. Since the RCAFF rating curve libraries were derived from the HEC-RAS models, we expected the differences in flood inundation depths and extents to be minor. We conducted this test to verify that our RCAFF inundation delineation procedure was just as good as the underlying hydraulic model from which we derived the rating curve library. We quantified the agreement between the two flood inundation extents with a fitness index (Bates and De Roo, 2000; Alfieri et al., 2014), defined as the ratio of inundation area commonly predicted by both maps to the union of inundation areas predicted by them.

In the final test, we compared the water surface elevations from RCAFF to observed water surface elevations at the most downstream USGS gages on the river reaches evaluated. This comparison was done to get a sense for how good flood inundation for the storm event could have been predicted with the ECMWF-RAPID forecasts produced two days before the storm occurred. Ideally, the assessment of the performance of flood inundation prediction should also include the evaluation of its predicted inundation extent against the observed flood extents. However, observed flood extents were not available for the period we evaluated for which ECMWF-RAPID outputs were available.

2.4 Results and Discussions

Figures 2.3 and 2.4 show the results of our first test, where we compared streamflows from the ECMWF-RAPID ensemble forecasts to observed streamflows for the downstream USGS gages on the Onion and Shoal Creeks respectively. Overall, the ECMWF-RAPID ensembles largely under predicted the peak observed streamflows at the USGS gages. This under prediction of the peak streamflow was worse for the smaller Shoal Creek watershed than for the larger Onion Creek watershed. Additionally, the ECMWF-RAPID ensembles were not able to resolve the rapid change in the rising limb of the hydrograph as well as the smaller time scale fluctuations in the observed streamflow. In addition to general errors that result from the selection of model parameters, initial/boundary conditions, and model structure, the inconsistency between the model results and the observed streamflows from the USGS gages may be attributed to: 1) the inability to capture some of the sub-grid dynamics due to coarseness of the computational grid and time step of the ECMWF general circulation model and H-TESSEL, 2) the downscaling process may have smoothed out some of the local scale dynamics, and 3) hydraulic structures including dams, reservoirs, etc., that can influence the predicted streamflows were not incorporated in the RAPID model. Both Onion and Shoal Creek are traversed at various points along their lengths by bridges. Additionally, Onion Creek has a few dams along its length. While the hydraulic models used to derive rating curves for both Onion and Shoal Creeks incorporated major hydraulic structures, these hydraulic structures were not included in the RAPID model used for deriving the downscaled streamflows.

The results of our second test, where we compared the inundation depths and extents generated by RCAFF to those from HEC-RAS using the peak streamflows observed for Shoal and Onion Creek as inputs for both models are shown in Figure 2.5 and Figure 2.6 respectively. The results showed good agreement between the two approaches with F values of 95% to 96% for the Shoal and Onion Creek respectively. The mean difference in the water surface elevation predictions by the two approaches was found to be only 0.06 m and 0.09 m for Shoal and Onion Creek respectively. The minor differences in the two approaches can be attributed to the errors from the linear interpolation between data points on the rating curve. Overall, as expected, the RCAFF outputs were comparable to the underlying HEC-RAS model from which we derived the rating curve library while providing savings on the computation time required to run each of the ensemble members. For example, in the case of Onion Creek, RCAFF took ~10 minutes (16 minutes total if the inundation probabilities are calculated) for each time step for the 51 ensemble members versus ~130 minutes for the manual execution of the HEC-RAS model and the subsequent HEC-GeoRAS delineation operations.

It is important to note that even though we were unable to obtain the original bathymetric data that was used to delineate the cross-sections for the hydraulic models from the City of Austin, we made sure that the projection system and vertical datum of the original bathymetric data were the same as the DEM data we interpolated from the LiDAR data that were used for the final inundation delineation for both the HEC-RAS output and the rating curve library approach. Additionally, the same cross sections from the hydraulic models were used in developing the rating curve libraries, providing a common basis for the comparison of the two approaches.

Since the ECMWF-RAPID streamflow forecasts were poor for Shoal Creek, we proceeded with our inundation forecasting using only Onion Creek. Figure 2.7 shows a comparison of the rating curve library derived from the HEC-RAS model versus the rating curve obtained from the field measurements made during the October 2013 to November 2015 period at the USGS gage 08159000. The rating curves compared well for streamflows below ~2000 m^3/s, which is much larger than the peak streamflow observed during the May 25, 2015 floods.

Figure 2.8 shows the inundation depth map produce by RCAFF for streamflow from the ECMWF-RAPID ensemble member with the highest peak flow of 252.8 m^3/s at 5/26/2015 6:00 A.M. Figure 2.9 shows the inundation probability map at the same time. Figures 2.10 and 2.11 show comparisons of the water surface elevations of the ensemble members from RCAFF to the two USGS gages 08158827 and 08159000, located upstream and downstream on the Onion Creek respectively. The RCAFF tool generally under predicted water surface elevations because the ECMWF-RAPID outputs under predicted streamflows for the storm event we evaluated. However, for the USGS gage 08159000, located downstream on Onion Creek, RCAFF predicted higher water surface elevations than were observed on the rising limb of the hydrograph. A plot of the observed streamflow versus stage shown in Figure 2.12 indicated that a looped rating curve condition occurred at that USGS gage perhaps due to the rapid change in flow on the rising limb of the observed hydrograph or downstream backwater conditions. From a risk assessment perspective, this over prediction of water surface elevation on the rising limb of the hydrograph for flows with variable energy slope is conservative.

At USGS gage 0815900, for those ensemble members whose streamflow values were comparable to the observed values on the recession limb of the hydrograph, the water surface elevations compared favorably with the observed water surface elevations. For instance, at 5/26/2015 6:00 A.M., the streamflow for ensemble 47 was 252.8 m^3/s versus an observed value of 260.8 m^3/s. The resulting water surface elevation for ensemble 47 was 141.0 m versus an observed value of 140.8 m.

2.5 Conclusions

We have presented in this paper, a new a rating curve library method for flood inundation delineation as an alternative to running full-blown hydraulic models in operational flood forecasting settings. The proposed approach can serve as an intermediary step until the challenges associated with solving the force based equations underlying hydraulic models over the regional and continental scales are addressed. We have also created the software tool, RCAFF, to automate the workflow associated with our approach. RCAFF treats each river reach as an independent element, placing its computations into the so called “embarrassingly parallel” class of algorithms. This property can be leveraged to scale the RCAFF tool to many river reaches by parallelizing the computations on high performance computing systems.

The interpolated water surface elevations from the RCAFF tool were as good as results from the hydraulic models from which we generated the underlying rating curve library. In the United States, there exists a vast number of calibrated hydraulic models that have been developed by several local jurisdictions to comply with National Flood Insurance Program requirements. These existing models are detailed incorporating backwater effects from hydraulic structures including bridges, culverts, and weirs. These models can be adopted to develop a local flood forecasting system using the RCAFF tool we have developed.

The rating curve library approach, however, has some limitations. Unstable cross sections arising from scouring and deposition of sediment, seasonal vegetation growth within or along river reaches, and ice formation can shift the rating curves overtime. Streamflow conditions that cause the energy slope to change can lead to looped rating curve conditions where the rating curve library conservatively over predicts water surface elevations on the rising limb of a hydrograph. Factors that lead to looped rating curve conditions include rapidly changing flows and backwater effects. A hybrid approach that incorporates unsteady simulations with hydraulic models for periods with rapidly changing flow conditions is an area that can be investigated as an approach to augment the rating curve library approach we have demonstrated. For backwater effects associated with the prescription of downstream water level boundary conditions, models can be extended downstream to ensure that backwater effects do not affect water surface elevations at the locations of interest upstream. For scenarios where this is not possible, Yen and Castro (2000) proposed the hydraulic performance graph approach, where the rating curve library is extended to include all possible backwater profiles.

It is apparent from our results that the numerical weather predictions, runoff from land surface models, and the downscaling process employed in converting the coarse scaled runoff into local streamflow predictions have the largest impacts on the ability to accurately forecast flood inundation. Improvements in these areas can be translated into better inundation forecasts. It is also important to note that the treatment of uncertainty presented in this paper only accounts for uncertainty related to the numerical weather predictions and the runoff from the ECMWF-HTESSEL simulations. Although climate inputs are understood to be the largest source of uncertainty for streamflow forecasting and, therefore, inundation predictions, uncertainty remains in the prescription of the initial conditions and parameters associated with the RAPID model; the downscaling of the ECMWF-RAPID runoff, the initial/boundary conditions and parameters of the hydraulic models used to derive the rating curve library, and the interpolation methods used in transforming streamflows to inundation depths using the rating curve library remains. For the uncertainties associated with the hydraulic modeling, the rating curve library can be extended including results from Monte Carlo simulations over a range of hydraulic model parameter distributions.

2.6 Supplemental Material

RCAFF can be downloaded at https://github.com/calebbuahin/rcaff.

Acknowledgments

This work was conducted as part of the National Flood Interoperability Experiment (NFIE) summer institute held at the National Water Center on the University of Alabama Campus, in Tuscaloosa from June 1st to July 17th, 2015. The goal of the experiment was to investigate approaches for conducting real-time, high spatial resolution, hydrologic modeling over the entire nation using open and interoperable frameworks for improved emergency response and community resilience. We wish to express our gratitude to the organizers of the institute including the Consortium of Universities for the Advancement of Hydrologic Sciences, Inc. (CUAHSI), the National Oceanic and Atmospheric Administration-National Weather Service (NOAA-NWS) and their federal partners for their financial support as well as the cyberinfrastructure and datasets they provided during the summer institute. We also wish to express our gratitude to Alan Snow for making the results of his ECMWF-RAPID results available for study. Finally, we wish to thank the NFIE steering committee, the theme leaders, and the course coordinators for their invaluable support and insights during the summer institute.

References

Alfieri, L., P. Burek, E. Dutra, B. Krzeminski, D. Muraro, J. Thielen, and F. Pappenberger, 2013. GloFAS - Global Ensemble Streamflow Forecasting and Flood Early Warning. Hydrology and Earth System Sciences 17:1161–1175. DOI:10.5194/hess-17-1161-2013

Alfieri, L., P. Salamon, A. Bianchi, J. Neal, P. Bates, and L. Feyen, 2014. Advances in Pan-European Flood Hazard Mapping. Hydrological Processes 28:4067–4077. DOI:10.1002/hyp.9947

Bailey, J.F. and H.A. Ray, 1967. Definition of Stage-Discharge Relation in Natural Channels by Step-Backwater Analysis. USGS Numbered Series, U.S. G.P.O.,. http://pubs.er.usgs.gov/publication/wsp1869A. Accessed 10 Jan 2016.

Balsamo, G., A. Beljaars, K. Scipal, P. Viterbo, B. van den Hurk, M. Hirschi, and A.K. Betts, 2009. A Revised Hydrology for the ECMWF Model: Verification from Field Site to Terrestrial Water Storage and Impact in the Integrated Forecast System. Journal of Hydrometeorology 10:623–643. DOI:10.1175/2008JHM1068.1

Bates, P.D. and A.P.J. De Roo, 2000. A Simple Raster-Based Model for Flood Inundation Simulation. Journal of Hydrology 236:54–77. DOI:10.1016/S0022-1694(00)00278-X

Braca, G., 2008. Stage-Discharge Relationships in Open Channels: Practices and Problems. Univ. degli Studi di Trento, Dipartimento di Ingegneria Civile e Ambientale. http://www.ing.unitn.it/dica/tools/download/Quaderni/Foralps_TR_11.pdf. Accessed Jan 2016.

Cloke, H.L. and F. Pappenberger, 2009. Ensemble Flood Forecasting: A Review. Journal of Hydrology 375:613–626. DOI:10.1016/j.jhydrol.2009.06.005

Lower Colorado River Authority, 2014. Colorado River Flood Guide. http://www.lcra.org/water/floods/Documents/2014-Colorado-River-Flood-Guide-11-14.pdf. Accessed Jan 2016.

Corbett, D.M., C.G. Paulsen, E.D. Burchard, M.H. Carson, J.J. Dirzulaitis, and H.E. Grosbach, 1943. Stream-Gaging Procedure, a Manual Describing Methods and Practices of the Geological Survey. USGS Numbered Series, U. S. Govt. Print. Off.,. http://pubs.er.usgs.gov/publication/wsp888. Accessed 10 Jan 2016.

David, C.H., D.R. Maidment, G.-Y. Niu, Z.-L. Yang, F. Habets, and V. Eijkhout, 2011. River Network Routing on the NHDPlus Dataset. Journal of Hydrometeorology

12:913–934. DOI:10.1175/2011JHM1345.1

David, C.H., Z.-L. Yang, and J.S. Famiglietti, 2013. Quantification of the Upstream-to-Downstream Influence in the Muskingum Method and Implications for Speedup in Parallel Computations of River Flow: Upstream-to-Downstream Influence in the Muskingum Method. Water Resources Research 49:2783–2800. DOI:10.1002/wrcr.20250

Dawdy, D.R., 1961. Depth-Discharge Relations of Alluvial Streams -- Discontinuous Rating Curves. USGS Numbered Series, U.S. G.P.O.,. http://pubs.er.usgs.gov/publication/wsp1498C. Accessed 10 Jan 2016.

Demeritt, D., H. Cloke, F. Pappenberger, J. Thielen, J. Bartholmes, and M.-H. Ramos, 2007. Ensemble Predictions and Perceptions of Risk, Uncertainty, and Error in Flood Forecasting. Environmental Hazards 7:115–127. DOI: 10.1016/j.envhaz.2007.05.001

Goodell, M.C.R., 2014. Breaking the HEC-RAS Code: A User’s Guide to Automating HEC-RAS. h2ls. ISBN-978-0-9908918-0-2

Hodges, B.R., 2013. Challenges in Continental River Dynamics. Environmental Modelling & Software 50:16–20. DOI:10.1016/j.envsoft.2013.08.010

Jung, Y. and V. Merwade, 2012. Uncertainty Quantification in Flood Inundation Mapping Using Generalized Likelihood Uncertainty Estimate and Sensitivity Analysis. Journal of Hydrologic Engineering 17:507–520. DOI:10.1061/(ASCE)HE.1943-5584.0000476

Lehner, B., K. Verdin, and A. Jarvis, 2006. HydroSHEDS. http://hydrosheds.cr.usgs.gov/index.php. Accessed 13 Dec 2015.

Lewis, J., 1998. Enhancements of River Forecasts Using Dynamic Hydraulic Flow Routing. Phoenix, Arizona. http://www.nws.noaa.gov/oh/hrl/papers/ams/ams98-5.htm. Accessed 8 Jan 2016.

Liu, F. and B.R. Hodges, 2014. Applying Microprocessor Analysis Methods to River Network Modelling. Environmental Modelling & Software 52:234–252. DOI:10.1016/j.envsoft.2013.09.013

Lott, N. and T. Ross, 2006. Tracking and Evaluating US Billion Dollar Weather Disasters, 1980-2005. 1.2. American Meteorological Society, Atlanta, GA. ftp://205.167.25.101/pub/data/papers/200686ams1.2nlfree.pdf. Accessed 12 Nov 2015.

McCarthy, G.T., 1939. The Unit Hydrograph and Flood Routing. Army Engineer District, Providence.

McKay, L., T. Bondelid, T. Dewald, C. Johnston, R. Moore, and A. Rea, 2012. NHDPlus Version 2: User Guide. ftp://ec2-54-227-241-43.compute-1.amazonaws.com/NHDplus/NHDPlusV21/Documentation/NHDPlusV2_User_Guide.pdf.

Merwade, V., F. Olivera, M. Arabi, and S. Edleman, 2008. Uncertainty in Flood Inundation Mapping: Current Issues and Future Directions. Journal of Hydrologic Engineering 13:608–620. DOI:10.1061/(ASCE)1084-0699(2008)13:7(608)

Mitchell, W.D., 1954. Stage-Fall-Discharge Relations for Steady Flow in Prismatic Channels. USGS Numbered Series, U.S. Govt. Print. Office,. http://pubs.er.usgs.gov/publication/wsp1164. Accessed 10 Jan 2016.

Moore, R., 2008. Towards a Theory of Digital Preservation. International Journal of Digital Curation 3:63–75. DOI:10.2218/ijdc.v3i1.42

Sauer, V.B., 2002. Standards for the Analysis and Processing of Surface-Water Data and Information Using Electronic Methods. U.S. Geological Survey, Reston, Virginia. http://pubs.er.usgs.gov/djvu/WRI/wrir_01_4044.pdf. Accessed Jan 2016.

Schmidt, A.R. and B.C. Yen, 2001. Stage-Discharge Relationship in Open Channels. Proceedings of the 3rd International Symposium on Environmental Hydraulics., Tempe, AZ. DOI:10.1.1.200.8029

Shewchuk, J.R., 1996. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. Applied Computational Geometry towards Geometric Engineering. Springer, pp. 203–222. DOI: 10.1007/BFb0014497

Smemoe, C.M., E.J. Nelson, A.K. Zundel, and A.W. Miller, 2007. Demonstrating Floodplain Uncertainty Using Flood Probability Maps. JAWRA Journal of the American Water Resources Association 43:359–371. DOI: 10.1111/j.1752-1688.2007.00028.x

Snow, A., 2015. A New Global Forecasting Model to Produce High-Resolution Stream Forecasts. Thesis, Brigham Young University, Provo, Utah. http://scholarsarchive.byu.edu/etd/5272. Accessed Jan 2010.

Thieken, A.H., H. Cammerer, C. Dobler, J. Lammel, and F. Schöberl, 2014. Estimating Changes in Flood Risks and Benefits of Non-Structural Adaptation Strategies - a Case Study from Tyrol, Austria. Mitigation and Adaptation Strategies for Global Change:1–34. DOI:10.1007/s11027-014-9602-3

Thielen, J., K. Bogner, F. Pappenberger, M. Kalas, M. del Medico, and A. de Roo, 2009. Monthly-, Medium-, and Short-Range Flood Warning: Testing the Limits of Predictability. Meteorological Applications 16:77–90. DOI:10.1002/met.140

U.S. Army Corps of Engineers, 2015. HEC-RAS 5.0 River Analysis System: User's Manual, Version 5.0.

Warmerdam, F., 2008. The Geospatial Data Abstraction Library. G. B. Hall and M. Leahy (Editors). Open Source Approaches in Spatial Data Handling. Springer Berlin Heidelberg, Berlin, Germany, pp. 87–104.

Yen, B. and J.G. Castro, 2000. Open-Channel Capacity Determination Using Hydraulic Performance Graph. Journal of Hydraulic Engineering 126:112–122. DOI: 10.1061/(ASCE)0733-9429(2000)126:2(112)

Figures

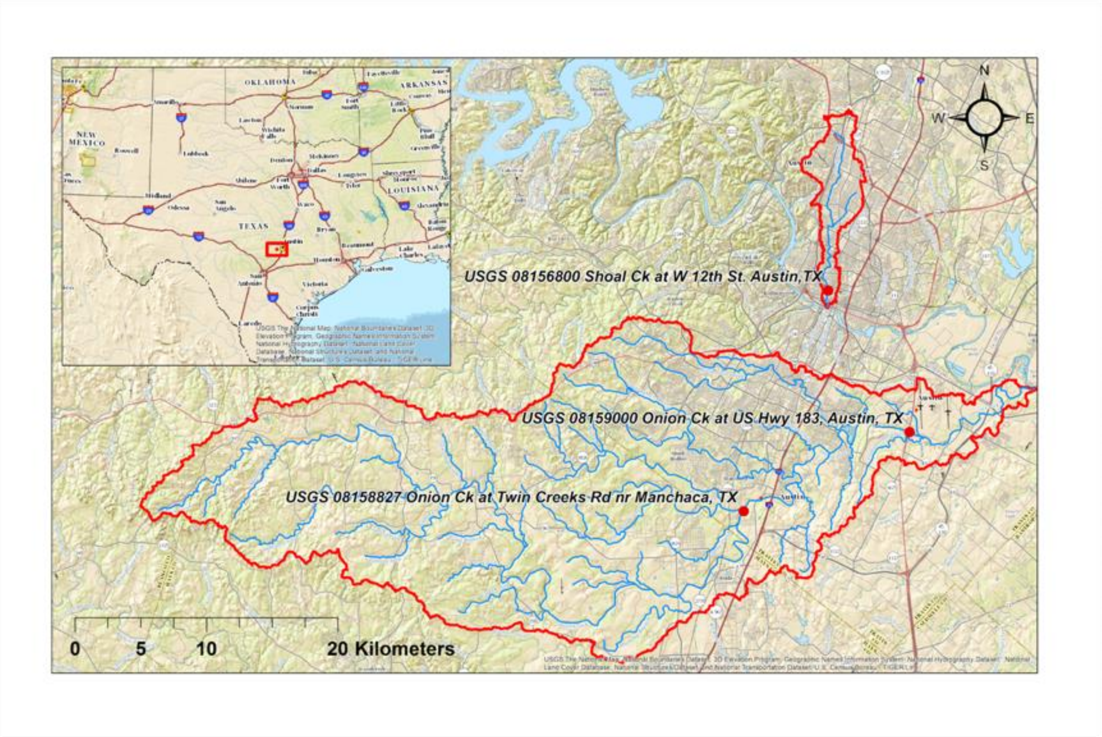

OKLAHOMA
ARKANSAS
NEW MEXICO
TEXAS
LOUISIANA
N
W
E
S
USGS 08156800 Shoal Ck at W 12th St. Austin,TX
USGS 08159000 Onion Ck at US Hwy 183, Austin, TX
USGS 08158827 Onion Ck at Twin Creeks Rd nr Manchaca, TX
0
5
10
20 Kilometers

Figure 2.1 Map of Onion Creek and Shoal Creek watersheds as well as USGS gages used to compare with model results.

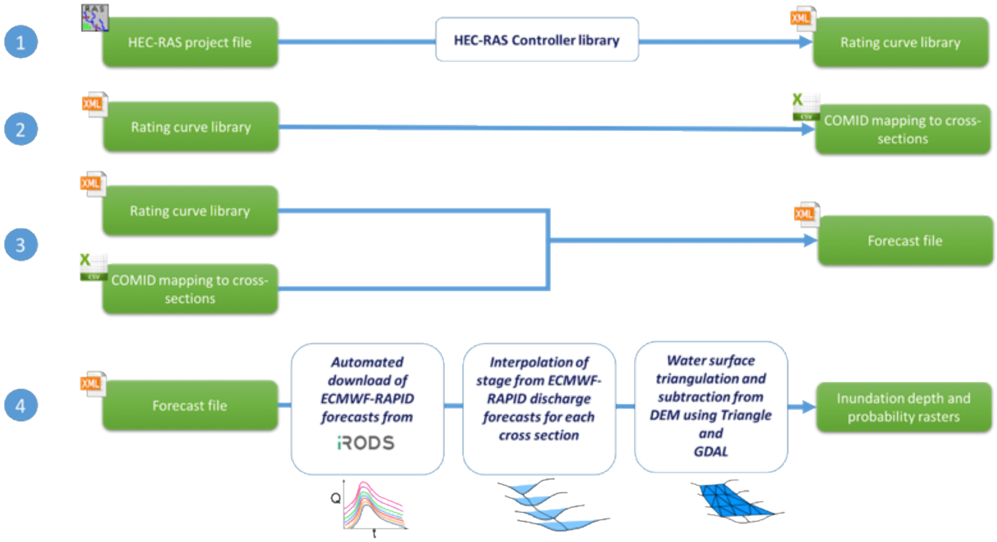

1
HEC-RAS project file
HEC-RAS Controller library
Rating curve library
2
Rating curve library
COMID mapping to cross-sections
3
Rating curve library
COMID mapping to cross-sections
Forecast file
4
Forecast file
Automated download of ECMWF-RAPID forecasts from iRODS
Interpolation of stage from ECMWF-RAPID discharge forecasts for each cross section
Water surface triangulation and subtraction from DEM using Triangle and GDAL
Inundation depth and probability rasters
Q
t

Figure 2.2 RCAFF workflow

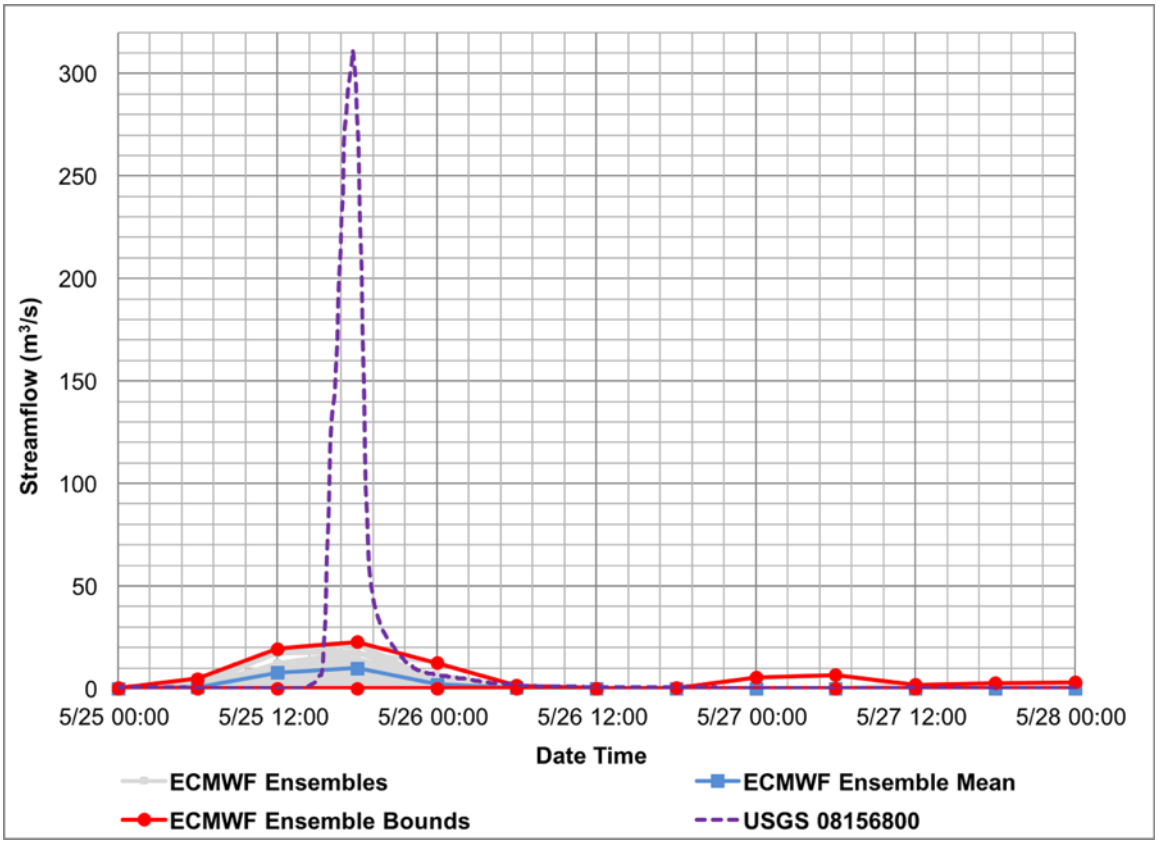


Figure 2.3 Time series plot of ECMWF-RAPID ensemble streamflows versus USGS 08156800 Shoal Creek at W 12th St, Austin, TX.

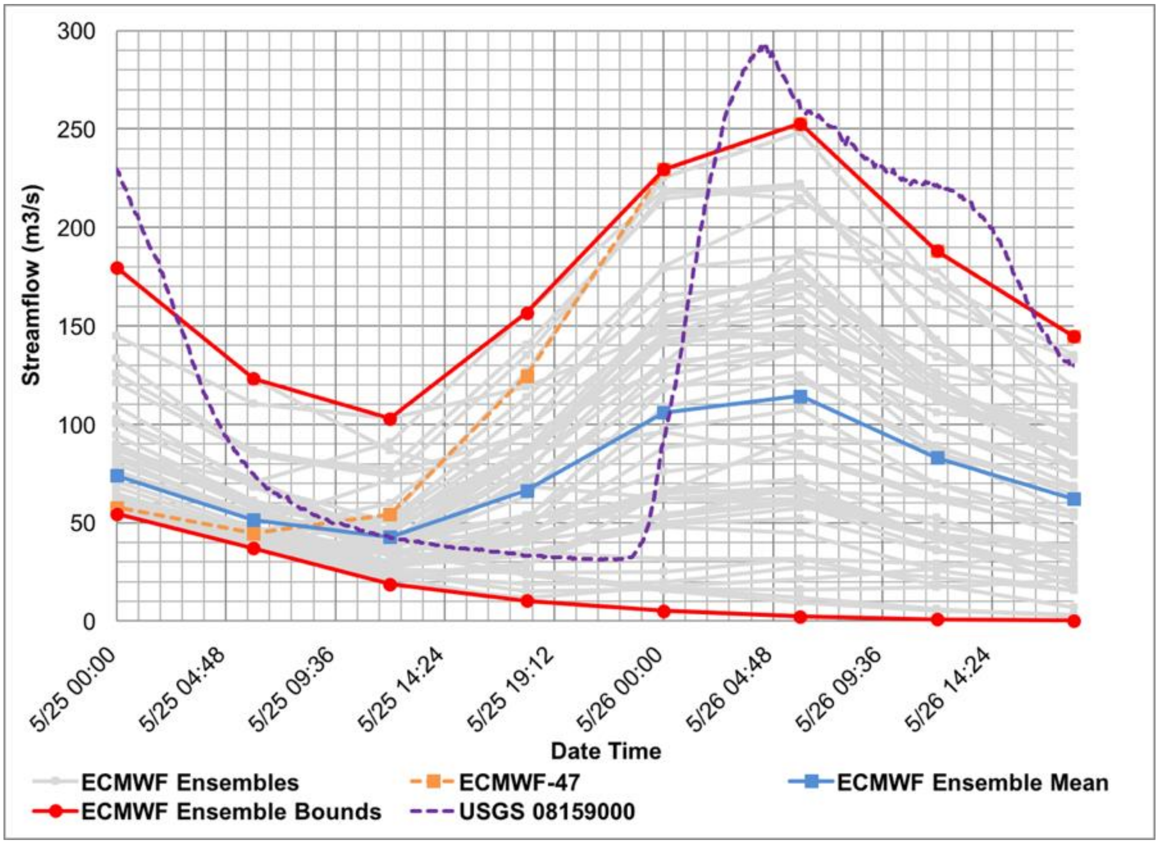


Figure 2.4 Time series plot of ECMWF-RAPID ensemble streamflows versus USGS 08159000 Onion Creek at US Hwy 183, Austin, TX.

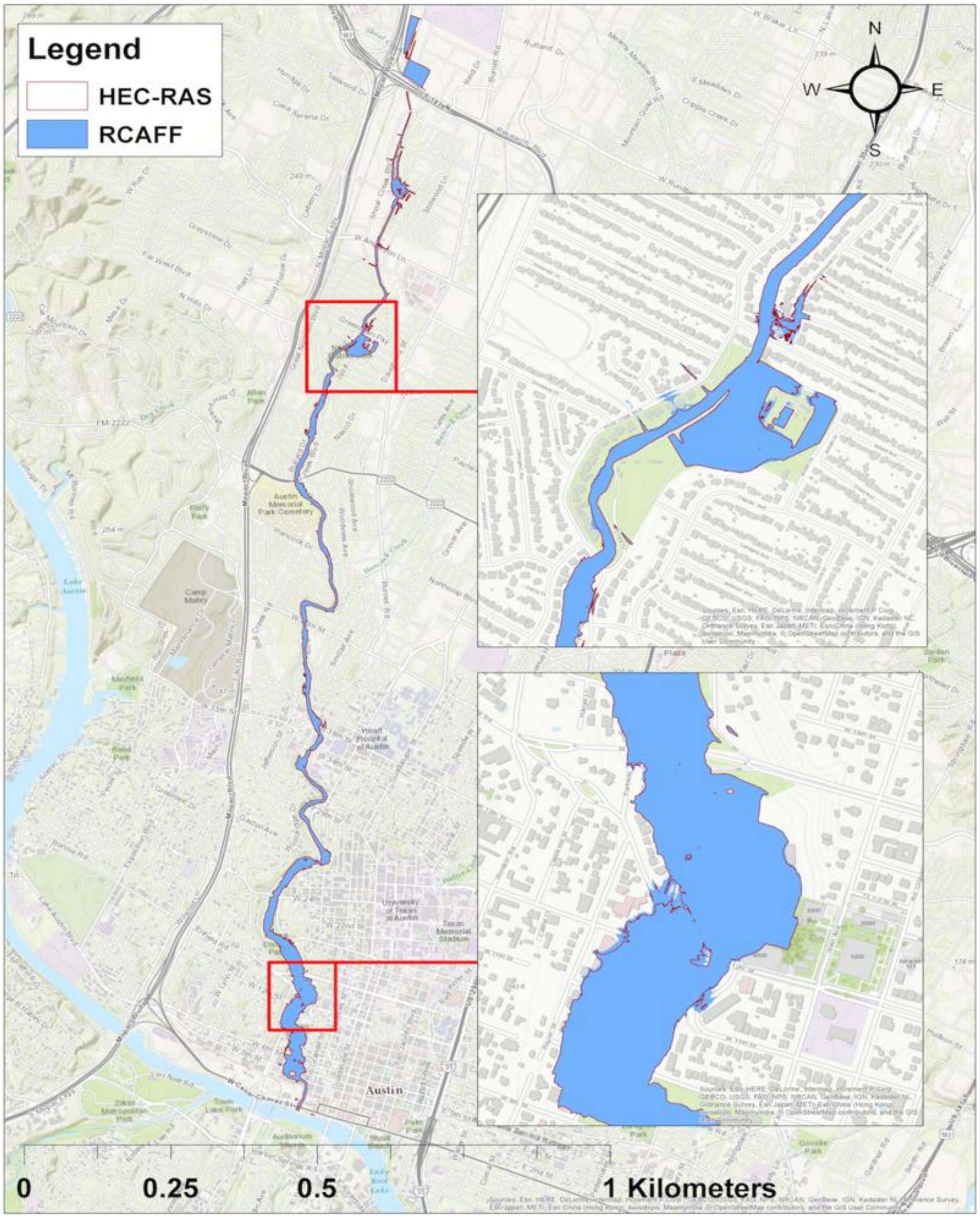


Figure 2.5 Comparison between flood extents generated using (a) Rating Curve approach and (b) HEC-RAS model for Shoal Creek at 05/25/2015 with peak streamflow of 306 m^3/s.

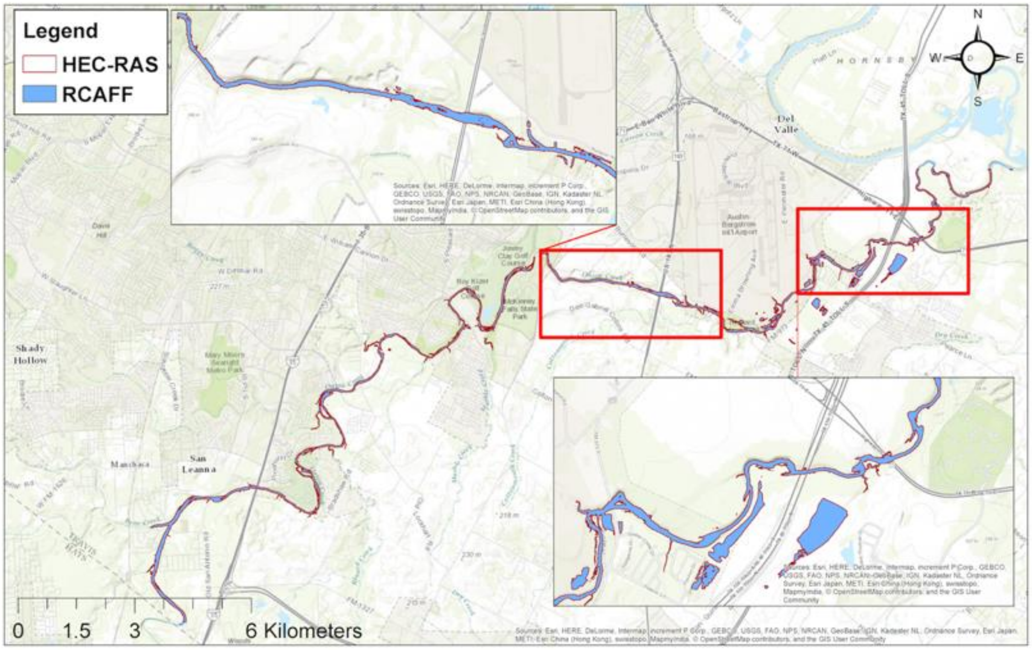

Legend
HEC-RAS
RCAFF
N
W
E
S
Del Valle
Shady Hollow
Manchaca
San Leanna
Austin Bergstrom Intl Airport
McKinney Falls State Park
Onion Creek
0 1.5 3 6 Kilometers
Sources: Esri, HERE, DeLorme, Intermap, increment P Corp., GEBCO, USGS, FAO, NPS, NRCAN, GeoBase, IGN, Kadaster NL, Ordnance Survey, Esri Japan, METI, Esri China (Hong Kong), swisstopo, MapmyIndia, © OpenStreetMap contributors, and the GIS User Community

Figure 2.6 Comparison between flood extents generated using (a) Rating Curve approach and (b) HEC-RAS model for Onion Creek, Austin, Texas at 05/25/2015 with peak streamflow of 292 m^3/s.

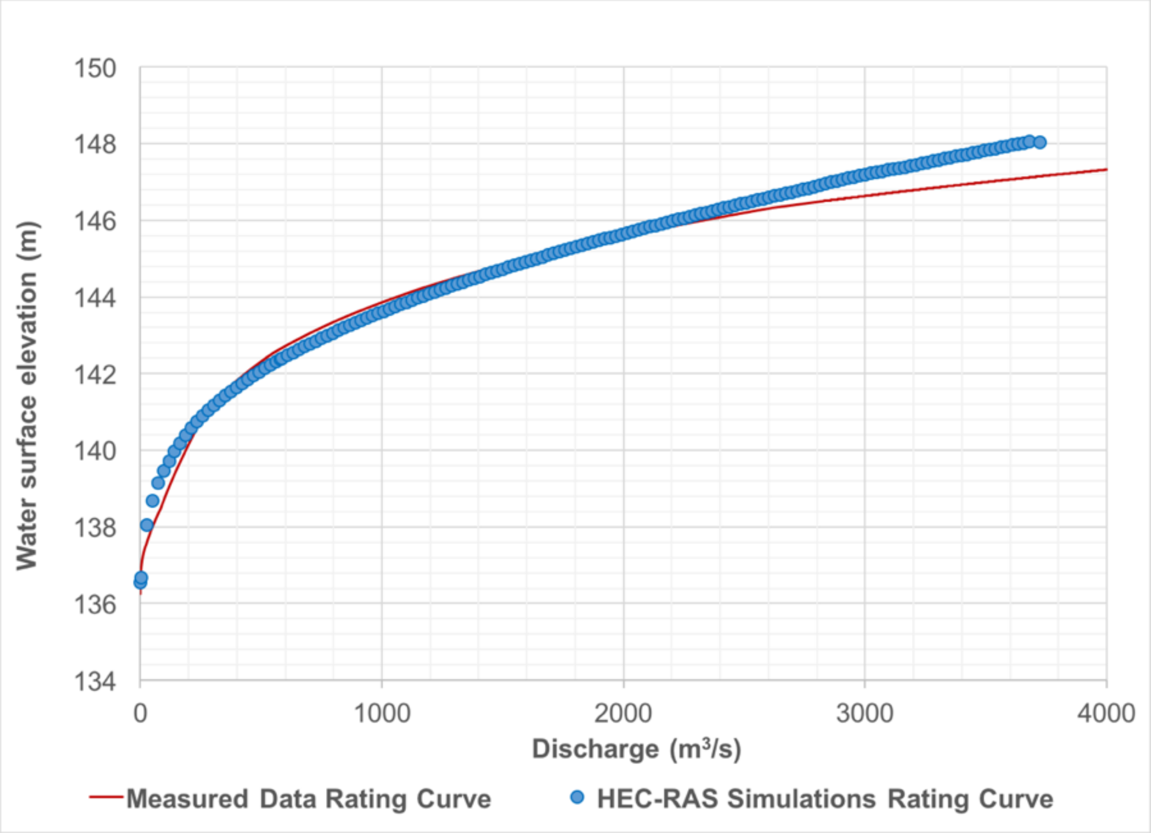


Figure 2.7 Comparison of rating curve derived from (a) hydraulic simulations (blue) and (b) measured data (red) at USGS gage 08159000 Onion Creek at US Hwy 183.

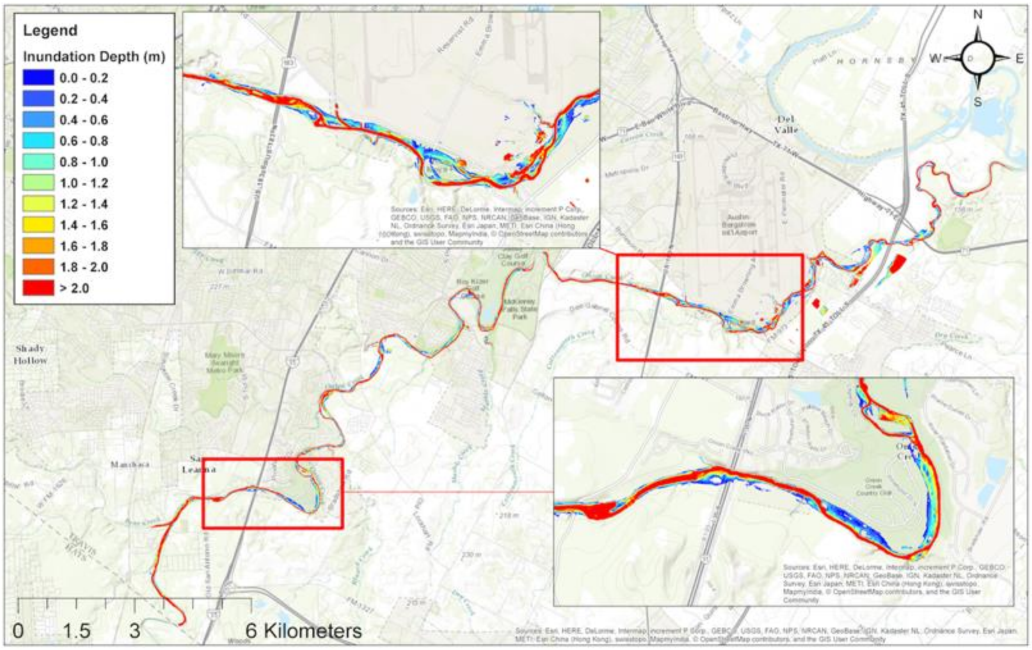


Figure 2.8 RCAFF inundation depth map for Ensemble 47 for Onion Creek at 5/26/2015 6:00 A.M.

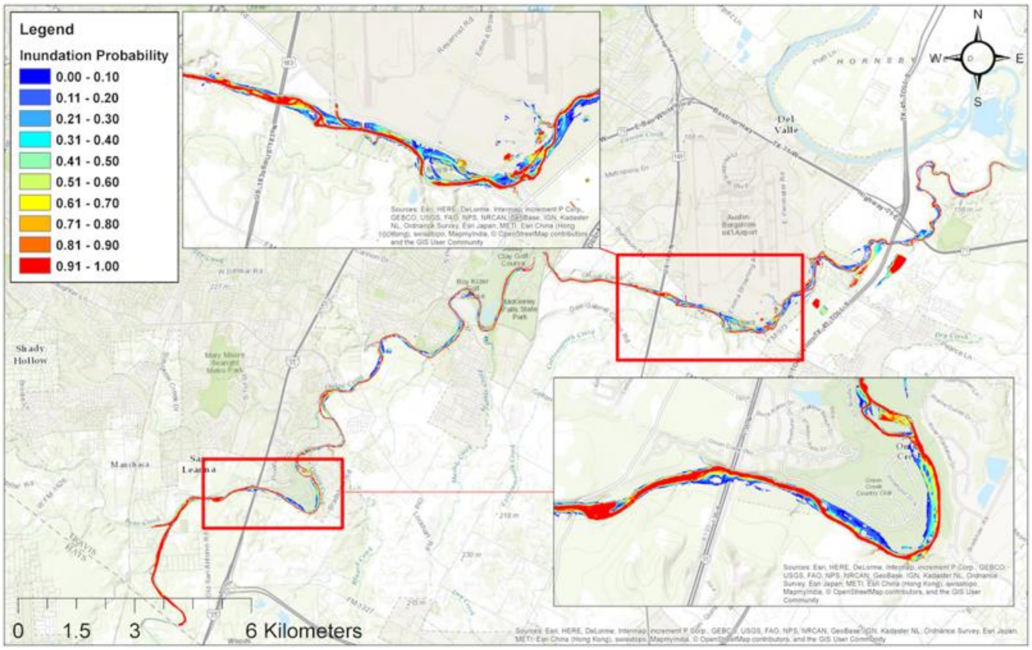


Figure 2.9 RCAFF inundation probability map for Onion Creek at 5/26/2015 6:00 A.M.

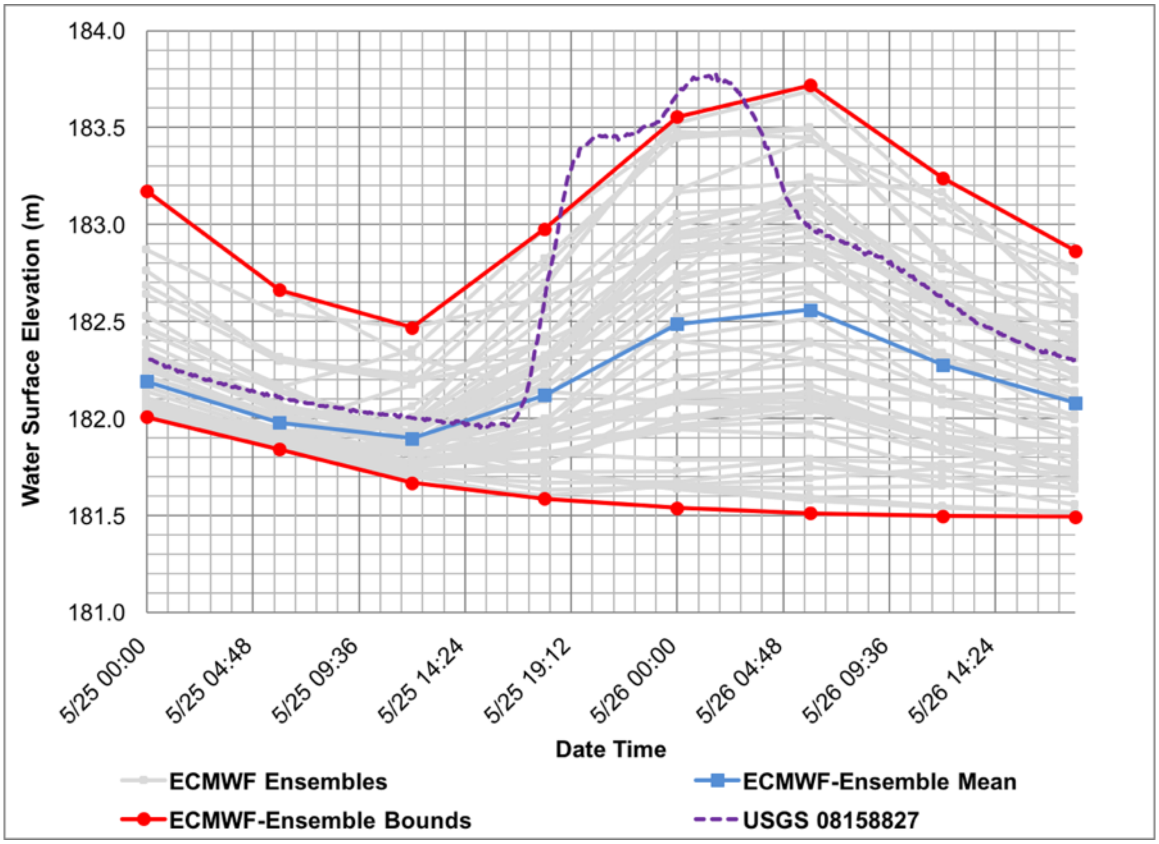


Figure 2.10 RCAFF Water Surface Elevation vs. USGS 08158827 at Onion Creek at Twin Creeks Road Near Manchaca, TX.

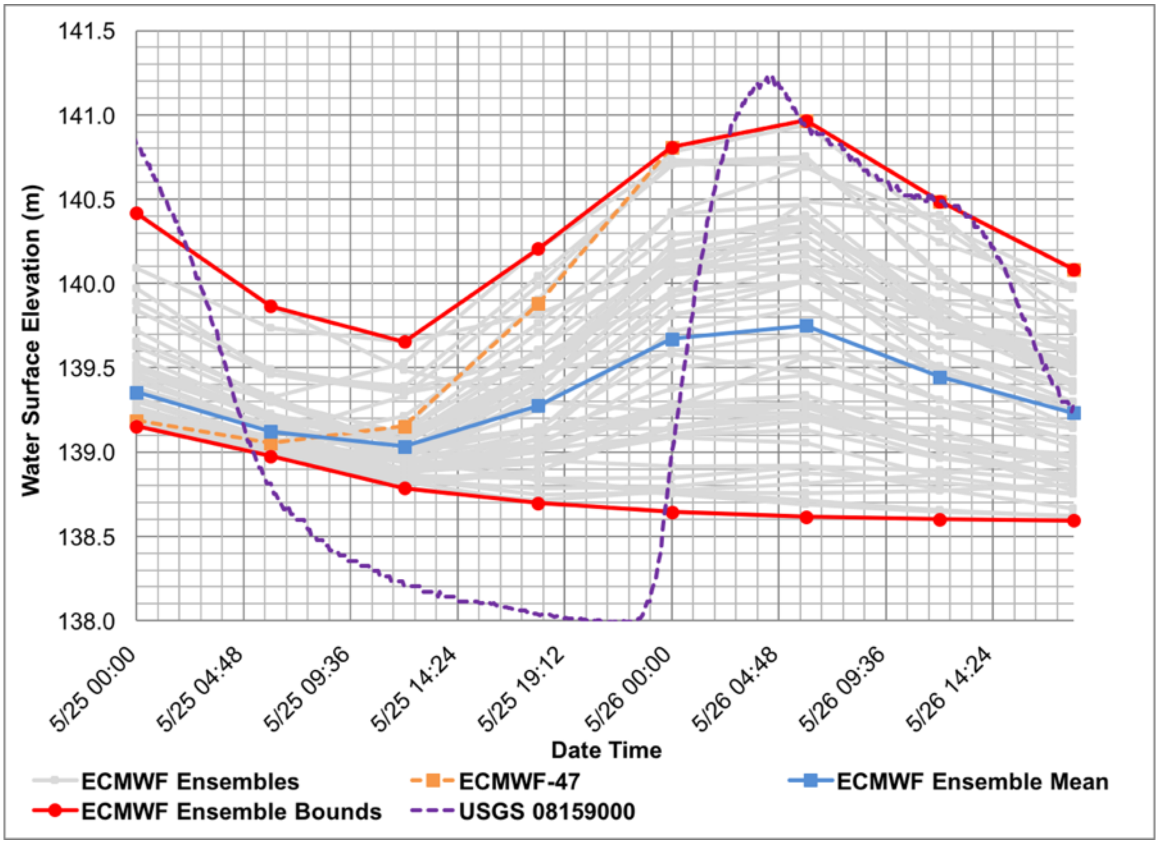


Figure 2.11 RCAFF Water Surface Elevation vs. USGS Gage 08159000 at Onion Creek at US Hwy 183, Austin Texas.

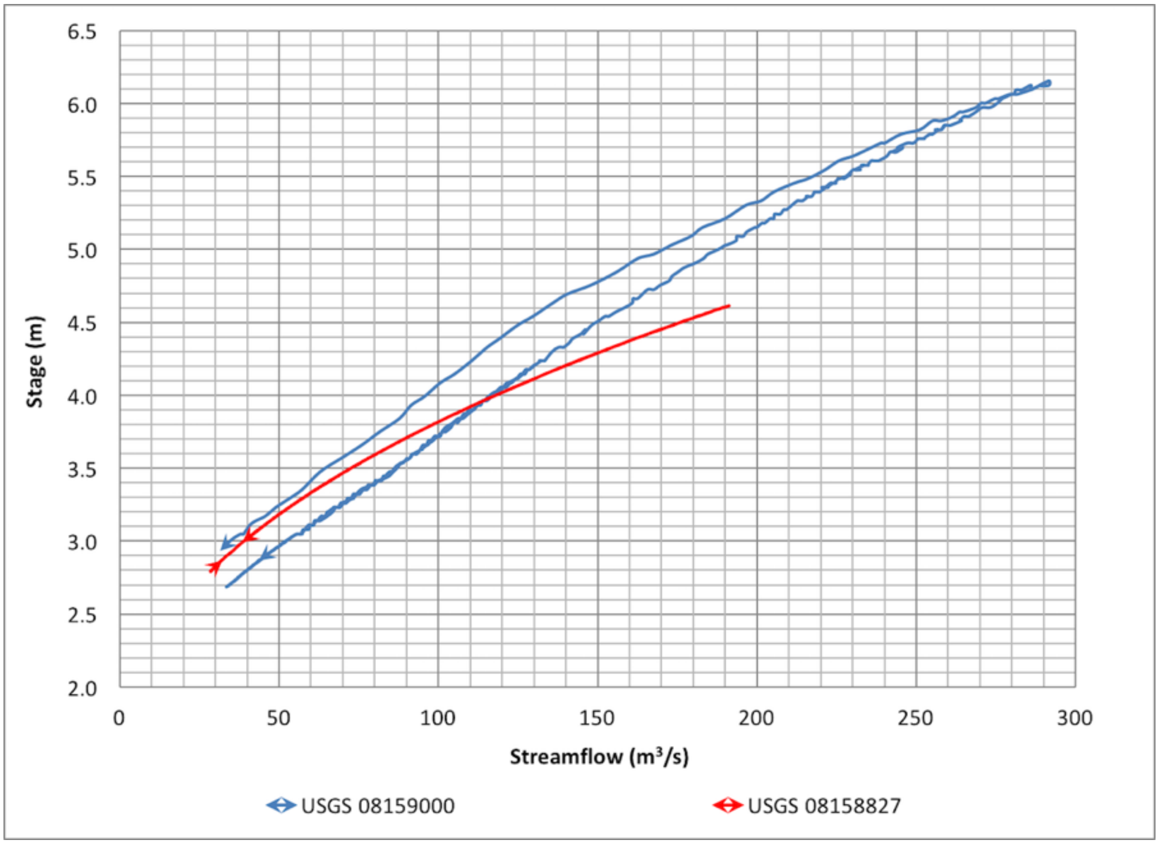


Figure 2.12 Observed stage vs. streamflow for USGS 08158827 and USGS 08159000 to check for hysteresis.

CHAPTER 3

EVALUATING THE SIMULATION TIMES AND MASS BALANCE ERRORS OF COMPONENT-BASED MODELS: AN APPLICATION OF OPENMI 2.0 TO AN URBAN STORMWATER SYSTEM[1]

Abstract

In making the decision whether to use component-based modeling, its benefits must be balanced against computational costs. Studies evaluating these costs using the Open Modeling Interface (OpenMI) have largely used models with simplified formulations, small spatial and temporal domains, or a limited number of components. We evaluate these costs by applying OpenMI to a relatively complex Stormwater Management Model (SWMM) for the City of Logan, Utah, USA. Configurations of coupled OpenMI components resulting from decomposing the stormwater model by process (i.e., runoff coupled to routing) and then by space (i.e., groups of catchments coupled together) were compared to a reference model executed in the standard SWMM configuration. Simulation times increased linearly with the number of connections between components, and mass balance error was a function of the degree to which a component resolved time series data received. This study also examines and proposes some strategies to address these computational costs.

[1] Buahin, C.A. and J.S. Horsburgh, 2015. Evaluating the Simulation Times and Mass Balance Errors of Component-Based Models: An Application of OpenMI 2.0 to an Urban Stormwater System. Environmental Modelling & Software 72:92–109.

Reproduced with permissions from Environmental Modeling and Software

3.1 Introduction

Traditional model development in the earth systems modeling field has largely been characterized by monolithic codes with highly interdependent functions compiled into a single executing unit. Models developed using this approach are referred to as tightly coupled models (Sui and Maggio, 1999). This approach to model development provides a developer complete control over a modeling system and has the advantage of allowing the optimization of computational performance across an entire modeling system (Castronova and Goodall, 2013). However, tightly coupled models are often overly specialized to the specific applications for which they were developed. This can make them inflexible for applications to different use cases. Additionally, adapting to technological advancements and isolating and remediating errors can be challenging for tightly coupled models (Szyperski, 2002).

Component-based modeling, or loose model coupling, is an alternative model development paradigm that has been proposed to overcome the challenges identified with tightly coupled models. It involves decomposing a complex system into smaller functional units called “components” that have specified interfaces, which allows them to be coupled together to represent a larger and more complex system. A definition that captures the critical properties a component must possess was provided by Szyperski and Pfister (1997). They defined a component as a unit of composition with contractually specified interfaces and explicit context dependencies that can be deployed independently and is subject to composition by third parties. This general trend toward the use of components for software development is part of a movement within software engineering toward the assembly of complex systems by building from readily available, reusable parts (Gross, 2005). This approach to development promises greater reusability, extensibility, and maintainability as well as faster development times and more robust applications with lower development and long-term maintenance costs (Löwy, 2003). For earth systems modelers, component-based modeling provides a way to represent the complex dynamics that describe earth processes more holistically by allowing them to experiment with different model formulations and ultimately selecting those that are most appropriate for their specific study. It also facilitates interdisciplinary studies and integrated natural resources management efforts by allowing researchers to better explore relationships between domains of different sub-disciplines that are typically studied and modeled independently (Argent et al., 1999; Moore and Tindall, 2005; Peckham et al., 2013).

A key requirement for the successful adoption and use of component-based models is the definition of clear and well-defined standards, specifications, and frameworks that describe how components are to be developed to enable interoperability as well as the proper context for their use. However, the addition of a component-based modeling framework that implements these standards on top of the native computational codes of existing models is likely to introduce performance costs into a modeling system. These costs may include increased total simulation times resulting from function calls and data transformations introduced by the underlying component-based framework. Additionally, errors may arise because of the discontinuities at the connection points between components over which data is exchanged. For component-based modeling frameworks to be adopted and used effectively, these computational performance cost considerations must be investigated and understood under realistic modeling scenarios.

An understanding of the computational costs of component-based modeling will be instructive to model developers trying to assess the tradeoffs between the two modeling development paradigms for a particular model development effort. Large computing resource demands (e.g., large total simulation times) may preclude the use of component-based modeling frameworks for complex modeling studies spanning large spatial and/or temporal domains – e.g., climate and land use change evaluations. These demands may also make it more difficult to conduct integrated natural resource management, impact mitigation, model calibration, or forecasting studies using stochastic, optimization, or data assimilation methods, which typically require many simulations of a particular model with varied inputs. Accurate model predictions are important for applications where there are elevated societal risks associated with the processes being simulated (e.g., simulation of flood risk), and an assessment of model error introduced through coupling is important in evaluating the appropriateness of model coupling for these purposes.

Examples of component-based modeling frameworks and standards in the earth systems field include: The Earth System Modeling Framework (ESMF) (Hill et al., 2004), Model Coupling Toolkit (MCT) (Warner et al., 2008), Community Surface Dynamics Modeling System (CSDMS) (Peckham et al., 2013), Object Modeling System (OMS) (David et al., 2002), and the Open Modeling Interface (OpenMI) standard (Moore and Tindall, 2005; Gregersen et al., 2005; Gregersen et al., 2007). Of these frameworks, OpenMI has recently been formally adopted as an Open Geospatial Consortium (OGC) standard (Vanecek and Moore, 2014) and is becoming more widely used.

In this study, we evaluated the performance of OpenMI as the number of coupled model components increases. We employed the United States Environmental Protection Agency's (USEPA) Stormwater Management Model (SWMM) to develop a reference SWMM model for the stormwater conveyance system for the City of Logan, Utah, USA. This reference model was run in its standard, tightly coupled SWMM configuration and compared to several coupled model configurations using OpenMI 2.0. To derive the coupled model configurations for comparison, the native SWMM computational code was first wrapped as an OpenMI 2.0 compliant component library. The reference stormwater model was then decomposed by process (i.e., runoff coupled to routing) and then by space (i.e., individual catchments or groups of catchments with their associated hydraulic routing elements coupled to each other). Each of the model components that were coupled was represented using the SWMM OpenMI component library that we developed. We chose SWMM because it is widely used in practice for urban stormwater studies and can simulate backwater flows, pressurized flows, flow reversals, and non-dendritic stream and pipe network layouts (Rossman, 2010). SWMM's underlying formulations and setup options are similar to many hydrologic and hydrodynamic models and is, therefore, a useful model for exploring the general challenges to be expected when loosely coupling models.

Comparison of the coupled modeling configurations and the reference model was accomplished by using total simulation time and total mass balance error (TMBE) as evaluation metrics. In general, for many hydrologic, hydraulic, and hydrodynamic models, using long time steps yield shorter total simulation times. However, longer time steps can produce larger TMBE. We therefore sought to create an experiment that could help assess the tradeoffs between TMBE and total simulation time by varying time steps for all the model configurations evaluated.

3.2 Background

3.2.1 The Open Modeling Interface 2.0 Standard

OpenMI was developed to provide a comprehensive modeling system to address the European Union's (EU) Water Framework Directive (WFD). The WFD aims to achieve good ecological status of surface waters through integrated river basin management by the year 2015 (Blind et al., 2005). The development of OpenMI resulted from a collaboration between researchers, practitioners, and some of Europe's well known commercial modeling software vendors with some sponsorship from the EU.

OpenMI was developed using an object-oriented approach with clear and logical inheritance relationships between classes. Interfaces underlie the OpenMI standard, as is the case with many component-based modeling frameworks. Interfaces define a logical grouping of method definitions that acts as a contract for inter-component communication and communication between components and their underlying frameworks; each component is free to provide details of its own interpretation of the method definitions provided by an interface (Fröhlich and Franz, 1999; Löwy, 2003). Frameworks provide the environment in which components interact with each other. To make migration to OpenMI 2.0 less tedious and give developers flexibility to optimize data exchange between components, the OpenMI developers proposed standard interface definitions that define direct runtime data exchange between components with no dependence on the framework in which they execute.

The interface defining a component in the OpenMI 2.0 specification is the *IBaseLinkableComponent*. This interface controls the underlying computational engine of a model component. It contains functions for initializing and validating the model component, applying input data, providing output data, performing time stepping, among others. The *IBaseLinkableComponent* interface also has two lists of *IBaseInput* and *IBaseOutput* exchange items that define the data that can be consumed and supplied by the component respectively. Exchange items define: (1) the variable being exchanged and units (e.g., flow (cms), flux (kg/m2/s) through the *IValueDefinition* and *IUnit* interfaces; (2) the geographic location at which the data are exchanged through the *ISpatialDefinition* and *IElementset* interfaces; (3) the period over which data are exchanged through the *ITime* and *ITimeset* interfaces; and (4) the values to be exchanged through the *IBaseValueSet* interface. An *IBaseInput* exchange item has a provider property, which is an *IBaseOutput* exchange item of another component that supplies the *IBaseInput's* data. Correspondingly, an *IBaseOutput* exchange item has a list of consumers, which are the *IBaseInput* exchange items requesting data from that *IBaseOutput*. These linkages effectively establish the coupling between models.

OpenMI 2.0 was developed using a pull-based, pipe-and-filter architecture (Buschmann, 1996), which consists of communicating components (source components and target components) that exchange memory-based data in a predefined way and format (Gregersen et al., 2007). To initiate data exchange and computation in the coupled component chain, the component at the end of the component chain serves as a trigger called by the framework for initiating a simulation. This trigger component requests the data it needs from source components linked to it and blocks any requests to itself until the requested data is returned. Components linked to this trigger component may propagate their own requests with blocking through the chained system and then compute the data requested from them upon receipt of responses to their requests (Figure 3.1). This process repeats itself autonomously until the trigger component finishes its simulation.

In the case of bi-directionally-linked components, where two linked components request data from each other, OpenMI prevents infinite recursion, as well as race conditions and deadlocking in multi-threaded implementations by blocking additional function calls to get values on a component's IBaseOutput exchange items if the component is already inside a function call to get values.

A type of an *IBaseOutput* exchange item called an *IBaseAdaptedOutput* can be used to mediate the data exchange between an *IBaseOutput* exchange item and an *IBaseInput* exchange item by transforming outputs into the type of data required by the *IBaseInput* exchange item. These transformations may include spatial and temporal data interpolation or extrapolation. *IBaseAdaptedOutput* exchange items can be chained together to provide proper order for data transformations as shown in Figure 3.2.

The data exchange over an OpenMI connection is inherently explicit in that each model component solves its own set of equations based on variable values that apply to the beginning of a time interval and returns variable values that apply to the end of the time interval or to the time interval as a whole (Fenske et al., 2011). This explicit time marching scheme is common to other loosely coupled frameworks and imposes the Courant-Fredrichs-Lewy (CFL) criteria (Courant et al., 1928) as a condition necessary (but not sufficient) for numerical stability and, therefore, model convergence. The purpose of the CFL criteria is to ensure that the distance traveled by a signal (e.g., a hydraulic wave) in one time step does not exceed the size of the spatial discretization (i.e., the size of an element that it travels through). Model time steps larger than the time step calculated because of the CFL criteria may lead to model instability and non-convergence, decreasing the degree to which the conservation laws are obeyed and giving rise to mass balance errors.

3.2.2 Component-Based Modeling using OpenMI

Applications of OpenMI have explored coupling models with spatially disparate discretizations and domains with feedbacks between model components (Elag et al., 2011; Yamagata et al. 2012), leveraging web service technologies to access model boundary data and to couple models over remote servers (Goodall et al., 2011; Castronova et al., 2013; Goodall et al., 2013; Bulatewicz et al., 2014), and for inter-disciplinary model coupling studies (Goodall et al., 2013; Knapen et al., 2013). Studies investigating the computational performance of OpenMI implementations have largely focused on hypothetical scenarios with simplified model formulations, often using small spatial and temporal domains and a limited number of components as examples to demonstrate methods and approaches. For example, in their study of the feedbacks between two temporally misaligned OpenMI components comprised of a surface water model component overlaying a sediment media model component, Elag et al. (2011) assumed a laterally averaged transport model over a simplified rectangular mesh. The three-dimensional advection-diffusion equation was simplified to only advection in the x-direction and diffusion in the z-direction for the water model and only diffusive transport in the z-direction for the sediment media model. Yamagata et al. (2012) provide another such example of a hypothetical study. They coupled the MIKE-SHE (Abbott et al., 1986; Refsgaard et al., 2010) distributed hydrologic model and the finite element subsurface flow and transport model FEFLOW (Diersch, 2014). Their study considered a simplified spatial domain and attempted to reproduce analytical results from the test problem proposed by Hunt (1999). This test is characterized by drawdown due to constant pumping in a simplified homogeneous aquifer bounded by a constant head on one side and by a stream boundary on the other. The few examples of OpenMI being used for realistic studies over large spatial and temporal domains provide some data on OpenMI's performance as the number of components increases. Performance has been expressed in terms of total simulation time or model accuracy metrics, but rarely both (Shrestha et al., 2013; Goodall et al., 2013).

Castronova and Goodall (2013) provide an example application detailing how OpenMI performs as the complexity of coupled components increases for a model application over a large spatial and temporal domain using standardized performance tests. In their study, they compared the widely used Hydrologic Engineering Center's Hydrologic Modeling System (HEC-HMS) rainfall runoff model with an equivalent version comprised of four coupled OpenMI components. These four components included a precipitation data component, an infiltration component, a surface runoff component, and a channel flow routing component. Model outputs from the HEC-HMS and the OpenMI model configurations were identical because Castronova and Goodall's process formulations (e.g., Muskingum routing) were the same as the HEC-HMS formulations and the unidirectional coupling employed represented the same coupling HEC-HMS uses for computational elements internally. The models' computational performance was evaluated using an endurance metric, in which the simulation time steps were decreased, and a load test metric where the number of model computational elements was increased. Their study showed no significant differences in computational performance between the two model development paradigms. However, the unidirectional data exchange (i.e., no feedbacks between components) used makes it difficult to translate their results to more complex hydraulic/hydrodynamic models that solve the mass and momentum conservation differential equations and require feedbacks between components. Additionally, the HEC-HMS model and its equivalent coupled OpenMI components used process formulations written using different programming languages. Therefore, a direct comparison of these two models could not be used to isolate the computational costs introduced by using the OpenMI implementation because the model code was not the same.

Talsma et al. (2012) conducted an OpenMI study that is similar in some respects to the study we present in here. In their study, up to six OpenMI components of the SOBEK hydraulic model (Stelling and Duinmeijer, 2003), which is comparable to

SWMM, were coupled, resulting in a maximum of 13 connection points. The six OpenMI components of the SOBEK model represented models from six adjacent water jurisdictional areas in Netherlands. The external and bi-directionally coupled OpenMI SOBEK components were compared to their equivalent, tightly coupled SOBEK model using total simulation time as the evaluation metric. Their results showed that, compared to the implicit time marching, tightly coupled SOBEK model counterpart, the computational effort of the external bidirectional coupling using OpenMI increased disproportionally with the number of model components (Talsma et al., 2012).

We extend the Talsma et al. (2012) study by significantly increasing the number of components involved in the coupling and assessing not only total simulation times but also total mass balance errors. Although not explicitly stated in their study, a cursory study of their model segmentation as well as the long time step used in their simulations (10 minutes) seem to indicate that their model was comprised of a small number of long computational elements (i.e., hydraulic routing elements including conduits, pipes, etc.). Our study involved a more complex network of conduits with varying sizes and other hydraulic structures (e.g., storage units and weirs). It is also important to note that their study, the other studies summarized here, and most OpenMI applications found in literature have used the older OpenMI Version 1.4. In our study, the latest OpenMI 2.0 specification was used.

3.3 Methods

3.3.1 SWMM OpenMI Component Development

The SWMM OpenMI component used in our study was developed using the OpenMI 2.0 C# Software Development Kit (SDK) supplied by the OpenMI Association (2014). This SDK was, however, incomplete and had to be completed for our study. We developed the SWMM OpenMI component to handle the runoff and routing processes of the SWMM model's native computational code. However, the code was designed so that its templates could be extended to include the other processes available in SWMM (e.g., water quality, snowmelt, groundwater, and evaporation processes). The complexity of the runoff computations is comparable to that of snowmelt, infiltration, and evaporation, which typically use algebraic equations. The routing portion's level of computational complexity is comparable to that of the transport and fate of constituents in water, which typically use differential equations that require initial and boundary conditions as well as approximate numerical solutions. These two processes are good representations for the two general levels of computational complexity in the SWMM model and indeed many other hydrologic and hydrodynamic models.

In SWWM, sub-catchments are modeled as non-linear reservoirs, with storage represented by maximum depression storage. This storage includes ponding, surface wetting, and interception. The difference between inflows (rainfall and snowmelt) and outflows (infiltration and evapotranspiration) that exceeds a sub-catchment's maximum depression storage is equal to runoff and is converted to a volumetric flow rate using Manning's equation (Rossman, 2010). Flow routing in SWMM is accomplished by solving the Saint Venant's equations (Equations 1 and 2) over the network formed from conduits (i.e., pipes, canals, and rivers) and sub-catchments connected at their endpoints by nodes (i.e., junctions, outfalls, storage units, and flow dividers) (Rossman, 2006):

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0 \tag{1}$$

$$\frac{\partial Q}{\partial t} + \frac{\partial\left(\frac{Q^2}{A}\right)}{\partial x} + gA\frac{\partial H}{\partial x} + gAS_f + gAh_L = 0 \tag{2}$$

where x is distance along a conduit, t is time, A is cross-sectional area of the conduit, Q is flow rate in the conduit, H is the hydraulic head of water in the conduit (elevation head plus any possible pressure head), S_f is the friction slope (head loss per unit length), h_L is the local energy loss per unit length of the conduit, and g is the acceleration due to gravity. SWMM uses an upwind, finite difference, explicit time marching scheme to solve these equations. At each time step, the equations are solved iteratively until the difference in the current iteration's estimated heads and the previous iteration's heads at all nodes connecting conduits are less than a certain specified convergence criteria value.

The SWMM computational engine allows a modeler to set a constant time step or to override the specified constant time step using an adaptive time step. This adaptive time step is the smaller of a user specified maximum time step or an under-relaxed time step calculated for each conduit using Equation 3 to comply with the CFL stability criterion:

$$\Delta t = R \times \frac{L}{V + \sqrt{gy}} \tag{3}$$

where R is the time step relaxation factor, L is the length of the conduit, and V and y are velocity and depth in a conduit, respectively.

TMBE is estimated in SWMM as the sum of the initial storage volume (V_i) and the total volume of model inflows (V_{in}) minus the sum of the final storage volume (V_f) and total volume of model outflows (V_{out}) as illustrated in Equation 4.

$$TMBE = (V_i + V_{in}) - \left(V_f + V_{out}\right) \tag{4}$$

To develop an OpenMI component from an existing model, Gregersen et al. (2007) recommended converting the existing model's native code into a library that can then be called externally from an OpenMI compliant component wrapper library. This approach minimizes the number of OpenMI specification implementations in the existing model's native code and ensures that the compiled library can be run in both its standard, tightly coupled and modified, loosely coupled OpenMI configurations. This approach also makes it easy to update the component when a newer version of the model's native code becomes available. Most importantly, legacy codes that have been accumulated by scientists over many years, which are widely used by modelers and accepted by regulatory agencies, can be re-used without having to rewrite them completely to fit a specific coupling framework. Using the wrapper approach, however, presents some challenges. Legacy model codes have been written using a wide variety of programming languages and programming paradigms. It takes some amount of engineering to ensure that the right functions and objects needed for compatibility with a component-based framework are exposed while ensuring that modifications to a native model’s code are minimal. Additionally, the mixing of programming languages and their underlying frameworks can lead to performance costs. The SWMM OpenMI component library used for this study was written using the C# programming language as a wrapper around the native SWMM C programming language code, which was compiled into a Windows dynamic linked library (DLL). The Microsoft .NET framework, on which the C# programming language is built, has a service called the Platform Invocation Service (PInvoke), which provides limited language interoperability capabilities. Function calls and marshalling of SWMM objects to and from the DLL compiled from the native SWMM code was accomplished using this service. Conduits (e.g., pipes, canals, streams), nodes (e.g., junctions, outfalls, dividers), and sub-catchment SWMM objects and their associated properties were exposed as input and output exchange items by creating wrapper classes within the C# code of the OpenMI component we developed. These classes call the native SWMM DLL to get and set the properties of their associated SWMM objects. Although not used in this study, the geographic features associated with the SWMM objects (i.e., points for nodes, lines for conduits, and polygons for sub-catchments) were also exposed as properties of their corresponding exchange items.

Minor modifications were made to the native SWWM code, including adding functions to marshal SWMM objects and their associated properties to the SWMM OpenMI component and vice-versa, and adding an indexed look-up container (i.e., hash map) to cache dynamic boundary conditions supplied by the underlying OpenMI component. The look-up container was added to ensure that boundary conditions could be retrieved efficiently and enforced at appropriate locations in the native SWMM code during the iterative solution procedure. The native SWMM code already contains functions to initialize a component, perform time stepping, write model results, and dispose of model resources upon completion of a simulation. Therefore, there was no need to reproduce them.

Using the PInvoke service as a means of communication between the managed C# wrapper library code and the native SWMM C code introduces some performance costs. PInvoke has an overhead of between 10 and 30 x86 instructions per function call; this is in addition to the fixed cost of marshaling objects (MSDN, 2015). These communication costs across the boundary between managed and unmanaged/native codes are not limited to only C# but to other managed languages (e.g., Java, python) and should be a consideration in the development of components. Although these costs are not necessarily due to the OpenMI standard, the interface specifications and implementations of OpenMI have been provided in C# and Java, which are both managed languages. One could implement the OpenMI standard using an unmanaged programming language like C++, but the same problem will be encountered if one wishes to create a component for a legacy model written using a managed programming language.

The configuration we used to couple any two OpenMI SWMM components adhered to mass and momentum conservation principles to be consistent with how SWMM internally couples two conduits. This was accomplished by first passing flow from an upstream component's conduit to a downstream component's junction node as an inflow boundary condition. This flow was then used to calculate head in the downstream component’s junction node. The calculated head was then passed from the downstream component's junction node back to the upstream component’s outlet node as the water surface elevation boundary condition (Figure 3.3). This coupling configuration gives priority to balancing mass over momentum. The reverse case, where the downstream component supplies water surface elevation to the upstream component first, gives priority to balancing momentum over mass. Details about the differences between the two coupling configurations and their effects on simulated water surface elevations and flows have been discussed in detail by Becker and Talsma (2013).

The sequence of function calls used for the bidirectional coupling between the upstream Component A and the downstream Component B in Figure 3.3 is shown in Figure 3.4. At the beginning of the simulation (time = t_o), Component A must obtain water surface elevation data from Component B before performing its time step. However, Component B must request flow data at time to before proceeding to perform its time stepping calculations. Component A is at time t_o, so it can supply flow to Component B without performing time stepping calculations. Component B uses this data as its inflow boundary condition and proceeds to perform its time stepping calls until its time (time = t_2), just exceeds the time Component A requested water surface elevation (i.e., t_o). Component B then supplies the water surface elevation data at time to, so that Component A can perform its time step. This process is repeated until the trigger component, Component A, finishes its simulation. For coupled components whose time steps are misaligned, an adapter is used to perform the necessary interpolation/extrapolation to ensure data is supplied at the requested time.

The bidirectional, external coupling method that we employed to solve for flow and head at the nodes where two SWMM components are coupled is essentially equivalent to a single internal iteration of the native SWMM computational code between any two connected conduits. This single iteration between any two components is less than the number of iterations performed by the native SWMM model (the native SWMM code uses at least two iterations) and is the main cause of potential model non-convergence and larger TMBE when running coupled SWMM components using OpenMI. We hypothesized that this problem could be addressed by using shorter time steps, which led to several coupled modeling scenarios to test this hypothesis (described in Section 3.3.3).

3.3.2 Study Area

The legacy of agricultural irrigation canals developed early in the history of the City of Logan, Utah continues to influence the management of its stormwater. As the Logan River exits Logan Canyon and flows westward through Cache Valley toward Cutler Reservoir, it is diverted at various points along its length into northward flowing irrigation canals as shown in Figure 3.5.

Streamflow in the ~646-km^2 Logan River watershed is characterized by high spring snowmelt runoff, which is typical of many semi-arid watersheds in the Western United States. Historical average monthly discharge at the USGS gage 1010900 above State Dam at the mouth of Logan Canyon ranges between 105 m^3/s and 625 m^3/s. The same canals that receive irrigation diversions from the Logan River also serve as major conveyance conduits for stormwater from within the city, with many stormwater outfalls piped directly into the canals. Logan, therefore, works collaboratively with private canal owners to ensure that stormwater is managed safely and effectively. Compounding the challenges associated with this arrangement is that, as the city continues to grow, impervious areas increase, leading to increases in peak flows and volumes associated with rainstorms. Modeling the stormwater conveyance system will allow planners to better understand the dynamics that govern the system and allow them to better plan management and mitigation measures for identified areas where capacity to convey stormwater is limited.

3.3.3 Creation of Model Configurations

The processes simulated for this study included surface runoff from a short duration design storm, infiltration, and hydraulic routing. The domain for the reference model we developed for Logan City was comprised of 172 sub-catchments, with areas that ranged from 0.12 to 71 ha, covering a total area of ~2929 ha. Runoff in these sub-catchments was simulated using the NRCS Curve Number method (NRCS, 1972), which accounts for infiltration. Curve Number values ranged between 51 and 98. Overland Manning’s roughness values for the sub-catchments ranged between 0.001 and 0.1. The routing component included 403 conduits, with Manning’s roughness values between 0.01 and 0.035. Conduit lengths totaled ~78 km and ranged between 0.5 m and 2126.3 m (Figure 3.5). The depths and diameters of the conduits ranged between 0.05 to 2.4 m. The routing component also included 397 junction nodes, 39 outfall nodes, 2 weirs, and 38 discharge orifices for 38 storage nodes. The model was executed using a 25-year, 24-hour design storm (~63.5 mm). The high intensity Natural Resources Conservation Service (NRCS) Type-II rainfall distribution curve (Cronshey, 1986) prescribed for Logan was also used to derive the rainfall time series at a 30-minute resolution (Figure 3.6).

Flows diverted from the Logan River into the canals were applied as external boundary conditions using the maximum allowable diversions to be conservative (i.e., the simulation represents the occurrence of the 25-year, 24-hour storm during a time when the agricultural canals are diverting at their maximum rate). These flows were 1.81 m^3/s, 1.36 m^3/s, and 1.39 m^3/s for the Hyde Park and Smithfield, Hyde Park and Logan North Field, and Logan Northwest Field canals respectively. These flows represent initial conditions in the canals, and it was assumed that there was initially no flow in the stormwater conduits. This high intensity storm coupled with the relative gentle slopes in the study area, varying conduit lengths, and flow constrictions at various locations in the city provided a relatively challenging computational application to explore OpenMI's scaling challenges. The simulation was run for 2 days from the beginning of the storm, with outputs produced at 5-minute intervals.

In our first test, we compared the reference model to its corresponding OpenMI component configuration. This test with a single component, no coupling, and therefore no data exchanges, was performed to get a sense of the increase in total simulation time incurred from how the underlying OpenMI specifications were implemented. The source of the increase is from the initialization of the component, its exchange items, and their properties as well as disposal of resources at the end of a simulation.

In our second test, we isolated the runoff from the routing process of the SWMM model by decomposing a copy of the reference model into two model components, one with the sub-catchment runoff process and the other with the hydraulic routing process. The two resulting model components were then coupled using OpenMI. The purpose of this test was to evaluate the increase in total simulation time and TMBE when coupling models by the processes they simulate.

In the final test, we sub-divided the reference model's spatial domain into progressively smaller sub-domains comprised of sub-catchments and their associated conveyance conduits, with each one spanning roughly half the total sub-catchment area of their parent models as shown in Figure 3.7. We then coupled the resulting model components using OpenMI to evaluate the effect of the number of coupling connections on total simulation time and TMBE. Table 1 lists all the model configurations evaluated and their descriptions.

The explicit time marching scheme of both the SWMM computational engine and the OpenMI specification requires a careful selection of a small enough time step to ensure model convergence and accuracy. Smaller time steps help to resolve the discontinuities in flows resulting from the high intensity rainfall and flow boundary conditions applied to the canals and help minimize the spurious oscillations that arise in numerical models with such flows. To explore the tradeoffs between TMBE and total simulation time with changing time steps, we executed all model configurations at constant time steps of 0.5 to 1 second at time intervals of 0.1 seconds and from 1 to 10 seconds at a 1-second interval. Additionally, we executed all model configurations using SWMM’s adaptive time step option with a 10-second maximum time step and time step relaxation factors of 0.1 to 1.0 at intervals of 0.1 to see if any benefits could be gained. This setup required adding an IBaseAdaptedOutput to perform linear temporal interpolations of the flow and water surface elevation data exchanged between the components because of the resulting differences in time steps between components.

The SWMM defaults of 1.5 mm for the convergence criteria and a maximum of eight iterations per time step were used for all model configurations tested. We selected these settings because they produced, what was in our judgment, reasonably small TMBE as a percentage of the total outflow volume plus the final storage for the reference model (e.g., less than 1 %). For the typical 5-second time step, the TMBE as a percentage of the total outflow volume plus final storage for the reference model was only 0.23%.

The operating system on a computer allocates computing resources in an inconsistent fashion. The implication of this variability is that each simulation of the same model will result in a slightly different total simulation time. Therefore, we ran several simulations for each test model configuration to obtain an average of the total simulation time. Ten simulations were run for each configuration. We arrived at this number by timing ~200 shorter test simulations to obtain a value for the standard deviation of the total simulation time to be expected on the machine used. These test simulations yielded a standard deviation of ~0.015 seconds for total simulation time. We used the sample size determination method suggested by Berthouex and Brown (2002) for calculating the number of simulations required to match the standard deviation of the 200 test simulations we ran. At a confidence level of 95%, and an arbitrary confidence interval of 0.01 seconds, the number of simulations to run for each model was calculated as nine simulations. However, this value was rounded up to 10 simulations. All simulations were conducted on a computer with a Quad-Core, Intel® Xeon® 3.00 GHz processor running a 64-bit Windows 7 operating system. Only the SWMM model and OpenMI environment were installed on the machine.

3.4 Results and Discussions

In the first test, we compared the reference model with its corresponding OpenMI implementation. In general, shorter time steps resulted in longer total simulation times as expected because of the resulting increase in the number of time stepping computations (Figure 3.8). The longest total simulation times were 296 seconds and 345 seconds for the reference model and its corresponding OpenMI version, respectively. The shortest total simulation times were 25 seconds and 63 seconds for the reference model and its corresponding OpenMI version respectively. The OpenMI implementation increased total simulation times by an average of about 28 seconds over all the time steps evaluated. The bulk of this increase is a result of the initialization, setup, and resource disposal portions of the SWMM OpenMI component as opposed to the time stepping function calls. We confirmed this by profiling the simulation to examine the time spent in executing portions of the SWMM OpenMI component’s code using Microsoft’s Visual Studio Profiling Tools (Table 2).

The TMBE values for the reference model over all time steps evaluated (Figure 3.9) show a general increase in TMBE with increasing time steps, as expected. TMBE values for the reference model executed in the standard, tightly coupled SWMM configuration, and the OpenMI component version of the full reference model were identical because there were no connections or data exchanges in the OpenMI version. The TMBE results for the OpenMI version of the full reference model are, therefore, not shown.

The largest TMBE occurred for the 10-second time step simulation (-653129 m^3 or 21.6% of the total outflow volume plus final storage volume), while the smallest occurred for the 0.5-second time step simulation (-1961 m^3 or 0.08% of the total outflow volume plus final storage volume). The abrupt jump in the TMBE when the time step was increased from 7 seconds to 8 seconds occurred because the model failed to converge at most of the time steps when they were greater than 7 seconds. For example, the model with the 7-second time step used an average of 2.56 iterations per time step and converged 95.3% of the time as opposed to an average of 6.12 iterations per time step converging only 47.8% of the time for the 8-second time step. This is also responsible for the slight increase in total simulation time when the time step increased from 7 to 8 seconds because more iterations had to be performed for the 8-second time step simulation.

For the adaptive time step simulations, results showed that once the CFL criteria had been satisfied, reducing the time step further by decreasing the time step relaxation factor did not produce any appreciable decrease in TMBE. The smallest TMBE for the adaptive time step model configurations was -1900 m^3 (0.078 % of the total outflow volume plus final storage volume) at a time step relaxation factor of 0.2, while the largest TMBE was -1924 m^3 (0.081% of the total outflow volume plus final storage volume) at a time step relaxation factor of 1.0. The largest TMBE for the adaptive time step was still comparable to the best TMBE for the model with the constant time step value of 0.5 seconds and had the additional benefit of a much shorter total simulation time (45% and 35% less time for the reference model and OpenMI respectively).

The difference in total simulation time between the coupled runoff and routing model and the reference model showed a significant amount of variability with changing time steps (Figure 3.10). This was largely because of the large number of OpenMI connections (332 individual OpenMI connections between the coupled model components) over which data had to be exchanged. The difference in total simulation times was more pronounced for smaller time steps because more OpenMI data exchanges were performed. The TMBE of the coupled runoff and routing model components was, however, not significantly different from the reference model with an average difference of just 1.8% (Figure 3.9). This was primarily because the rainfall data forcing the runoff model was at a temporal resolution of 30 minutes. Therefore, the longest time step of 10 seconds was still small enough to resolve this rainfall.

As with the first test, the adaptive time step option reduced total simulation time while maintaining comparable TMBEs with the best constant time step simulation. The adaptive time step with a relaxation factor of 1.0 for instance, yielded a TMBE of -2036 m^3 (0.086% of the total outflow volume plus final storage volume) versus -2022 m^3 (0.085% of the total outflow volume plus final storage volume) at the 0.5-second constant time step, while decreasing total simulation time by 35%.

In the final test, where we decomposed the reference model spatially, the results showed total simulation time increased linearly with the number of OpenMI connections as shown in Figure 3.11. This occurs because of the resulting increase in data exchanges as the number of connections increase. These results are different than those obtained by Talsma et al. (2012) who observed that the simulation time increased disproportionately as the number of components increased. For example, the percentage increases in total simulation time over the tightly coupled model for their study were 591% for their 2-component simulation and 733% for their 6-component simulation (13 connections). In comparison, for our 0.5-second time step, the percentage increase in total simulation over the tightly coupled simulation was 4% (6 connections) for the 2-component simulation and 17% for the 8-component simulation (18 connections). This improvement may be due to several factors. For one, the Sobek model uses an implicit time marching scheme, which requires solving a system of equations and is generally more computationally expensive than SWMM's explicit time marching scheme. Additionally, how a component is implemented and the degree to which data exchange is optimized by a developer can play role. Finally, the improved data exchange specification for OpenMI 2.0 over OpenMI 1.4 may also have played a role.

Using SWMM's adaptive time step option helped reduce total simulation time for the loosely coupled models (Figure 3.12). This reduction in total simulation time occurs because model components with computational elements having long flow lengths or elements with slow velocities yield longer time steps. This reduces the number of time stepping function calls and helps model components execute faster because of fewer interpolation computations and data exchanges.

Table 2 summarizes results from profiling the OpenMI simulation of the reference model, the 2-component, and the 161-component coupled OpenMI model configurations for the 0.5-second and 10-second time steps. Results are grouped under: (1) percent of time spent reading input files, initializing and disposing components, and saving output files, (2) percent of the time spent calling the time stepping function across the managed and unmanaged code boundary, and (3) time spent marshalling objects back and forth across the managed and unmanaged code boundary. The remainder of the time was largely spent on the data exchange between components in the managed portion of the code at runtime. From the results, the source of the increased total simulation times as the number of components increases and time steps decrease is largely the result of the process of marshalling objects back and forth across the managed and unmanaged code boundary and to a lesser extent data exchange between components in the unmanaged parts of the code. Evidence for this is indicated by the large increase in the proportion of time spent marshalling objects (up to 88.1% of total simulation time for the 161-component simulation at the 0.5-second time step) and reduction in the proportion of total simulation time spent for the remaining functions as the number of coupled components increase and the time step is reduced.

TMBE values for the spatially decomposed models are shown in Figure 3.13. The results showed that decomposing the reference model spatially at least doubled the TMBE of the full reference model in the best-case scenario. However, TMBE as a percentage of total outflow and final storage is a more useful metric for the subjective exercise of determining what the level of acceptability of model errors are. For the dynamic storm, we simulated over a two-day period, only the simulations of the model configuration having 161 OpenMI SWMM components had an error greater than 1% (Figure 3.14), a level below which simulations are likely adequate for many stormwater modeling applications.

For the adaptive time step simulations, the results indicated that large reductions in total simulation time could be obtained while still obtaining TMBE values comparable to the best constant time step simulations. As an example, the adaptive time step simulation with a relaxation factor of 1.0 for the 8-model component simulation, reduced total simulation time by 69.7% over the simulation with the smallest constant time step of 0.5 seconds and still achieved comparable TMBE (-5057 m^3 versus -4811 m^3 for the constant time step).

Finally, to illustrate how the differences in TMBE come about, Figure 3.15 shows the hydrographs for a section of the Logan Northwest Field Canal for all coupling configurations and the 0.5-second time step (which produced the best results in all simulations). As the number of coupled components increases (i.e., roughly representing an increase in the level of spatial decomposition of catchments and associated conduits along the canal), the hydrographs reflect an earlier propagation of the flood wave through the canal than the reference model, as indicated by the earlier rising limbs and recession limbs of the hydrographs produced by the highly-decomposed models. This happens because the single iteration at the OpenMI bi-directional connections is not enough to adequately propagate the effects of downstream water surface elevation boundary conditions upstream.

3.5 Summary and Conclusions

Our testing of the OpenMI 2.0 implementation indicates that the two primary sources for the increased total simulation time for model components in a loosely coupled modeling environment are the costs associated with initialization, setup, and disposal of model components and the costs from data transformations and transfers between components. The increased total simulation time costs associated with the former are dependent on the complexity and number of a model's exchange items. For instance, a two-dimensional hydrodynamic model component, which provides every cell in its computational grid as an exchange item, will likely increase this cost over a comparable SWMM model because more exchange items would have to be initialized and the two-dimensional information being passed would be more complex. However, for a model setup, this cost is largely fixed irrespective of the time step and time domain of the model. For most hydrologic/hydrodynamic model components, this cost should be much less than the increased total simulation time costs from data transformations and transfers between components. This cost can be reduced by providing only those items and properties that are going to be involved in the data exchange process for a specific model component.

The increased total simulation time costs associated with data transformations and transfers between components depend on the complexity of model components and their associated exchange items, the number of model components, as well as the duration of the model time steps. These form the bulk of increases in total simulation time as the number of components increases. Our results showed that the total simulation times scaled linearly with the number of OpenMI connections between model components. Barring reducing the number of connections between model components by consolidating models either by process or by spatial extents, the straightforward way to deal with the total simulation time increases is to increase the time step of a model component. This decreases the overall number of time stepping function calls and, therefore, the number of data transformations and transfers between model components. Results from profiling our model simulations suggest that marshalling objects across the managed and unmanaged code boundary in the data transfer process becomes particularly expensive as the number of coupled components increases. One potential approach to avoid this in C# would be to pass values as blittable types (i.e., types that have the same representation in managed and unmanaged code - e.g. integers, doubles, etc.) rather than marshalling entire objects as was done in this study. There are no marshalling costs for blittable types (MSDN, 2015). However, this may require extensive modifications to the native computational code of the model to ensure that proper context is given to the data being exchanged. Additionally, the ability to utilize the very useful self-introspection and method invocation capabilities provided by C# (and other managed programming languages) on marshalled objects will be lost.

Although increasing time step reduces total simulation time, there is a tradeoff between time step and TMBE. Increasing the time step for time marching models like SWMM reduces their ability to resolve discontinuities in time varying data (e.g., mass, velocity, constituent concentrations) being transported across their space discretizations, resulting in a reduction of the conservation quality of a model. A general increase in TMBE with increasing time step was observed in the results from all our tests. To be able to use longer time steps without diminishing the accuracy of model results, unnecessarily short/small computational elements (e.g., conduits, cells) should be avoided when performing the space discretization.

Most explicit time marching models like the SWMM allow modelers to select an adaptive time step option. This option calculates an optimal time step to use to comply with the CFL stability criteria. Using this option can help decrease total simulation times by increasing the time step for components with longer/larger computational elements or computational elements with slow velocities. Different time steps are realized for different model components, and so a model component receiving data from another model component must therefore, have a short enough time step to resolve the time varying data being supplied to it to minimize errors. Although we only used the linear interpolation method to interpolate data for the temporally misaligned model components, the use of other more efficient interpolation, extrapolation, and data aggregation methods with better conservation properties is an area of research we will continue to pursue.

Since the bidirectional coupling configuration we employed is essentially equivalent to a single iteration of SWMM's internal calculations, an iterative procedure applied across OpenMI connections that checks for model convergence at each connection would theoretically help minimize TMBE. In OpenMI, this can be accomplished by placing an iterative controller model component between OpenMI connections. The increased total simulation time resulting from this approach may, however, be prohibitive for most applications. For most cases, the adaptive time step option coupled with a careful selection of a small enough maximum time step and an appropriate interpolation method may be enough to obtain acceptable TMBE. The degree of acceptability of a TMBE is based on professional judgment or regulatory requirements and can be evaluated by calculating the volume of the error as a percentage of the total outflows plus final storage volume in a model.

The work we have presented here evaluates the costs associated with using an implementation of the OpenMI standard in terms of total simulation times and TMBE. It highlights the importance of striking a balance between the benefits promised by component-based modeling frameworks and the costs that are incurred from their use. It also illustrates the importance of considering time stepping in model coupling, especially where spatial and/or temporal discontinuities may occur at the connections between coupled model components. These are critical considerations for loose model coupling applications using any combination of models and will be instructive for modelers investigating loose coupling for new model development. Although we have illustrated that the costs of loosely coupled modeling can be significant, users may still wish to apply this modeling strategy where experimentation with process formulations is required, where advanced data exchange between models is required, or where multidisciplinary process representations must be coupled. These benefits may outweigh the associated costs. We have illustrated some strategies for minimizing the costs, and future work will evolve around developing efficient data transformation algorithms to resolve scale disparities that arise between model components.

Software Availability

Software and libraries: We developed the SWMMOpenMIComponent (a C# SWMM Component), SWMMOpenMINoGlobals (a modified, native C SWMM computational engine code underlying the component), and a modified OpenMI C# project. We forked the OpenMI 2.0 C# project, including the Software Development Kit (SDK), the command line interface, and the OpenMI Configuration Editor found at http://sourceforge.net/p/openmi/code/HEAD/tree/trunk/src/csharp for this study. In addition to implementing minor bug fixes to ensure that the code compiled, we implemented a new graphical user interface for creating connections with chained adapters in accordance with the OpenMI 2.0 specification, a new simulation monitoring dialog, and fixed the project file reading and writing classes to ensure that connections with adapters are read and written properly.

Hardware: PC running Microsoft Windows

License: The C# SWMM Component (SWMMOpenMIComponent) and its underlying modified native C SWMM library (SWMMOpenMINoGlobals) are freely available under the GNU Lesser General Public License (LGPL) license at https://github.com/cbuahin/SWMMOpenMIComponent. The source code for the modified version of the OpenMI C# project can be found at https://github.com/cbuahin/OpenMI under the LGPL license.

Acknowledgments

This research was supported by National Science Foundation EPSCoR Grant IIA 1208732 awarded to Utah State University as part of the State of Utah EPSCoR Research Infrastructure Improvement Award. Any opinions, findings, and conclusions or recommendations expressed are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Abbott, M. B., Bathurst, J. C., Cunge, J. A., O'Connell, P. E., and Rasmussen, J. (1986). "An introduction to the European Hydrological System — Systeme Hydrologique Europeen, 'SHE', 1: History and philosophy of a physically-based, distributed modelling system." Journal of Hydrology, 87(1–2), 45–59. DOI:10.1016/0022-1694(86)90114-9

Argent, R. M., Grayson, R. B., and Ewing, S. A. (1999). "Integrated models for environmental management: Issues of process and design." Environment International, 25(6–7), 693–699. DOI:10.1016/S1364-8152(03)00029-X

Becker, B., and Talsma, J. (2013). "On the External and Iterative Coupling of Multiple Open Channel Flow Models with OpenMI." Revista de Ingeniería Innova, 6, 51–56.

Blind, M., Gregersen, J. B., Krause, P., Kralisch, S., and Flügel, W. (2005). "Towards an Open Modelling Interface (OpenMI) the HarmonIT project." Advances in Geosciences, 4.

Brown, L. C., and Berthouex, P. M. (2002). Statistics for Environmental Engineers, Second Edition. CRC Press, Boca Raton.

Bulatewicz, T., Andresen, D., Auvenshine, S., Peterson, J., and Steward, D. R. (2014). "A distributed data component for the Open Modeling Interface." Environmental Modelling & Software, 57, 138–151. DOI:10.1016/j.envsoft.2014.02.017

Buschmann, F. (1996). Pattern-Oriented Software Architecture Volume 1: A System of Patterns. Wiley, Chichester ; New York.

Castronova, A. M., and Goodall, J. L. (2013). "Simulating watersheds using loosely integrated model components: Evaluation of computational scaling using OpenMI." Environmental Modelling & Software, 39, 304–313. DOI: 10.1016/j.envsoft.2012.01.020

Castronova, A. M., Goodall, J. L., and Ercan, M. B. (2013). "Integrated modeling within a Hydrologic Information System: An OpenMI based approach." Environmental Modelling & Software, 39, 263–273. DOI:10.1016/j.envsoft.2012.02.011

Courant, R., Friedrichs, K., and Lewy, H. (1928). "Über die partiellen Differenzengleichungen der mathematischen Physik." Mathematische Annalen, 100(1), 32–74. DOI:10.1007/BF01448839

Cronshey, R. (1986). Urban hydrology for small watersheds. US Dept. of Agriculture, Soil Conservation Service, Engineering Division.

David, O., Markstrom, S., Rojas, K., Ahuja, L., and Schneider, I. (2002). “The Object Modeling System.” Agricultural System Models in Field Research and Technology Transfer, L. Ahuja, L. Ma, and T. Howell, eds., CRC Press.

Diersch, H.-J. (2014). FEFLOW - Finite Element Modeling of Flow, Mass and Heat Transport in Porous and Fractured Media.

Elag, M. M., Goodall, J. L., and Castronova, A. M. (2011). “Feedback loops and temporal misalignment in component-based hydrologic modeling: Component-Based Hydrologic Modeling.” Water Resources Research, 47(12). DOI:10.1029/2011WR010792

Fenske, J., Banta, E. R., Piper, S., Donchyts, G., and Wexler, E. J. (2011). “Coupling HEC-RAS and MODFLOW using OpenMI.” MODFLOW and More: Integrated Hydrologic Modeling.

Fröhlich, P. H., and Franz, M. (1999). Component-oriented programming in object-oriented languages. Information and Computer Science, University of California, Irvine.

Goodall, J. L., Robinson, B. F., and Castronova, A. M. (2011). “Modeling water resource systems using a service-oriented computing paradigm.” Environmental Modelling & Software, 26(5), 573–582. DOI:10.1016/j.envsoft.2010.11.013

Goodall, J. L., Saint, K. D., Ercan, M. B., Briley, L. J., Murphy, S., You, H., DeLuca, C., and Rood, R. B. (2013). “Coupling climate and hydrological models: Interoperability through Web Services.” Environmental Modelling & Software, 46, 250–259. DOI:10.1016/j.envsoft.2013.03.019

Gregersen, J. B., Gijsbers, P. J. A., Westen, S. J. P., and Blind, M. (2005). “OpenMI: the essential concepts and their implications for legacy software.” Advances in Geosciences, 4(4), 37–44. DOI:10.5194/adgeo-4-37-2005

Gregersen, J. B., P. J. A. Gijsbers, and S. J. P. Westen. (2007). “OpenMI: Open Modelling Interface.” Journal of Hydroinformatics, 9(3), 175. DOI:10.2166/hydro.2007.023

Gross, H.-G. (2005). Component-Based Software Testing with UML. Springer Science & Business Media.

Hill, C., DeLuca, C., Suarez, M., and Da Silva, A. (2004). “The architecture of the earth system modeling framework.” Computing in Science & Engineering, 6(1), 18–28. DOI:10.1109/MCISE.2004.1255817

Hunt, B. (1999). “Unsteady Stream Depletion from Ground Water Pumping.” Ground Water, 37(1), 98–102. DOI:10.1111/j.1745-6584.1999.tb00962.x

Knapen, R., Janssen, S., Roosenschoon, O., Verweij, P., de Winter, W., Uiterwijk, M., and Wien, J.-E. (2013). “Evaluating OpenMI as a model integration platform across disciplines.” Environmental Modelling & Software, Thematic Issue on the Future of Integrated Modeling Science and Technology, 39, 274–282. DOI:10.1016/j.envsoft.2012.06.011

Löwy, J. (2003). .NET Components. O’Reilly.

Moore, R. V., and Tindall, C. I. (2005). “An overview of the open modelling interface and environment (the OpenMI).” Environmental Science & Policy, Research & Technology Integration in Support of the European Union Water Framework Directive Proceedings of a workshop held in Ghent (Belgium) on 4-5 October 2004, 8(3), 279–286. DOI:10.1016/j.envsci.2005.03.009

MSDN. (2015). “Calling Native Functions from Managed Code.” Microsoft Developer Network, <https://msdn.microsoft.com/en-us/library/ms235282.aspx> (May 21, 2015).

NRCS. (1972). SCS National Engineering Handbook, Section 4: hydrology. Soil Conservation Service.

OpenMI Association (2014). “OpenMI Software Development Kit.” <http://sourceforge.net/projects/openmi/> (Nov. 21, 2014).

Peckham, S. D., Hutton, E. W. H., and Norris, B. (2013). “A component-based approach to integrated modeling in the geosciences: The design of CSDMS.” Computers & Geosciences, 53, 3–12. DOI:10.1016/j.cageo.2012.04.002

Refsgaard, J. C., Storm, B., and Clausen, T. (2010). “Système Hydrologique Europeén (SHE): review and perspectives after 30 years development in distributed physically-based hydrological modelling.” Hydrology Research, 41(5), 355. DOI:10.2166/nh.2010.009

Rossman, L. A. (2006). Storm Water Management Model, Quality Assurance Report: Dynamic Wave Flow Routing. US Environmental Protection Agency, Office of Research and Development, National Research Management Research Laboratory.

Rossman, L. A. (2010). Storm water management model user’s manual, version 5.0. National Risk Management Research Laboratory, Office of Research and Development, US Environmental Protection Agency.

Shrestha, N. K., Leta, O. T., De Fraine, B., van Griensven, A., and Bauwens, W. (2013). “OpenMI-based integrated sediment transport modelling of the river Zenne, Belgium.” Environmental Modelling & Software, 47, 193–206. DOI:10.1016/j.envsoft.2013.05.004

Stelling, G. S., and Duinmeijer, S. P. A. (2003). “A staggered conservative scheme for every Froude number in rapidly varied shallow water flows.” International Journal for Numerical Methods in Fluids, 43(12), 1329–1354. DOI:10.1002/fld.537

Sui, D. Z., and Maggio, R. C. (1999). “Integrating GIS with hydrological modeling: practices, problems, and prospects.” Computers, environment and urban systems, 23(1), 33–51. DOI: 10.1016/S0198-9715(98)00052-0

Szyperski, C. (2002). Component Software: Beyond Object-oriented Programming. ACM Press.

Szyperski, C., and Pfister, C. (1997). “Workshop on Component-Oriented Programming, Summary.” Special Issues in Object-Oriented Programming – ECOOP ’96 Workshop Reader, Dpunkt Verlag, Heidelberg.

Talsma, J., Becker, B. P. J., Gao, Q., and Ruijgh, E. (2012). “Coupling of multiple channel flow models with OpenMI.” 10th International Conference on Hydroinformatics HIC.

Vanecek, S., and Moore, R. (2014). “OGC® Open Modelling Interface Standard.” Open Geospatial Consortium. <http://www.opengis.net/doc/is/openmi/2.0> (May. 05, 2014).

Warner, J. C., Perlin, N., and Skyllingstad, E. D. (2008). “Using the Model Coupling Toolkit to couple earth system models.” Environmental Modelling & Software, 23(10–11), 1240–1249. DOI:10.1016/j.envsoft.2008.03.002

Yamagata, K., Butts, M. B., Grooss, J., Clausen, T. H., Graham, D. N., Clausnitzer, V., Gründler, R., and Monninkhoff, B. (2012). “OpenMI coupling of FEFLOW and MIKE SHE.” 34 th Hydrology and Water Resources Symposium, Sydney, Australia.

Tables

Table 3.1 Coupled model configurations descriptions.

Model Configurations	Description	Number of OpenMI Components	Number of OpenMI Connections
Ref. SWMM Model	Reference model using the standard tightly coupled SWMM configuration	N/A	N/A
Ref. Model OpenMI	Full reference model using OpenMI	1	0
Routing & Runoff	Coupled routing and runoff model using OpenMI	2	332
Ref. Model Split Into 2 Components	Reference model decomposed spatially into 2 OpenMI components	2	6
Ref. Model Split Into 4 Components	Reference model decomposed spatially into 4 OpenMI components	4	10
Ref. Model Split Into 8 Components	Reference model decomposed spatially into 8 OpenMI components	8	18
Ref. Model Split Into 16 Components	Reference model decomposed spatially into 16 OpenMI components	16	40
Ref. Model Split Into 32 Components	Reference model decomposed spatially into 32 OpenMI components	32	74
Ref. Model Split Into 64 Components	Reference model decomposed spatially into 64 OpenMI components	64	138
Ref. Model Split Into 161 Components	Reference model decomposed spatially into 161 OpenMI components	161	336

Table 3.2 Results from profiling a sample of the model configurations.

Model Configurations	Time step (s)	% Time Spent Reading Input Files, Initializing and Disposing Components, and Saving Output Files	% Time Spent Calling Time stepping Function Across Managed and Unmanaged Code Boundary	% Time Spent Marshalling Objects Across Managed and Unmanaged Code Boundary
Ref. Model OpenMI	0.5	3.8	96.1	0.0
Ref. Model Split Into 2 Components	0.5	4.9	82.0	3.0
Ref. Model Split Into 161 Components	0.5	0.9	1.1	88.1
Ref. Model OpenMI	10	38.4	61.4	0.0
Ref. Model Split Into 2 Components	10	48.8	43.7	1.2
Ref. Model Split Into 161 Components	10	4.8	10.8	5.7

Figures

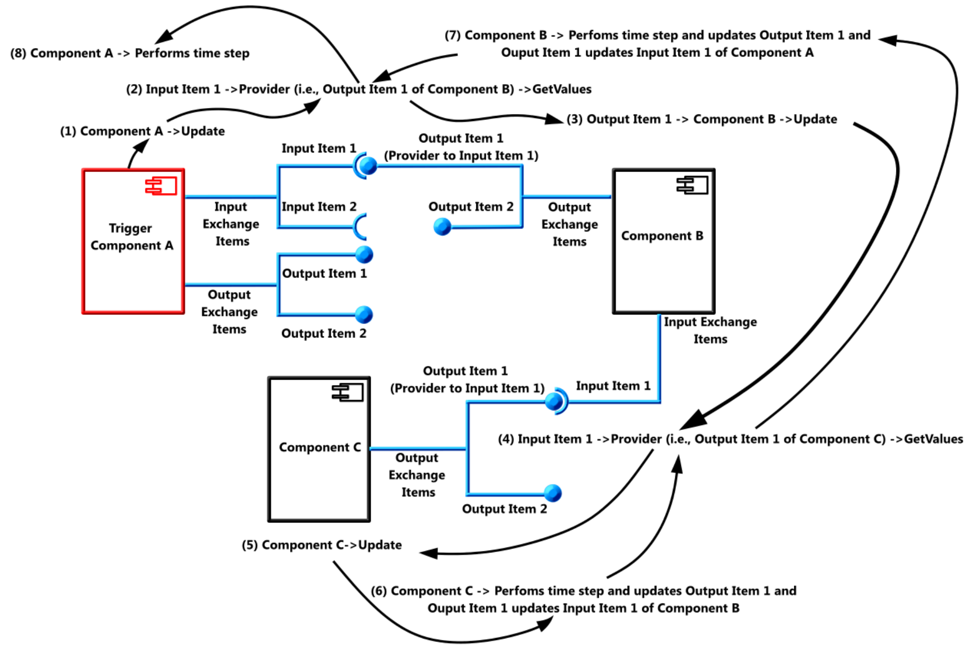

(8) Component A -> Performs time step
(7) Component B -> Perfoms time step and updates Output Item 1 and
Ouput Item 1 updates Input Item 1 of Component A
(2) Input Item 1 ->Provider (i.e., Output Item 1 of Component B) ->GetValues
(3) Output Item 1 -> Component B ->Update
(1) Component A ->Update
Input Item 1
Output Item 1
(Provider to Input Item 1)
Trigger
Component A
Input
Exchange
Items
Input Item 2
Output Item 2
Output
Exchange
Items
Component B
Output Item 1
Output
Exchange
Items
Output Item 2
Input Exchange
Items
Output Item 1
(Provider to Input Item 1)
Input Item 1
Component C
(4) Input Item 1 ->Provider (i.e., Output Item 1 of Component C) ->GetValues
Output
Exchange
Items
Output Item 2
(5) Component C->Update
(6) Component C -> Perfoms time step and updates Output Item 1 and
Ouput Item 1 updates Input Item 1 of Component B

Figure 3.1 Unidirectional OpenMI data exchange between three time stepping Components A, B, and C. Component C supplies data to Component B, which supplies data to Component A.

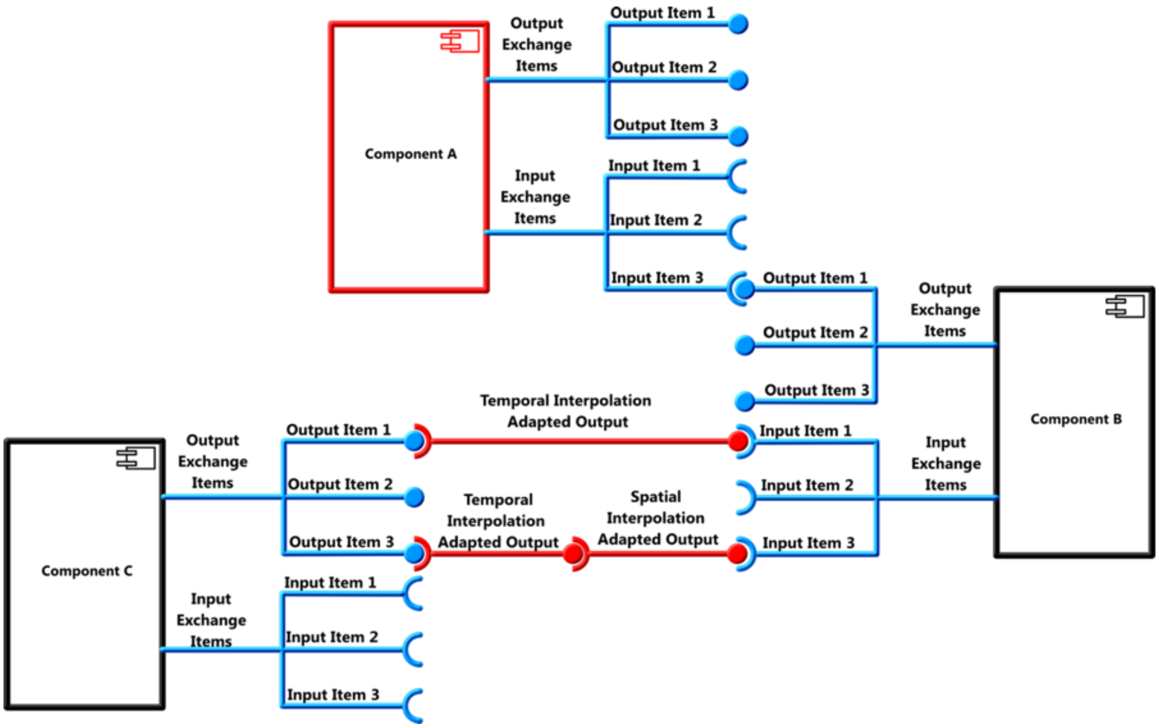


Figure 3.2 Example of adapted outputs in an OpenMI composition of three Components A, B, and C. Output Item 1 of Component C is temporally interpolated by an adapter and supplied to Input Item 1 of Component B. Output Item 3 is temporally interpolated and then spatially interpolated by chained adapters and supplied to Input Item 3 of Component B.

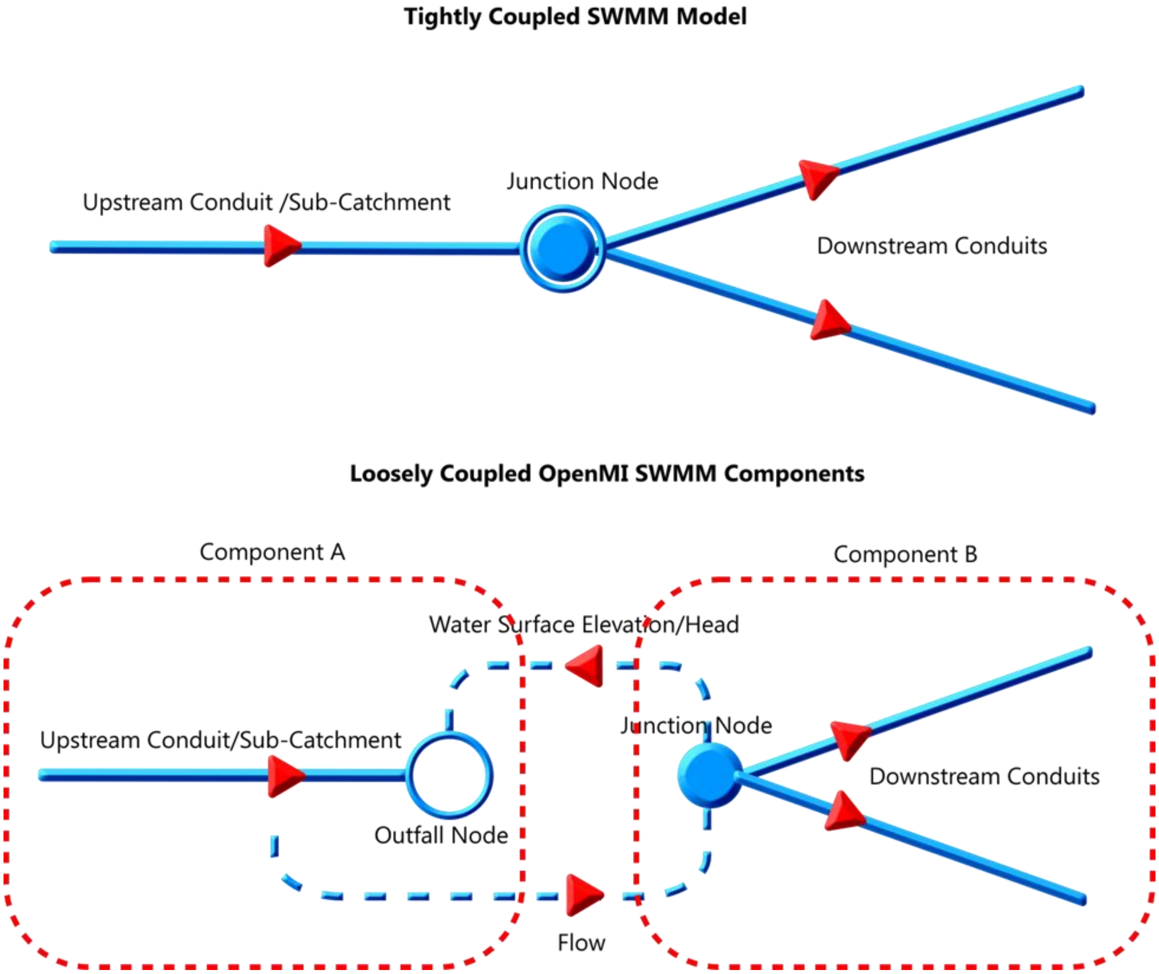

Tightly Coupled SWMM Model
Upstream Conduit /Sub-Catchment
Junction Node
Downstream Conduits
Loosely Coupled OpenMI SWMM Components
Component A
Component B
Water Surface Elevation/Head
Upstream Conduit/Sub-Catchment
Junction Node
Downstream Conduits
Outfall Node
Flow

Figure 3.3 SWMM OpenMI component coupling configuration between upstream Component A and downstream Component B.

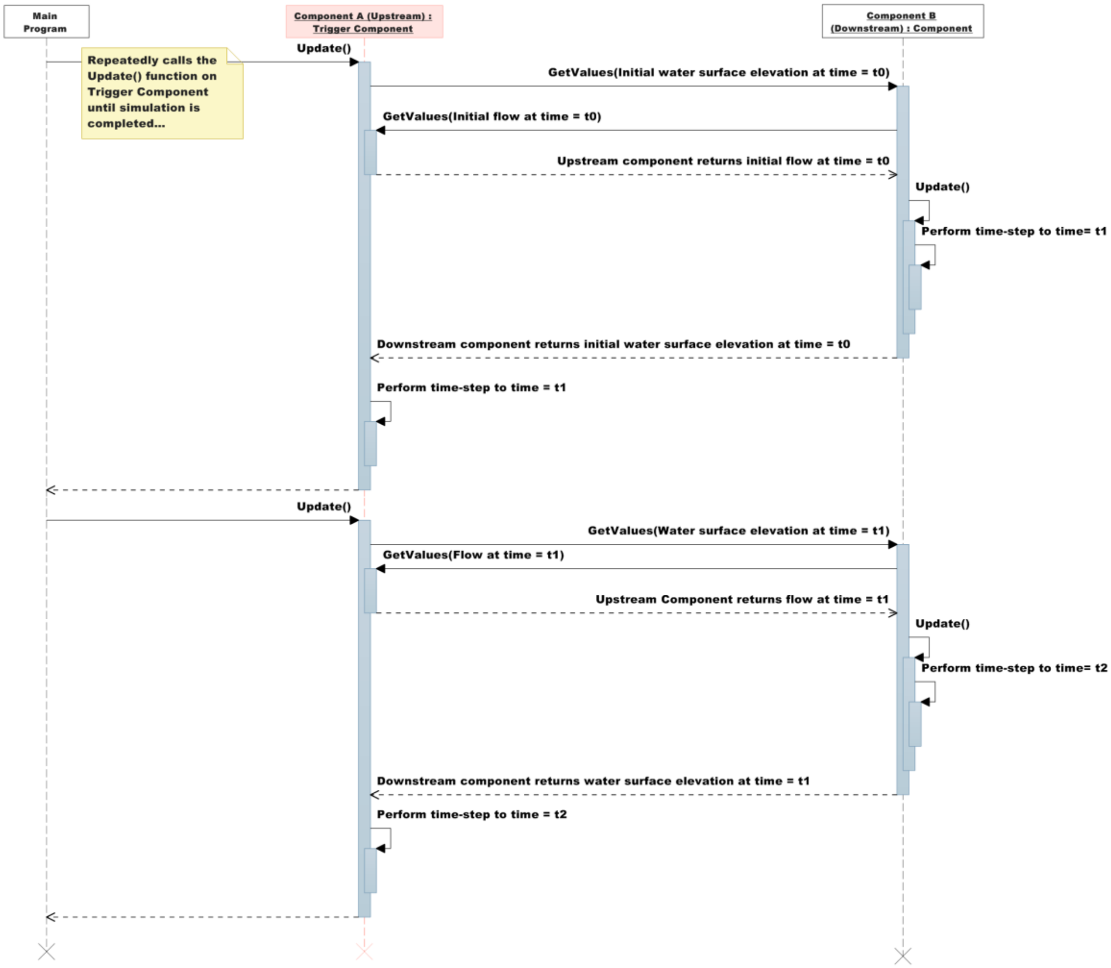

Main Program
Component A (Upstream) : Trigger Component
Component B (Downstream) : Component
Repeatedly calls the Update() function on Trigger Component until simulation is completed...
Update()
GetValues(Initial water surface elevation at time = t0)
GetValues(Initial flow at time = t0)
Upstream component returns initial flow at time = t0
Update()
Perform time-step to time= t1
Downstream component returns initial water surface elevation at time = t0
Perform time-step to time = t1
Update()
GetValues(Water surface elevation at time = t1)
GetValues(Flow at time = t1)
Upstream Component returns flow at time = t1
Update()
Perform time-step to time= t2
Downstream component returns water surface elevation at time = t1
Perform time-step to time = t2

Figure 3.4 Sequence diagram for bidirectional data exchange between an upstream SWMM Component A, and a downstream SWMM Component B.

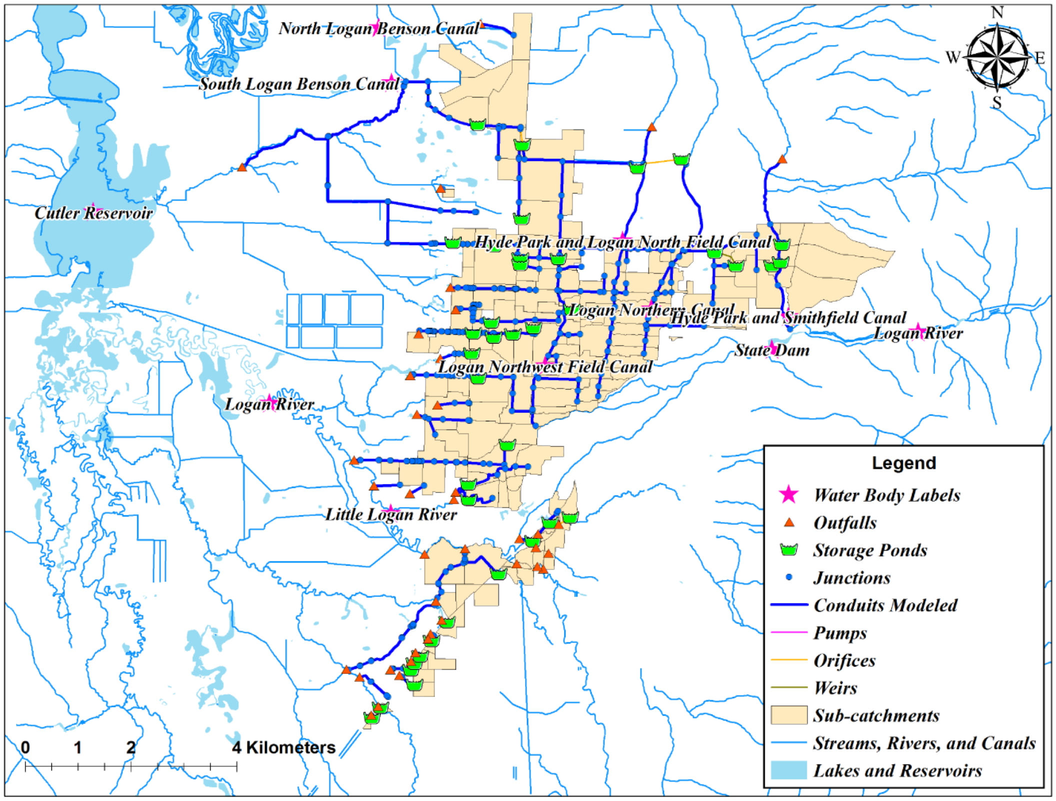


Figure 3.5 Modal spatial domain.

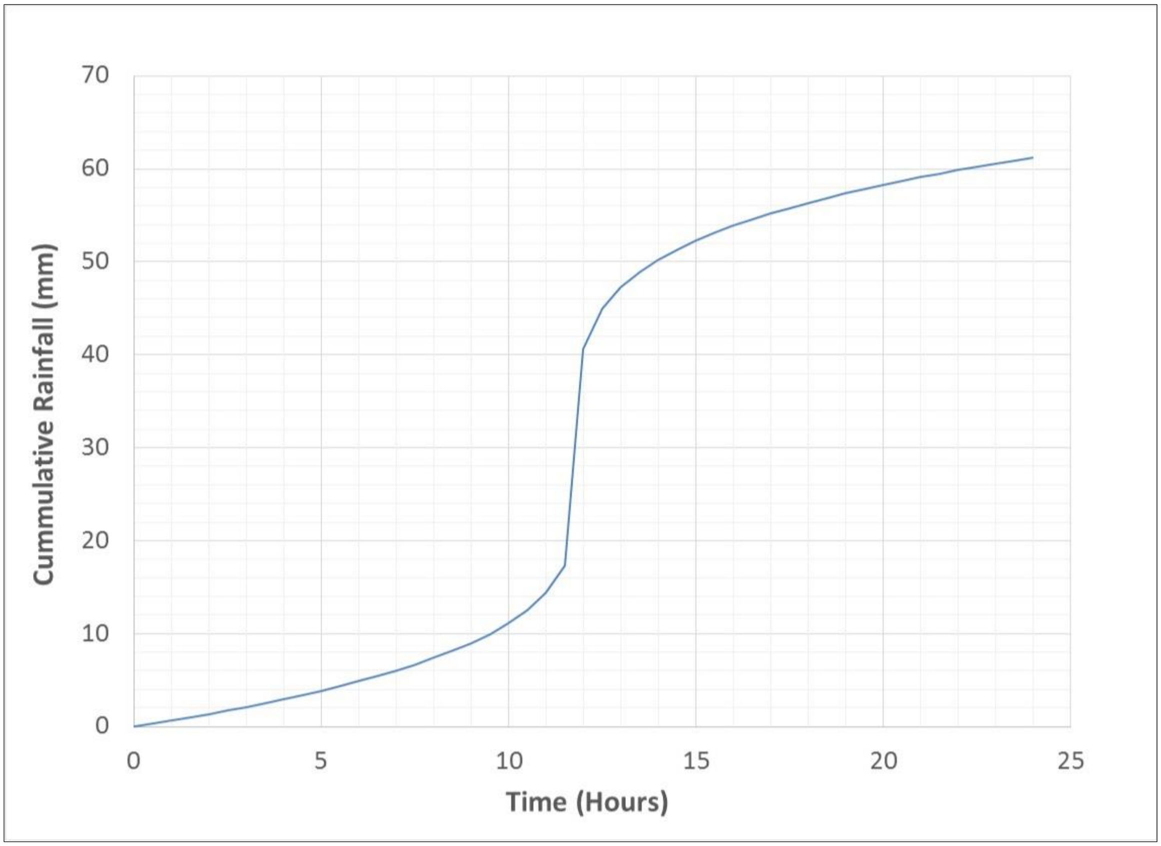


Figure 3.6 Design storm hyetograph shown as a cumulative distribution

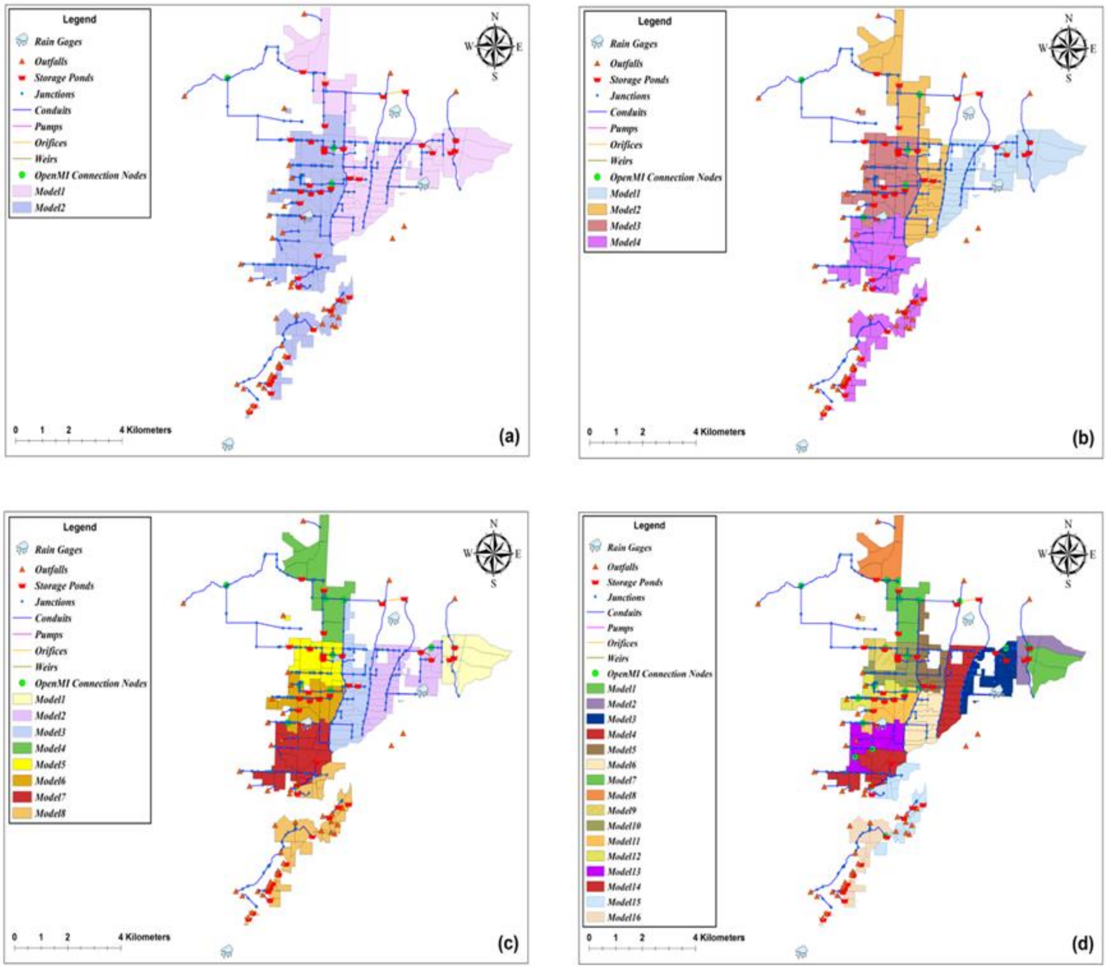


Figure 3.7 Reference SWMM model spatially decomposed into: (a) two, (b) four, (c) eight, and (d) 16 OpenMI SWMM model components.

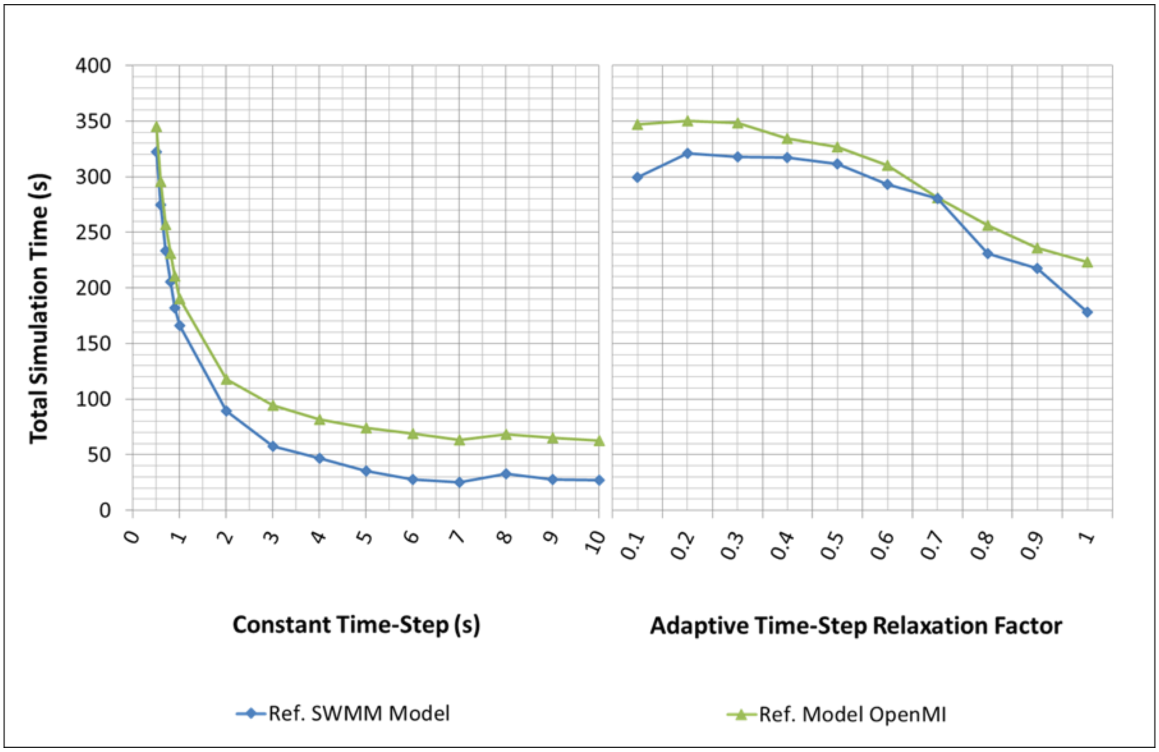


Figure 3.8 Total simulation times of the reference model executed using the standard SWMM configuration versus execution as an OpenMI component.

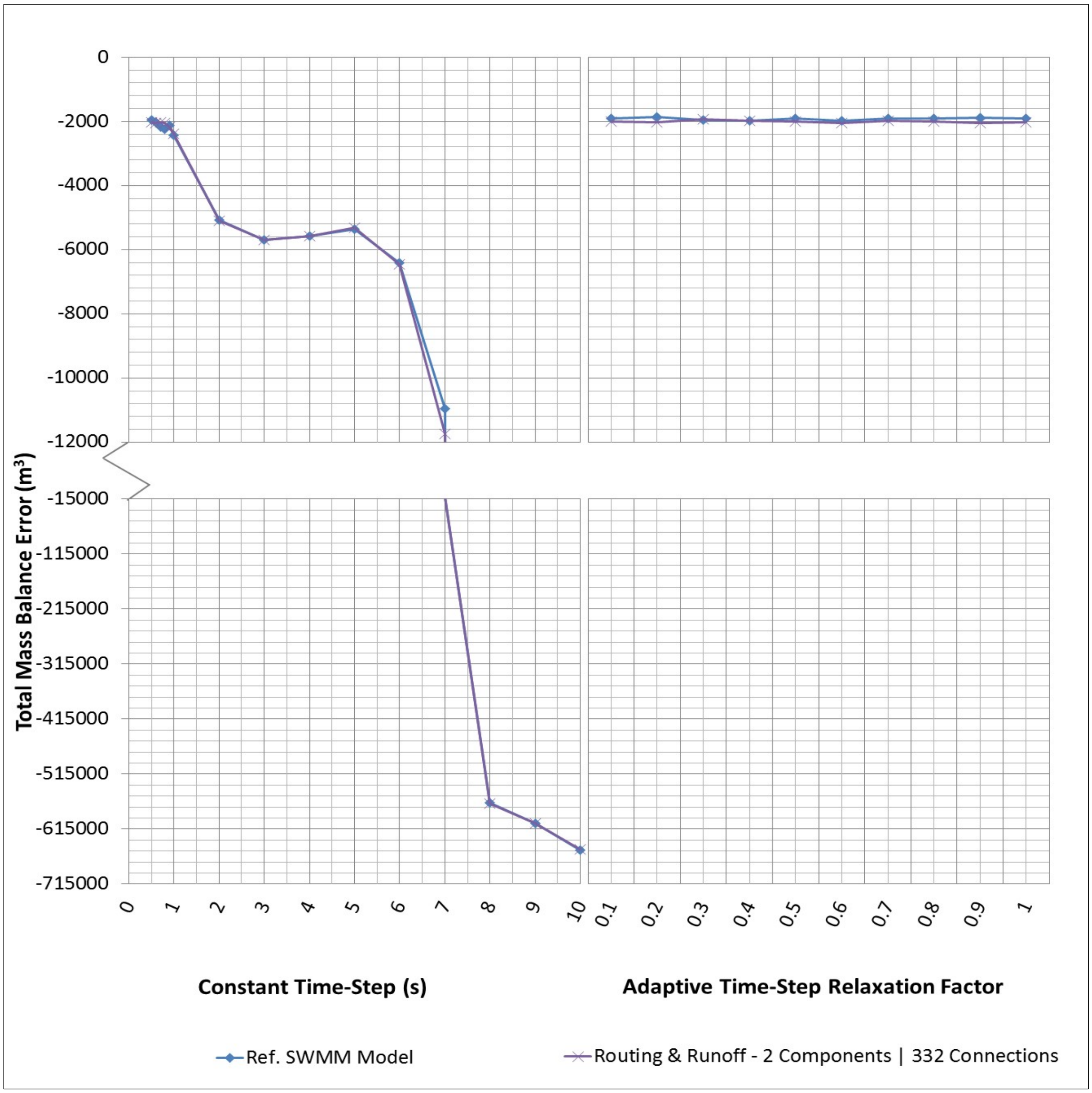


Figure 3.9 TMBE of the reference model versus coupled runoff and routing model OpenMI components.

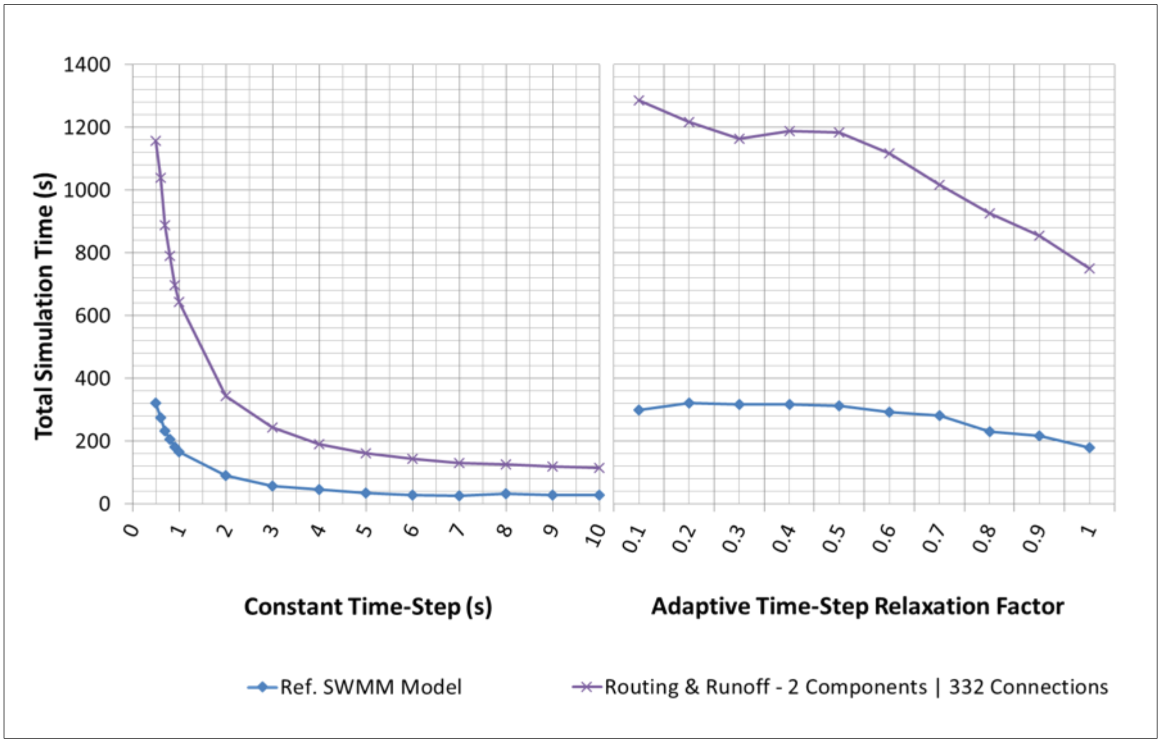


Figure 3.10 Total simulation time of the reference model versus OpenMI coupled runoff and routing.

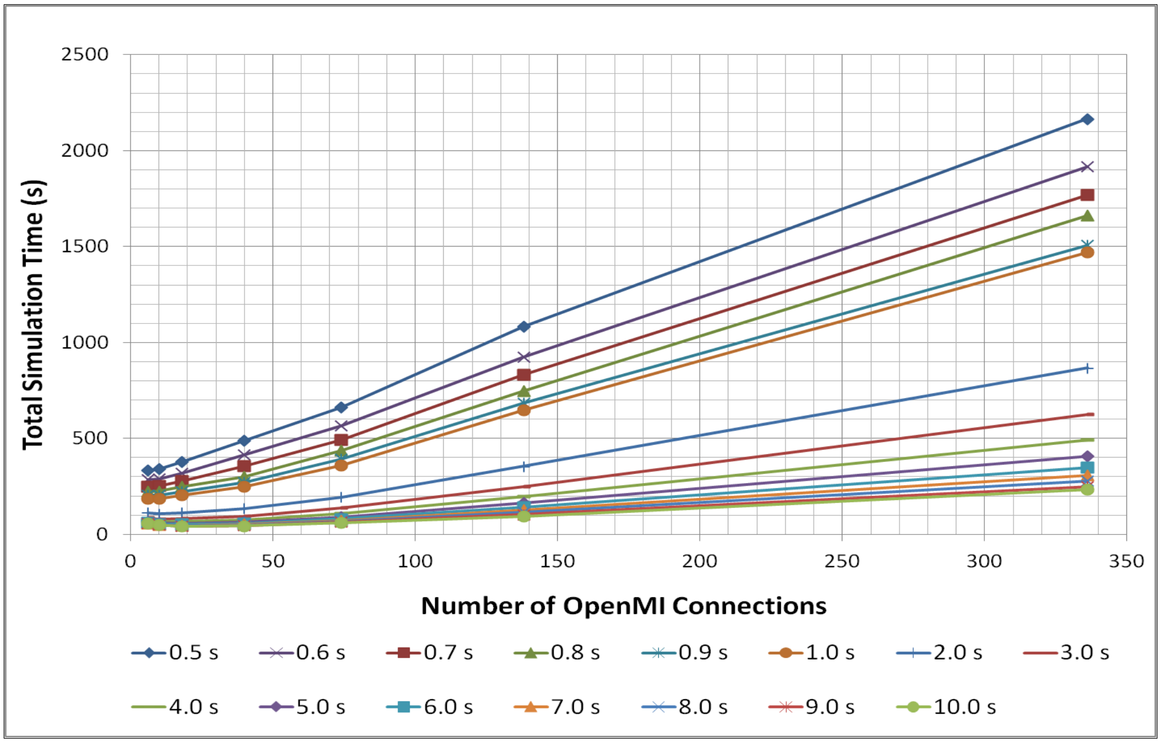


Figure 3.11 Total simulation time versus number of OpenMI connections for spatially decomposed models and different simulation time step lengths.

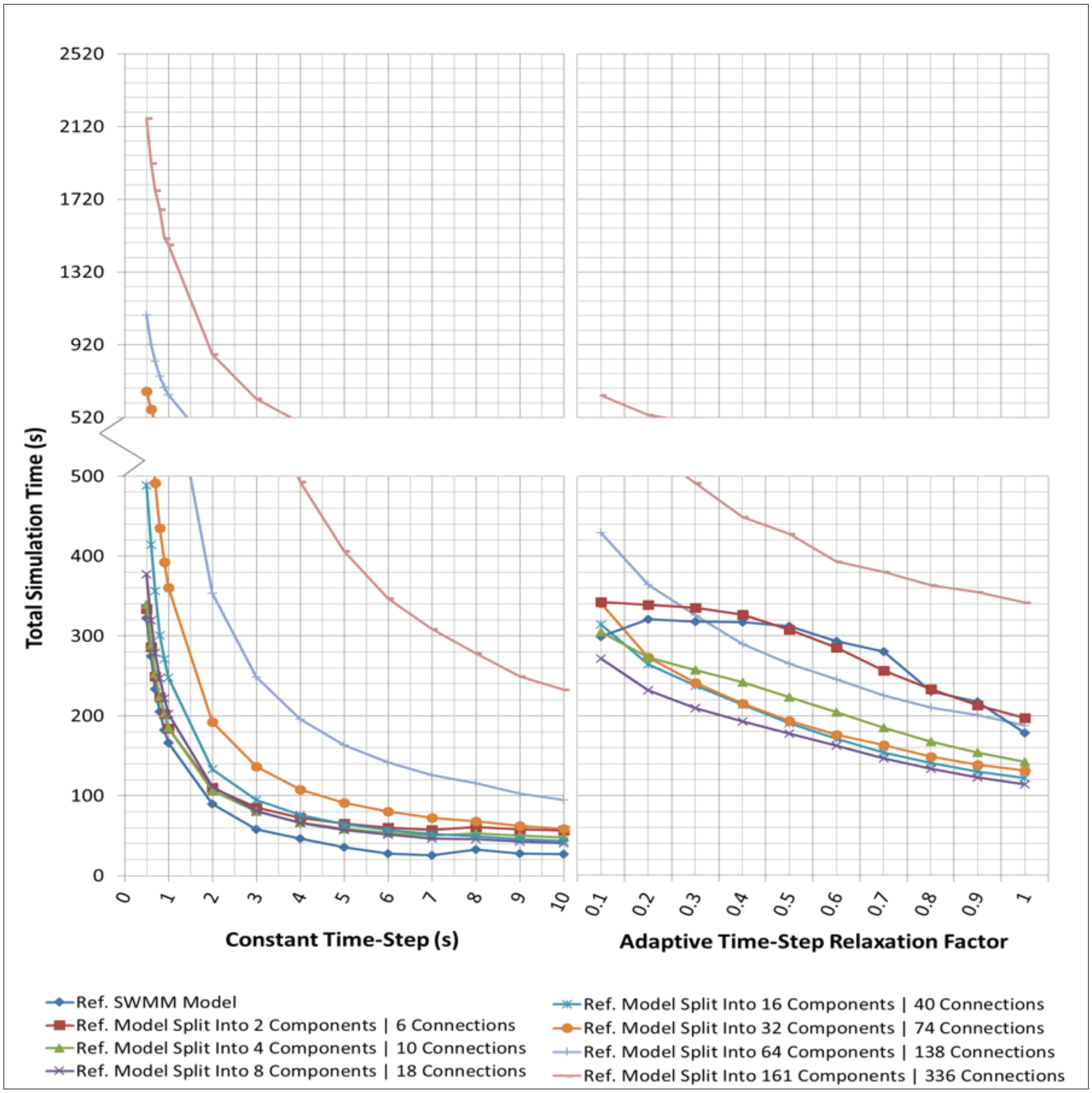


Figure 3.12 Total simulation time of the reference model versus spatially decomposed models with different time step settings.

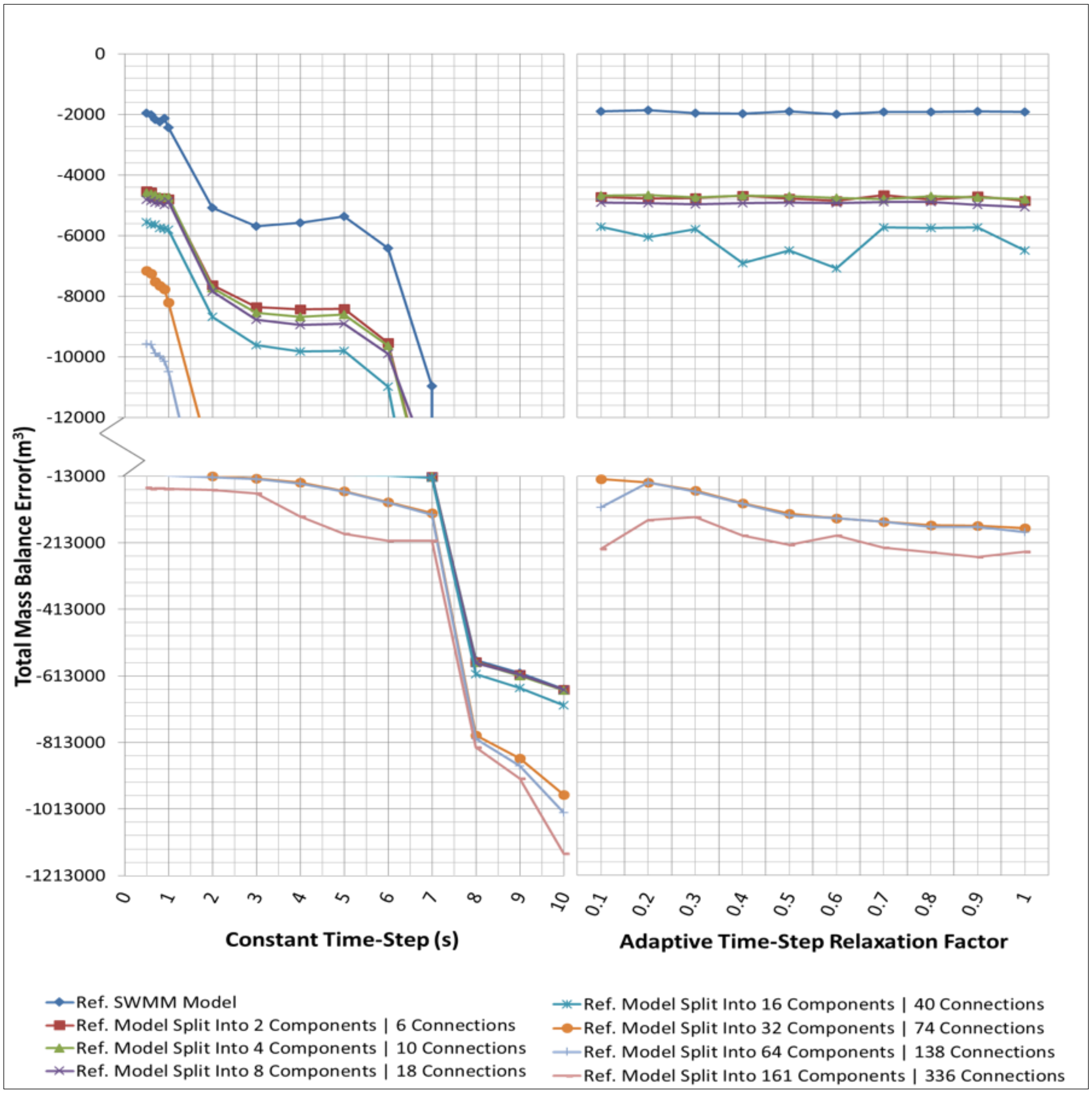


Figure 3.13 TMBE of the reference model versus spatially decomposed models with different time step settings.

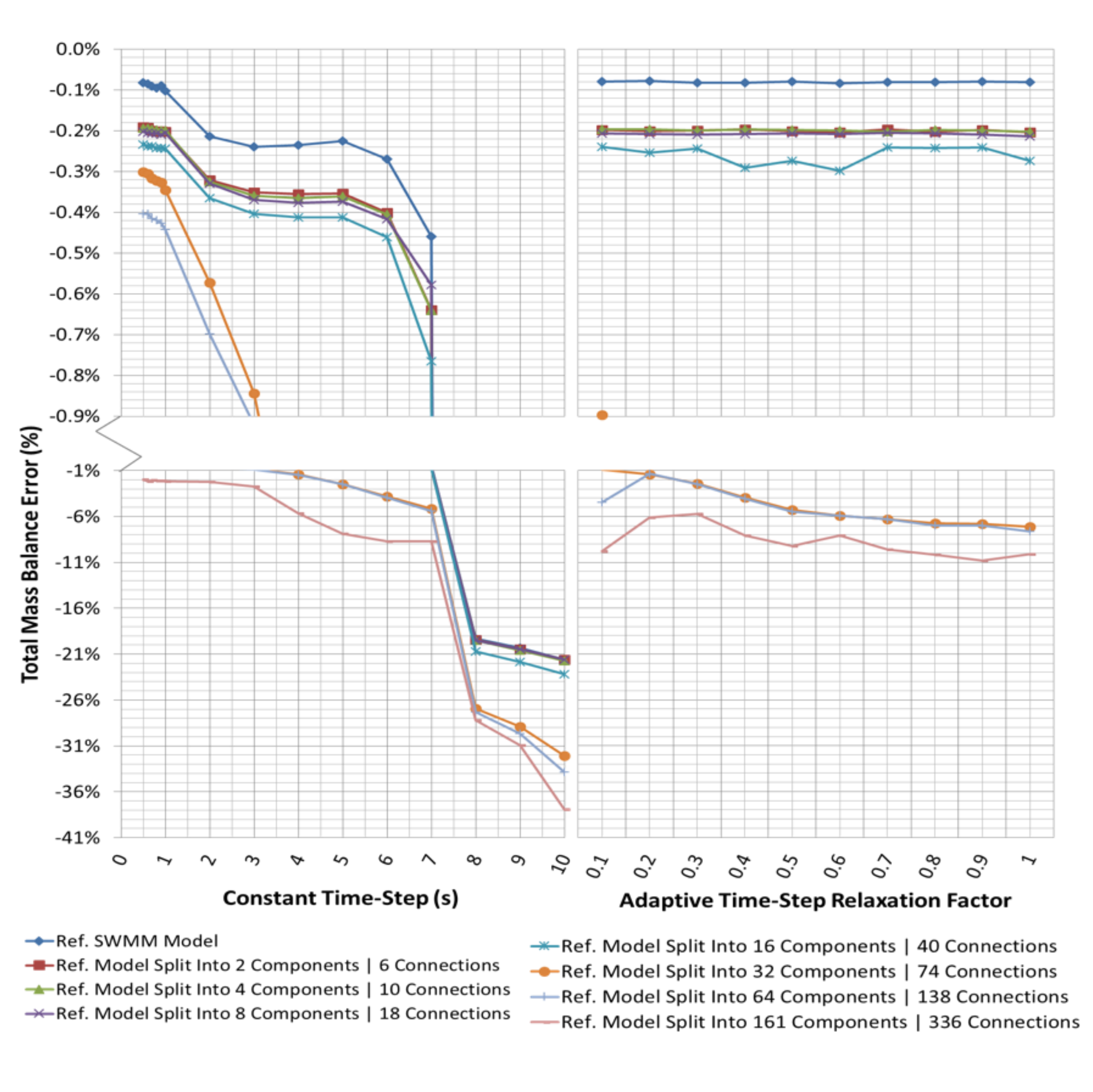


Figure 3.14 TMBE as a percentage of total outflows plus final storage for the reference model versus spatially decomposed models with different time step settings.

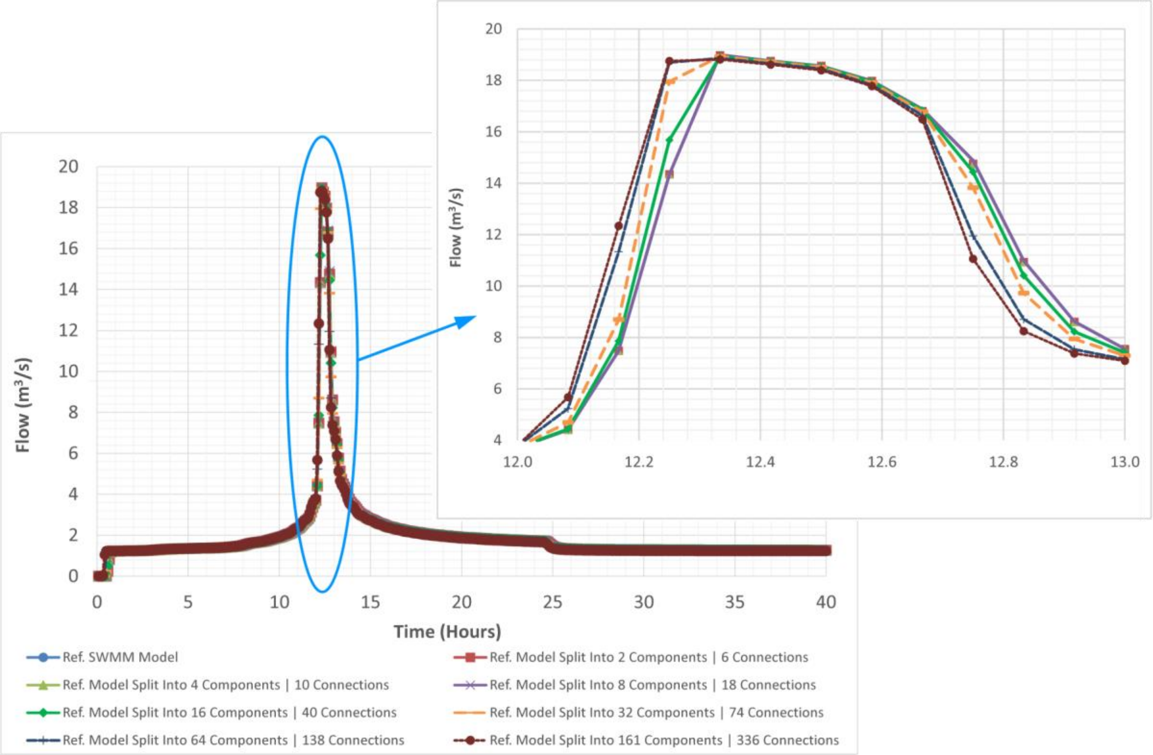


Figure 3.15 Hydrographs of the reference model versus spatially decomposed models at the 0.5-second time step on the Logan Northwest Field canal.

CHAPTER 4

HYDROCOUPLE: ADVANCING COMPONENT-BASED MODELING FRAMEWORKS FOR INTEGRATED WATER RESOURCES MODELING[1]

Abstract

The use of existing component-based modeling frameworks for integrated water resources modeling is currently hampered for some important use cases because they lack support for commonly used, topology-aware, geo-temporal data structures. Additionally, existing frameworks are often accompanied by large software stacks with steep learning curves. Others lack specifications for deploying them on high performance, heterogeneous computing (HPC) infrastructure. This puts their use beyond the reach of many water resources modelers. In this paper, we describe new advances in component-based modeling using a framework called HydroCouple. This framework largely adopts the Open Modelling Interface (OpenMI) 2.0 interface definitions, but demonstrates important advances for water resources modeling. HydroCouple explicitly defines standard and widely used geospatial data formats and provides interface definitions to support simulations on HPC infrastructure. In this paper, we illustrate how these advances can be used to develop efficient model components through a coupled urban stormwater modeling exercise.

[1] Co-authored by Caleb Amoa Buahin and Jeffery S. Horsburgh

4.1 Introduction

The goal of integrated assessment in environmental and natural resources management is to provide information within a decision making context that brings together a broader set of domains, methods, styles of study, and/or degrees of certainty than would typically characterize a study of the same issue within the bounds of a single research discipline (Parson, 1995; Laniak et al., 2013). In order to make the complexity surrounding integrated assessment studies more tractable, a need has also arisen to integrate computer models from diverse fields so that scientists can conduct more holistic assessments. In particular, for water resources specialists, the need for model integration arises frequently because, although many individual hydrologic processes have existing mathematical models that are able simulate them under a given set of circumstances, there is rarely a single model that can simulate all of them at the different scales and complexities desired while accounting for feedbacks between the various sub-processes for integrated assessment studies (Beven et al., 1980; Argent et al., 1999). Selecting a particular hydrologic model requires a consideration of the specific management challenges of concern, the spatial and temporal scales of interest, model input data availability, and computing requirements, among other considerations (Beven et al., 1980; Leavesley et al., 2002; Argent, 2004; Voinov and Shugart, 2013; Chowdhury and Eslamian, 2015; Clark et al., 2015). When a single model cannot meet the needs of a modeling study, it is common for modelers to couple elements of multiple models together to form a more holistic or accurate representation of a water system.

Although model developers in the earth systems and environmental modeling field have used several approaches to couple models for their integrated modeling efforts, the component-based modeling approach is increasingly receiving more attention. This is because, in contrast to monolithic approaches where models are compiled into a single code base or executable unit, component-based model development promises improved flexibility in the selection and re-use of different models as well as more maintainable and extensible models (Fröhlich and Franz, 1999; Szyperski, 2002). The component-based modeling paradigm involves the provision of interface definitions describing standard data structures and functions that models must implement so that they can be deployed independently to exchange information at runtime with other models. Model developers across diverse fields who adopt these interface definitions can develop models that can be coupled with other models to simulate complex earth and environmental systems. Component-based modeling provides a natural avenue for experimenting with different model formulations since model components can be removed and added to a composition in a "*plug-and-play"* fashion.

Several component-based modeling frameworks and interface standards with varying degrees of complexity and application domains have been developed over the years. These include the Earth Systems Modeling Framework (ESMF, Hill et al., 2004), Community Surface Dynamics Modeling System (CSDMS, Peckham et al., 2013), the Object Modeling System (OMS, David et al., 2002), and others. In the water resources modeling arena, the Open Modeling Interface (OpenMI, Moore and Tindall, 2005) definitions have been tested and used extensively (e.g., Smolders et al., 2008; Castronova and Goodall, 2009; Goodall et al., 2011; Buahin and Horsburgh, 2015). The appeal of OpenMI revolves around the fact that instead of providing a framework that has an accompanying large and complicated software stack that is beyond the expertise of many water resources modelers, the OpenMI developers provide a set of standardized, programming language agnostic interface definitions that can be adopted to develop model components that can communicate with each other directly at runtime. The OpenMI interface definitions are object oriented, with clear and well-defined inheritance relationships. Another major advantage is that the latest OpenMI 2.0 version has been adopted as an Open Geospatial Consortium (OGC) standard (Vanecek and Moore, 2014), which means that it has been reviewed and vetted by a large community of modelers.

Despite these attractive features, our experience using OpenMI revealed a few areas where advancements to the interface definitions would be useful, especially for water resources modeling applications. First, while the OpenMI interface definitions are programming language agnostic, the example interface definitions as well as software development kits (SDK) provided by the OpenMI developers use the C# and Java programming languages. These languages are both interpreted languages, where code is compiled into an intermediate bytecode before being translated into the native instructions of a target machine to be executed using a virtual machine software infrastructure (i.e., Common Language Runtime (CLR) for C# and Java Virtual Machine (JVM) for Java). Additionally, C# has traditionally been restricted to computers that run Windows operating systems.

Conversely, many legacy model codes used in the water resources modeling field, and in the earth systems and environmental modeling field more generally, have been developed using programming languages like Fortran, C, and C++. These programming languages are employed for computational models because they are compiled directly into native instructions for a target machine and, therefore, generally have lower memory footprints, faster performance, and can be compiled on many operating systems. To convert these legacy codes into components that can be coupled loosely to other models, one needs to resolve the programming language mismatch between the interface definitions and the computational codes of these legacy models. Though there are ways to bridge this programming language mismatch (e.g., using Platform Invocation Service for C# and Java Native Interface for Java), the costs of marshalling data across this language divide can lead to increased memory usage and increased simulation times. We encountered and quantified these costs in a previous study where we converted the Environmental Protection Agency's Stormwater Management Model (SWMM), which is written using the C programming language into an OpenMI compliant component using the C# OpenMI interface definitions for a spatial domain decomposition urban stormwater coupling exercise (Buahin and Horsburgh, 2015).

Second, while the OpenMI specification provides interface definitions for representing geometric primitives (e.g., points, lines, and polygons) and their associated time varying data, these definitions lack some of the more common geospatial dataset formats used by water resources modelers (e.g., meshes, vector datasets, rasters, etc.). Also, the geospatial interface definitions provided by the OpenMI specification lack the topological relationship information that is important for many water resources modeling applications. For instance, a hydrologist simulating flows in a river network will need to know which upstream tributaries flow into any selected river reach. These types of topological relationships are not currently supported by the OpenMI 2.0 standard.

Finally, like other earth systems and environmental modelers, hydrologic modelers often embark on experimental simulations where the same model is executed multiple times with varied inputs (e.g., optimization, uncertainty assessment, calibration, etc.). These types of simulations fall into the so called "embarrassingly parallel" class of simulations and benefit from using high-performance computing (HPC) resources. Many research institutions provide access to computing clusters comprised of heterogeneous hardware configurations of multi-core Central Processing Units (CPU) as well as graphical processing units (GPU) and Many Integrated Cores (MIC) architecture accelerators that can be used for more efficient computations. However, the current OpenMI standard provides little direction on how to take advantage of these increasingly ubiquitous HPC infrastructures for more efficient simulations.

The contribution of this paper is the presentation of a set of component-based modeling interface definitions called HydroCouple and its associated SDK and coupled model compositions tools. HydroCouple uses the OpenMI 2.0 interface definitions as its foundation, but advances new interface definitions to address the challenges enumerated in the preceding paragraphs and others. We chose to build from OpenMI because it has been tested and used within the water resources modeling domain and because it has recently been advanced as an OGC standard.

We describe the various changes and new interface definitions we have implemented with HydroCouple and illustrate how these advancements can be used to convert existing, legacy codes into model components and develop new ones from scratch. We also use a coupled urban stormwater modeling exercise example to illustrate how these advancements can help usher more water resources modelers into the HPC realm so that they can embark on more efficient simulations. This exercise involved coupling a one-dimensional (1D) hydraulic model developed from the Environmental Protection Agency (EPA) stormwater management model (SWMM) and a two-dimensional (2D) hydraulic model that we developed called the Finite Volume Hydrologic Model (FVHM).

4.2 Design of Component-Based Modeling Frameworks

The component-based modeling approach and its precursor, the component-based software development approach, have their origins in the object-oriented programming approach with its notions of re-use through encapsulation, inheritance, and polymorphism. Components are typically paired with software frameworks, and they serve to extend the capabilities of frameworks, while frameworks provide an environment for executing components (Fröhlich and Franz, 1999). This concept extends to model components and modeling frameworks in earth systems and environmental modeling. For instance, CSDMS components are executed within the Common Component Architecture Fast Framework Example In Need of Everything (CCAFFEINE, Allan *et al.*, 2002). The definition of standard interfaces forms the basis of interaction between components and frameworks and between components themselves by describing the assumptions they make about each other (Fröhlich and Franz, 1999). The design of OpenMI deviates from other approaches for developing component-based models by forgoing the pairing of interface definitions for components with a software framework. Instead, it prescribes interfaces that let components communicate directly with each other independent of a framework. This design choice was made to give more flexibility to component developers to optimize the data exchange process between model components.

The interface definitions for components are specified in a number of ways for different frameworks depending on the programming languages supported by the framework. Programming languages that were developed primarily for object oriented programming like C++, C#, and Java have formal ways of specifying interfaces so that they can be inherited and implemented with details of their functioning to create framework compliant components. For example, C++ interfaces are specified in header files as classes with only pure virtual functions. The OpenMI interfaces for C# and Java are specified this way. On the other hand, for component-based modeling frameworks that support languages like C and Fortran (e.g., ESMF and CSDMS), models are required to register pointers to their standard functions with the framework. These standard function pointers are then stored in virtual function tables that can be accessed by other components to achieve the interfacing functionality. While the OMS framework was written using Java, which is an object-oriented programming language, the OMS interfaces are specified by marking classes, functions, and fields of a component with standardized java annotations (e.g., @In, @Out, @Execute, etc.), which serve as a form of syntactic metadata. Through Java's reflection capabilities, annotated classes, fields, and functions can be accessed and invoked at runtime.

The typical core interfaces defined for models in component-based modeling frameworks are the *initialize*, *run*, and *finalize* functions, or the so called IRF (Syvitski *et al.*, 2011). The *initialize* interface function is implemented to instantiate the resources and inputs needed by a model for a simulation. The *run* interface is responsible for performing the underlying computations of a model (e.g., performing a time-step). The *finalize* interface is implemented to dispose of the computational resources used by the model (e.g., closing output file streams, de-allocating memory, etc.). In addition to these core interfaces, models also specify interfaces for the types of inputs and outputs that can be consumed and shared with other components respectively. CSDMS attempts to standardize these elements that are shared by various component-based modeling frameworks by providing a core set of interface definitions called the Basic Modeling Interface (BMI) definitions (Peckham *et al.,* 2013).

In the following sections, we discuss three important areas in the design of existing component-based modeling frameworks and standards used in the earth systems and environmental modeling arena where we identified opportunities for improvements to be made. We focus on their support for geo-temporal datasets, options for data exchange workflows between components at runtime, and support for simulations on HPC infrastructure. Given that OpenMI was the base from which we built HydroCouple, we discuss how OpenMI is designed with respect to these three areas and contrast it with the design of other component-based modeling frameworks. Limitations in these three areas are real impediments to using OpenMI and, more generally, component based modeling in practice because important geospatial data structures used by many models are not currently supported, data exchange workflows are not easily implemented or customized, and HPC simulations are either complex or not supported at all by existing frameworks.

4.2.1 Definition of Geo-Temporal Data Structures

The types of inputs that can be consumed and the outputs that can be supplied by models in a component-based modeling framework are typically organized as multi-dimensional arrays of data that can be accessed using indexes along their respective dimensions. These inputs and outputs are often further abstracted into domain specific types for many component-based modeling frameworks. For instance, OpenMI provides a time-space input/output specialization that associates geometric primitives including points, polylines, polygons, and polyhedra with time varying data.

In water resources modeling applications, these features are often used for delineating a model's spatial domain and prescribing boundary or input data for models - e.g., polygons for watershed boundaries, polylines for alignments of rivers, river cross-sections, etc. Missing from the OpenMI definitions is the topological information that provides the spatial relationships between adjacent geometries. Yet, this information is critical for many applications. For example, although the individual cells of a two-dimensional computational grid used for a hydrodynamic model of a reservoir may be represented by a list of polygons, the adjacency information between cells that is needed to numerically approximate spatial gradients of variables are missing. On the other hand, the CSDMS suite of tools for component-based modeling, which were developed for ice, terrestrial, coastal, and marine applications as well as ESMF, which was developed for global weather and climate predictions, focus on providing interfaces for gridded datasets, including logically rectangular, unstructured, and curvilinear grids in one, two, and three dimensions. Explicit support for Geographic Information System (GIS) geospatial data types like those provided in OpenMI are, however, not supported.

4.2.2 Data Exchange Workflows

The model execution and data exchange workflow between components in a component-based model composition is handled in a variety of ways for different frameworks. ESMF requires a modeler to write a driver program called an "*AppDriver*" that contains the "main" routine for an ESMF application (Collins *et al.*, 2005). This "*AppDriver*" needs to be compiled for each application and is responsible for directing the time-stepping and the data exchange between components. The drawback with this approach is that a new driver needs to be written and compiled for each composition. Therefore, a modeler will have to be a programmer with the know-how and compilation tools to compile the application, putting it out of reach for many water resources modelers who are typically not expert programmers.

In contrast to ESMF, OMS provides a way for modelers to direct model execution and data exchange between components externally. This is accomplished by letting users write the business rules for a simulation using a OMS prescribed domain specific language (DSL) specified in an external file (David et al., 2013). In contrast to a general-purpose programming language, a DSL is a relatively simple specification language dedicated to a particular problem domain, a particular problem representation approach, and/or a particular solution technique (Deursen and Financial, 1997; Deursen et al., 2000; David et al., 2013). The benefit of this setup is that, model users are able to direct the data exchange between components without recompiling the entire code or being restricted to the in-built workflows provided by a component-based modeling framework.

The primary model execution and data exchange mechanism in OpenMI is based on the pull-based pipe and filter architecture (Buschmann, 1996). With this method, components exchange memory-based data directly using a “request and reply” mechanism (Gregersen et al., 2007). The most downstream component in a composition is designated as the controller/trigger for an entire simulation where it requests the data it needs from upstream components that it is coupled to and waits for the data it requests to be returned before proceeding with its computations. Upstream components issue their own requests to their respective upstream components in a cascading fashion and wait to receive the data they requested before performing their computations as illustrated in Figure 4.1. Data exchange between an input (defined by the *IBaseInput* interface) and an output (defined by the *IBaseOutput* interface) can be mediated by an adapter (defined by the *IBaseAdaptedOutput* interface) that performs the necessary data transformations (e.g., temporal interpolation, spatial interpolation, unit conversion, etc.) that are needed to supply the correct requested data from one component to another. Contextual adaptors are generated by a factory interface definition called an *IBaseAdaptedOutputFactory* that uses the input and output that are to be mediated as query variables to generate appropriate adaptors.

While the “request and reply” data exchange approach works for many applications, it does not work for those compositions that have two or more downstream components as illustrated in Figure 4.2. In Figure 4.2, components D and E are not executed since they are not involved in the request and reply chain of the trigger component, A. The OpenMI developers recognized that the pull-driven data exchange approach might be too restrictive for certain applications. Therefore, they suggested a loop driven data exchange workflow where the coupled system will loop over all components to let them check if they need to take action before proceeding (Moore, 2010). However, the implementation for this loop driven approach were not provided as part of OpenMI 2.0.

4.2.3 Support for Simulations on HPC Infrastructure

ESMF was designed to support both data and task parallelism on HPC infrastructure by making ESMF components the very units of parallel execution to aid model developers in writing highly efficient and scalable codes (Collins et al., 2005). A virtual machine (VM) approach is adopted to abstract away the details underlying model execution in HPC environments using the Message Passing Interface (MPI) standard. Each component in an ESMF composition is assigned a single VM comprised of one or more persistent execution threads (PET), where each PET is equivalent to a single MPI process. A component may allocate its PETs to child components. Additionally, a developer can employ fined grained, shared memory parallelism within a PET using OpenMP or other multi-threading approaches. Inter- and intra- VM communications are handled by the framework in a fashion similar to MPI where data must be provided as raw, language specific, one-dimensional, contiguous arrays.

OMS adopts a similar approach to ESMF to achieve parallelism by making components themselves the units of parallel execution. However, unlike ESMF, OMS only supports shared memory parallelism applications using multi-threading, where each model component is executed in its own separate thread. On the other hand, OpenMI provides little direction on how to take advantage of HPC resources for more efficient simulations. For scenarios where a downstream component requests input data from one or more upstream components, an OpenMI component may make these requests using parallel threads in a multi-threading environment for efficiency. However, this approach only scales up to the number of upstream components that supply data to a single component.

To the best of our knowledge, existing component-based modeling frameworks lack support for automated parallel simulations for those experimental evaluations that are “embarrassingly parallel” involving executing the same coupled model instances repeatedly with varied input parameters - e.g., automated parameter estimation, uncertainty assessment, optimization, ensemble simulation, etc. While, these types of applications can be currently undertaken within existing component modeling frameworks, they can either only be configured to run sequentially, require manual intervention, or require writing code outside of the scope of a component-based modeling framework.

4.3 Design of the HydroCouple Framework

To address the challenges discussed in the preceding sections, we developed the HydroCouple component-based modeling interface specifications and associated software. HydroCouple builds on the strengths of OpenMI while advancing new interface definitions to deliver standard geo-temporal dataset interfaces with their associated topological information and providing support for more efficient simulations on HPC infrastructure. Like OpenMI, the HydroCouple interface definitions are language agnostic. However, we implemented the HydroCouple interface definitions using C++ because in addition to the benefits enumerated in the previous sections, it also provides bindings with many programming languages and serves as a good foundation for a framework that seeks to integrate legacy model codes and support models written using different programming languages. For simplicity and ease of learning, the interfaces for HydroCouple have been provided in only four C++ header files with about 1500 lines of code that provide the core interface definitions, a spatial extension, a temporal extension, and a spatio-temporal extension. In the following section, we discuss the key choices that were made in the design of HydroCouple and the motivation behind them.

4.3.1 Types of HydroCouple Components

The core interface definition of the OpenMI standard is the *IBaseLinkableComponent* interface. This interface encapsulates the computational engine of a model and defines the IRF interfaces. It is also responsible for defining the types of inputs a component can consume and the type of outputs a component can supply to other components. In HydroCouple, the *IBaseLinkableComponent* interface has been superseded by the *IComponentInfo* interface definition as the core interface definition. This interface definition allows model developers to provide details about a component including, documentation about the formulations employed in the component, limitations of the component, coupling configurations that can be employed, data transformation adaptors that can be employed, as well as details about the coupling scenarios for which the use of the component is appropriate. Other details that can be specified as part of this interface and can be used model users to seek further assistance include, the name of the developer of the component, contact information for the developer, classification of the component, and component version. These details are currently missing for components developed for many frameworks leading to poor understanding about their capabilities and their proper usage in component-based modeling applications.

In contrast to OpenMI, which only allows a single component type, the *IComponentInfo* interface creates instances of three types of specialized components. These specializations include the *IModelComponentInfo*, the *IWorkflowComponentInfo*, and the *IAdaptedOutputFactoryComponentInfo* interfaces as shown in Figure 4.3. The *IModelComponentInfo* is responsible for creating instances of the *IModelComponent* (Figure 4.4) interface, which is equivalent to the OpenMI specification's *IBaseLinkableComponent*. The *IAdaptedOutputFactoryComponentInfo* is responsible for creating instances of a new component type called the *IAdaptedOutputFactoryComponent* (Figure 4.4). This new interface definition can be implemented as an independently deployable component that is equivalent to OpenMI's *IAdaptedOutputFactory* interface definition. However, unlike the OpenMI definition, it is not bound to any particular component and can be reused between different models to generate adaptors for data exchange mediation. Finally, the *IWorkflowComponentInfo* is responsible for generating another new type of component called the *IWorkflowComponent* (Figure 4.4) that is responsible for managing simulations and the data exchange workflow between components. These new component types have been added to HydroCouple to avoid the duplicative work of developing new data exchange workflows and data adaptors for each component or coupled modeling application when using OpenMI and component-based modeling frameworks in general. For example, a time series interpolation adapter could be developed as an independently deployed *IAdaptedOutputFactoryComponent* that can be reused by different model component compositions. This contrasts with the current OpenMI setup where an adaptor is bound to a component current and needs to be implemented for every model component that needs to use it.

4.3.2 HydroCouple Geo-Temporal Data Structures

In order to reduce the cost of converting legacy models into components, it is important that the low-level geospatial data structures that hydrologists widely use are made available in component-based modeling frameworks. To achieve this goal, HydroCouple provides an explicit implementation of the Open Geospatial Consortium’s (OGC) Simple Feature Access (SFA) specification (Herring, 2011). In addition to providing interfaces to describe various types of geometries, the advantage of implementing the OGC SFA is that it defines functions for performing topological queries (e.g., checking for intersection between features, checking if one feature touches another, etc.) and geospatial operations (e.g., unions, buffering, symmetric difference, etc.) that are useful in water resources modeling applications. For instance, an intersection operation can be used between gridded precipitation output from a weather forecast model and the boundary of a watershed in a hydrologic model to estimate the fraction of each grid cell that contributes precipitation flux to that watershed.

HydroCouple additionally prescribes interface definitions for various gridded dataset types for hydrodynamic and hydraulic modeling applications as well as representing some other gridded dataset types often utilized in the water resources area. For example, HydroCouple explicitly defines an interface for multi-banded raster datasets that includes the number of raster grid cells, cell size, data type, no data value, etc. This interface can be employed to represent various types of datasets in models, including digital terrain models (DTM), aerial imagery, time varying land use data, etc. In addition to this definition, HydroCouple defines interfaces for cartesian, rectilinear, and curvilinear grids in two and three dimensions for hydraulic and hydrodynamics modeling applications. Time-varying data for these grids may be associated with the nodes, faces/edges, and volume/area of their individual cells. For networks, unstructured meshes, and polyhedra, HydroCouple adopts the quad edge data structure proposed by (Guibas and Stolfi, 1985). With this data structure, a directed edge stores the complete topological information about the resulting polyhedra/network by storing pointers to its left and right face polygons and its origin and destination vertices.

These geo-temporal interface definitions have been provided to abstract away the details of the various types of the same data structures provided by various software libraries. For example, a raster dataset can be stored as a GeoTIFF file or a NetCDF file which are accessed using different software libraries. This has been done so that data can be accessed within HydroCouple using common interfaces, regardless of the underlying file format of the data. A companion software development kit was also developed to enable reading and writing of geo-temporal data to and from files on disk using multiple file formats. Details of the implementation of these interfaces within the software development kit using existing libraries are provided in subsequent sections.

4.3.3 HydroCouple Data Exchange Workflows

Following the OMS approach where the data-exchange workflow can be specified externally, the *IWorkflowComponentInfo* interface definition was introduced in HydroCouple as a way to provide modelers with flexibility in prescribing the data exchange workflow in a component-based application. For example, OpenMI currently supports only the “request and reply” data exchange approach, whereas HydroCouple adds the ability to create independent workflow components that can execute using a loop-driven data-exchange approach as well as others. This interface creates an instance of an *IWorkflowComponent* that is responsible for keeping track of all components involved in a simulation and coordinating their computations and data exchange until the completion of a simulation. Just as the *IModelComponent* interface has the IRF functions, we defined the *IWorkflowComponent* with these same functions to enable initializing the workflow component, updating components associated with the workflow by asking them to perform their computations, and finalizing the components upon completion of a simulation. If the *IWorkflowComponent* interface is not instantiated for a simulation, all components must default to the original OpenMI “request and reply” approach to data exchange. This new interface was added so that modelers can have fine grained control over a simulation beyond the “request and reply” data exchange mechanism provided by OpenMI. Like model components, a suite of independently developed and deployed workflow components may be developed so that model users are able to select and integrate the workflows that are most suitable for their simulations.

4.3.4 HydroCouple on HPC Infrastructure

HydroCouple supports parallelized, experimental simulations by introducing a new interface definition called the *ICloneableModelComponent* interface that inherits from the *IModelComponent* interface. This new interface introduces a *clone* function that must be implemented for a component to make a copy of itself. The *clone* function has been added so that independent copies of a model instance can be made for parallelized simulations. Details of the cloning approach are left up to the model component developer. A parent model component keeps track of all of its child clones, which can be accessed using the *children* function. Linkages with other model components are left up to the caller of the *clone* function. This cloning process may involve making a copy of the parent *ICloneableModelComponent* class and initializing it with the same arguments as the parent while making sure that outputs from the parent and child do not conflict. An example optimization application of this cloning feature is described by Buahin and Horsburgh (2016).

HydroCouple was designed to support simulations on HPC infrastructure by supporting parallel simulations on both shared and distributed memory systems as well as providing support for simulations that use general purpose computing on graphics processing units (GPGPU). Support for distributed memory parallel computing was designed using the Message Passing Interface (MPI) standard, while the GPGPU parallelism was designed to support the Open Computing Language (OpenCL) and NVidia’s CUDA framework.

In designing HydroCouple’s support for simulations on HPC platforms, we followed the original OpenMI design choice of limiting the role of the framework in mediating the data exchange between components. Along this line, HydroCouple applications adopt MPI for coarse grained parallelism, where all components involved in a coupled modeling application are initialized and coupled directly on the MPI task/process with rank 0 (i.e., the master MPI Process 1) as illustrated in Figure. For components that support MPI, a user may partition the remaining MPI processes to them to ensure the computational load is balanced (Figure 4.5). This is done by specifying the MPI ranks available to each component in a coupled model application configuration file. Frameworks like ESMF automatically map computational resources into virtual machines that can be allocated to components for model execution on HPC resources. While convenient in abstracting away the complexity of computational resources available, it elevates the role the framework plays in mediating the data exchange between components. Additionally, getting up to speed with the large software code stack and its accompanying complexity is beyond the expertise of many water resources model developers. HydroCouple uses relatively few lines of code and distributes HPC resources directly by dividing available MPI ranks among components. Each component can group its allocated MPI ranks into an MPI communicator that can be shared among the component and its children. This approach can be more transparent for users of shared HPC infrastructure that use job scheduling software like Slurm and reduces the role of the framework.

Developers of components need not necessarily conduct their computations on the master MPI process. Components initiated on the master MPI process could be developed as proxy-stubs that serve as pathways for communicating results computed on child components on worker MPI processors. For example, in Figure 4.5, the role of Component A, which is initialized on the master MPI process to communicate directly with Components B and C, may only involve collating computed results from its worker components initialized on MPI processes 2, 3, and 4 and sharing with other components. Once initialized, data sharing between components on different MPI processes can be conducted using standard MPI calls. Components on the master MPI processes are responsible for issuing messages to components on the worker processes to dispose themselves upon completion of a simulation. Fine grained parallelism may be employed over the CPU cores allocated to each MPI process using shared memory parallelism application programming interfaces (APIs) like OpenMP.

Like MPI, GPGPU frameworks abstract away the complexity underlying computing resources from users and present a single, virtual interface for accessing the hardware. To ensure that GPGPU resources are distributed efficiently between components, HydroCouple lets users prescribe the GPGPU device as well as the maximum number of CUDA blocks or OpenCL work groups each component can use on each MPI process. This gives users flexibility to partition jobs to GPGPU devices in a way that is tailored to the particular hardware layout on an HPC system. This partitioning of computing resources can be accomplished using job scheduling software like the Slurm Workload Manager that is widely using on many high-performance computing centers' systems.

4.3.5 HydroCouple Composer and Software Development Kit

In order to facilitate the development of components for HydroCouple, we have developed a software development kit (SDK) that implements many of the core interfaces needed to develop a component as a software class that has initialization arguments, exchangeable inputs, exchangeable outputs, and all of the methods required to do initialize the component by reading and validating its initialization arguments, and describe the variables exchanged by a component's inputs and outputs. This enables model developers to focus on the computational portions of their code, which are placed within the correct locations inside the HydroCouple SDK classes. The core interfaces that have been implemented as classes include those for describing and identifying components and their inputs and outputs, the spatio temporal domains of these inputs and outputs, the variable types consumed and supplied by these input and outputs, and the units of these variables.

The SDK uses the Geospatial Data Abstraction Library (GDAL; Warmerdam, 2008) extensively to provide support for reading various geospatial vector data formats (e.g., shapefiles, GeoJSON, GML, CAD, KML, etc.) into objects defined by OGC's SFA that can then be accessed programmatically by model components. The SDK also enables writing these objects to files on disk (e.g., for exporting model output). The HydroCouple SDK also uses the GDAL library to provide similar support for various raster dataset formats including the GeoTIFF, GRIB, SQLite, JPEG, and HDF5 formats. Work is also underway to implement the UGRID version 1.0 (http://ugrid-conventions.github.io/ugrid-conventions) convention, which is a proposal for storing unstructured mesh data in standard format using the Unidata Network Common Data Form (NetCDF) file using the Climate and Forecast (CF) metadata convention (Eaton et al., 2011) as a starting point.

In addition to the SDK, we developed the HydroCouple Composer graphical user interface (GUI) to provide model developers with a visual environment for composing coupled model configurations (Figure 4.6). The HydroCouple Composer software displays all available model component libraries and allows users to drag and drop them onto to a graphical palette. Coupled model compositions are specified through a configuration file that lists the model components and their connection nodes as well as the workflow and adaptor components that are utilized. Each model component can be initialized using an initialization file that contains parameters specific to the component. After a component is initialized, compatible outputs and inputs from other components can be coupled interactively. Compatibility for coupling is determined by a function called *canConsume* that is called when a model user tries to create a link between and input and an output. This function returns true or false indicating whether the coupling is valid based on business rules defined by the component developers. Selecting an existing connection between an input and output displays the available contextually relevant adaptors that can be inserted to mediate data exchange.

HydroCouple Composer is responsible for partitioning computing resources to model components based on a user's specifications in the configuration file and launches simulations. The HydroCouple Composer also monitors the progress of simulations and displays them to a user, logs messages from components, and provides rudimentary visualization capabilities for the geo-temporal inputs and outputs of model simulations. Finally, when the executable of the HydroCouple composer is launched using predefined command line arguments, it doubles as a command line interface for launching simulations on HPC resources.

4.4 Case Study: Coupling a 1D and a 2D Hydraulic Model Using HydroCouple

In order to illustrate how HydroCouple's new interfaces facilitate the development of components and more efficient simulations, we applied them to couple a 1D hydraulic model component that simulates flow through the pipes, culverts, inlets, outfalls, and other urban stormwater infrastructure with a 2D hydraulic model component that simulates flow in rivers, canals, and overland areas. The 1D hydraulic model component was developed using the EPA's SWMM model, while the 2D model component is a new formulation we developed called the Finite Volume Hydrologic Model (FVHM). These two models were specifically chosen to illustrate: 1) how the HydroCouple interface definitions and SDK can be used to convert legacy models into components; 2) how to develop new component from scratch; 3) how to handle the potential coupling configurations across models having differing dimensionality (i.e., 1D versus 2D); and 4) demonstrate how coupled model components can be executed in parallel on HPC environments for more efficient simulations.

This particular type of 1D-2D hydraulic model coupling is widely implemented in the water resources modeling field because the tradeoffs between 1D and 2D hydraulic models complement each other. 1D models are appropriate for simulating flows accurately and efficiently in channels, pipes, and other conduits with well-defined shapes. Although 1D hydraulic models are generally more computationally efficient, they are unable to accurately simulate lateral movement of flood waves into the floodplain, and they incorporate topography and bathymetry using cross-section profiles at various sections along the length of a river/pipe whose placements are relatively subjective (Samuels, 1990; Hunter et al., 2007). On the other hand, 2D hydraulic models are more suitable for simulating landscape processes and overland flows, albeit at a generally higher computational expense. Many urban hydrologic modeling scenarios (e.g., stormwater runoff, flooding, design of green infrastructure, and assessment of stormwater best management practices) require accurate representations of both the drainage network and the urban landscape, making a 1D-2D hydraulic model coupling ideal for simulating these scenarios. The 1D-2D coupled model discussed in this manuscript was developed for a sub-catchment in the City of Logan, Utah. To illustrate the benefits of using HPC resources, we evaluated the simulation time speed up as more computational cores were allocated to the coupled model.

4.4.1 Study Area

The City of Logan’s stormwater conveyance system has its foundations in agricultural canals developed at the founding of the City to divert irrigation water from the Logan River for farming. As the Logan River flows westwards through Cache Valley, it is diverted at various locations along its length into these unlined canals that flow northward. These same canals serve as the primary conduits for stormwater conveyance, with many of the City’s stormwater outfalls piped directly into them. The boundary for the 5.81 km^2 area we used for our modeling study encompasses the area that drains into one such canal called the Northwest Field canal (Figure4. 7).

4.4.2 SWMM Model Development

The SWMM code solves the 1D dynamic, diffusive, or kinematic wave equations for flows and water surface elevations over a network of conduits (i.e., pipes, canals, and rivers) and sub-catchments connected together at their endpoints by nodes (i.e., junctions, outfalls, storage units, and flow dividers) (Rossman, 2006).

We developed the SWMM HydroCouple Component from the EPA SWMM source code by modifying the code to expose those boundary data that are needed for coupling including the inflows, outflows, and water surface elevations at the inlets and outlets of the stormwater conveyance system. In developing the SWMM HydroCouple component, we employed the HydroCouple network interface definition that utilizes to represent the network of SWMM nodes connected together by conduits. The benefit of this is that the inflows and outflows as well as water surface elevation data that the SWMM component supplies to components coupled to it provides the topological information that is needed to traverse the entire stormwater conveyance system.

The SWMM model code has been parallelized in many sections to improve performance using OpenMP as described by Burger et al. (2014). The parallelism introduced in SWMM is a shared memory type of parallelism (i.e., it uses a single MPI process but scales with number of cores allocated to the MPI process). This feature served as the basis for our investigation of the performance of the coupled model as more CPU resources were added.

We developed the SWMM model instance using detailed survey data of sizes for stormwater pipes, inlets, and outfalls provided in a geographic information systems (GIS) format by Logan City. The conduit diameters ranged from 0.30 to 1.38 m, with lengths ranging from 0.5 to 390 m. This dataset resulted in a SWMM model with 1769 conduits, 2093 junction inlets, and 138 outfalls. We executed the SWMM model using SWMM's adaptive timestep option with a minimum timestep of 0.01 seconds and a maximum time step of 5 seconds. A maximum number of 20 iterations was selected for each time-step.

4.4.3 FVHM Model Development

FVHM was developed to solve the shallow water equations over an unstructured triangular irregular network (TIN) mesh using an implicit finite volume method. Details for the formulations and hydrologic process representations used are provided in Appendix A. In the context of this case study, FVHM was developed to simply route flows in riverine and overland areas without many of the hydrologic processes that are typically represented in hydrologic models (e.g., infiltration, evapotranspiration, etc.). However, FVHM fully exposes the geospatial data structures needed to demonstrate how different configurations can be used to couple 1D and 2D models and run them on HPC. We purposefully developed FVHM for this case study to focus on the model coupling data structures and configurations without the complexity introduced by representing many hydrologic processes in the 2D model component. Because the hydrologic processes represented in the 2D model are independent of the data structures and coupling configuration, the techniques we demonstrate here are generalizable to 2D models that include more detailed hydrologic process representations. Given that our 2D model does not represent processes like infiltration, the model configuration presented here represents a worst-case scenario simulation and is a useful exercise for evaluating the performance of the stormwater infrastructure under a high intensity rainfall event.

In developing the FVHM model, we directly utilized the data structures provided as part of the HydroCouple interface definitions. For example, the computational mesh used in FVHM adopts a TIN interface definition with its associated quad-edge data structure. For prescribing boundary conditions, the geometry interfaces prescribe by the OGC SFA specification were adopted. For instance, polygons were used to demarcate the area where precipitation inputs apply. Polylines were used to define the mesh edge boundaries where inflows and outflow apply. The implementations of these interfaces and the file input and output implementations provided by the HydroCouple SDK simplified the model development process by allowing us to focus on the computational parts of the code we were actually interested in.

We were also able to use the capabilities of HydroCouple to enable parallelism in executing the model. FVHM uses both fine and coarse-grained parallelism in its code. Fine grained parallelism using OpenMP is employed in several areas in looping over each of the computational cells to calculate spatial gradients of water surface elevations and velocities, friction, and to apply boundary conditions. Since FVHM uses an implicit time marching scheme, systems of linear equations need to be solved at each time step to compute velocities and water surface elevations for each cell. FVHM uses the hypre software library (Falgout and Yang, 2002), which solves large, sparse linear systems of equations on massively parallel computers. FVHM partitions the system of equations it solves at every time step into smaller chunks to be solved in different MPI processes using the hypre library.

For the FVHM model instance, we developed its computational mesh using sub-meter, high resolution light detection and ranging (LIDAR) data collected in 2005. This dataset was supplemented with the 10-m elevation data from the United States Geological Survey's (USGS) 3D Elevation Program. The mesh contained 44861 cells with sizes ranging from 0.1 to 18900 m^2. This range of cell areas was the result of refining the model mesh along the canal where we were interested in evaluating in more detail and coarsening the mesh in the upstream overland areas where we were only interested in estimating runoff. For the boundary conditions, a 30-minute resolution, time varying precipitation time series for the nearest rain gauge was developed using the 25-year, 24-hour design storm totaling 61.2 mm and the Natural Resources Conservation Service (NRCS) Type-II (Cronshey, 1986) rainfall distribution curve. This storm is the prescribed storm for designing urban stormwater infrastructure in Logan. Thus, is a useful test for the coupled model. For diversion flow from the Logan River into the Northwest Field canal, we applied the maximum legally allowable irrigation diversion of 1.351 m^3/s for the entire duration of the storm to evaluate worst cast inundation conditions (e.g., an intense storm during a time where the canal was full of irrigation water). We executed the FVHM model using the adaptive time step option with a minimum timestep of 0.01

seconds and a maximum time step of 5 seconds. A maximum number of iterations of 200 was specified for the for each time step.

4.4.4 1D-2D Model Coupling Configurations

In coupling SWMM and FVHM, we adopted different coupling configurations depending on the relationship between the water levels in the coupling cell of the FVHM model, water level in the coupling inlet or outfall/outlet of the SWMM model, ground surface elevation at the coupling interface, and the invert elevations of coupling inlets or outfalls/outlets in the SWMM model (Figure 8). For all cases, we adopted a bi-directional exchange of boundary condition data at the coupling node interfaces between the two models. This bi-directional data exchange proceeds by passing the water surface elevation from the FVHM model to the SWMM model and then passing outflows from the SWMM model to the FVHM model.

For the case where the water surface elevation Z_m of an inlet in the 1D model is less than the bottom elevation of its overlying cell in the 2D model Z_s and the water depth in the 2D cell is greater than A_{in}/w (Figure 10a), the free weir equation (Equation 1) was used to estimate discharge into the inlet:

$$Q = c_w w\sqrt{2g}(Z_w - Z_s)^{\frac{3}{2}} \tag{1}$$

where Q is the discharge into the inlet, c_w is the weir discharge coefficient, *w* is the weir crest width, g is acceleration due to gravity, Z_w is the water surface elevation in the overlying 2D cell, and A_{in} is area of the inlet opening. If the water depth in the 2D cell is greater than A_{in}/w (Figure 4.10b), the orifice equation (Equation 2) is used:

$$Q = c_O A_{in}\sqrt{2g}(Z_w - Z_s)^{\frac{1}{2}} \tag{2}$$

where c_O is the orifice discharge coefficient. The calculated discharges are added as sinks in the continuity equation of the 2D cell and as sources to the inlet of the 1D model.

For the case where an inlet node of the 1D model is completely submerged by the water in the 2D model (Figure 10c), the water level in the corresponding 2D cell is set as the water level in the receiving inlet of the SWMM model. The resulting surcharge or inflow values for the inlet are then applied as a sink/source term in the continuity equation of the 2D cell.

4.4.5 Performance of Various HPC Configurations

We executed the coupled model on up to 10 HPC nodes/MPI processes to evaluate the benefits of devoting more computing resources. Figure 4.9 shows maximum inundation depths for each cell computed for the entire duration of the simulation.

The results of the experiment comparing the relative speed up as the number of MPI processes are increased to the optimal, linear speed up desired are shown in Figure 4.10. Despite the rudimentary parallelism introduced into the FVHM model, the results show up to a 5 times speedup with 10 HPC nodes. Different models and different coupling configurations may achieve different levels of speedup, but this demonstration illustrates the capabilities of HydroCouple to enable simulations on HPC.

4.5 Discussion and Conclusions

The proliferation of many component-based modeling frameworks has made it difficult to fully achieve their promises of reusability and ability to conduct more holistic modeling evaluations. In this paper, we describe a set of interface definitions called

HydroCouple and associated software tools that build on the strengths of the OpenMI 2.0 standard. HydroCouple advances interface definitions to better facilitate water resources modeling applications and data structures that have heretofore been missing in existing component-based frameworks. These interfaces include new topologically aware geospatial data structures based on widely accepted standards like the OGC SFA specification, customizable data exchange workflows, and support for simulations on HPC infrastructure.

While the HydroCouple interface definitions are programming language agnostic, we have implement the interface definitions using C++ to ensure that the code can be compiled on most operating systems. Additionally, the C++ HydroCouple implementation provides a way to avoid the data marshalling costs that often arise when legacy models are wrapped for component-based modeling frameworks using interpreted languages. C++ provides direct memory access to data for models developed using languages like C and Fortran, which have traditionally been used for model development because of their efficiency. Existing OpenMI components developed by wrapping legacy models can be ported to HydroCouple components with minimal effort since HydroCouple adopts most of the OpenMI data structures and concepts. Additionally, HydroCouple provides explicit implementations of standard geospatial dataset formats and associated topological information that are missing from existing component-based modeling frameworks. These data structures are widely employed for delineating model domains and prescribing boundary conditions in many water resources modeling applications.

Unlike their atmospheric modeling counterparts who have had a long history of executing their models on HPC infrastructure, water resources modeling practitioners have traditionally executed their models on single desktop machines. There is, however, an increasing recognition that HPC is needed to tackle challenging problems such as simulating the interaction of land surface hydrologic processes with the atmosphere at the global scale, evaluation of different model structures, and uncertainty assessment. HydroCouple supports this direction by prescribing interface definitions that allow users to partition CPU and GPGPU computing resources among components for their simulations. The 1D-2D coupled hydraulic modeling exercise described illustrates the performance benefits to be gained especially in the hydrologic modeling community by moving into the HPC arena. This application was developed to test the model code and also serve as an example that can be used to develop new model components for HydroCouple. The model has been shared in the GitHub repository (https://github.com/hydrocouple/fvhmcomponent/examples).

The SDK and HydroCouple Composer model coupling GUI environment facilitate the development of components that use HydroCouple. The SDK implements the core HydroCouple interfaces as well as geo-temporal datasets and their file input and output operations so that the costs of converting existing models into components and in developing new models are reduced. The HydroCouple Composer GUI allows users to interactively select and couple models, launch simulations, monitor simulations, and visualize results of simulations on a single desktop or on HPC systems.

We have advanced these new HydroCouple interface definitions in hopes that they can be considered for inclusion in future versions of the OpenMI standard. While we have developed HydroCouple based on the lessons we learned using OpenMI so that it addresses the drawbacks we encountered, we understand that it may not completely support the wide array of simulation types undertaken by water resources modelers. We envision the continued improvement of the interfaces prescribed through HydroCouple as a community effort and have, therefore, shared the interface definitions, HydroCouple Composer GUI, as well as all the components we have developed so far in a transparent manner in a publicly accessible source code repository (https://github.com/hydrocouple). Through this repository, users can contribute new HydroCouple components and improvements to the HydroCouple interfaces definitions, its associated software, and available components back to the repository for the benefit of the larger component-based modeling community.

It is clear that in order to avoid the challenge of over proliferation of component-based modeling frameworks, framework interoperability needs to be a priority among component-based modeling practitioners. Many component-based modeling frameworks share common characteristics (e.g., the specification of IRF interfaces). These common characteristics can be used as leverage to create components that are able use different frameworks. In furtherance of this goal, work is ongoing to create a wrapper component that automatically wraps components that implement the CSDMS BMI interface definitions so that they can be coupled to other models within the HydroCouple framework.

Software Availability

Name of software: The software described in this paper includes the HydroCouple component-based modeling interface definitions, the HydroCouple software development kit (SDK), and the HydroCoupleComposer component coupling graphical user interface (GUI) and command line interface (CLI). The two HydroCouple compliant components we developed for this study include a two-dimensional hydraulic model component called the Finite Volume Hydrologic Model (FVHM) and a component developed from the Environmental Protection Agency's Stormwater Management Model (SWMM) code.

Developers: Caleb A. Buahin

Contact: Caleb A. Buahin; Address: 8200 Old Main Hill, Logan, UT 84322-8200, USA; Email: caleb.buahin@usu.edu

Required hardware and software: Any computer that runs the Windows, Linux, or Macintosh operating system. The performance of the SWMM and FVHM components benefit from compilation with a compiler that supports OpenMP (http://www.openmp.org). FVHM can be executed on distributed memory multiprocessors if compiled using any of the Message Passing Interface (MPI). Compilation of all the software codes enumerated requires the Qt 5.7 framework (http://www.qt.io).

Cost: The HydroCouple interface definitions, software stack, and model components in this manuscript are freely available under the GNU Lesser General Public License

(LGPL) license and can be downloaded from the HydroCouple Github repository (https://github.com/hydrocouple).

Acknowledgments

This research was supported by National Science Foundation EPSCoR Grant IIA 1208732 awarded to Utah State University as part of the State of Utah EPSCoR Research Infrastructure Improvement Award. Any opinions, findings, and conclusions or recommendations expressed are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Allan, B.A., R.C. Armstrong, A.P. Wolfe, J. Ray, D.E. Bernholdt, and J.A. Kohl, 2002. The CCA Core Specification in a Distributed Memory SPMD Framework. Concurrency and Computation: Practice and Experience 14:323–345. DOI: 10.1002/cpe.651

Argent, R.M., 2004. An Overview of Model Integration for Environmental Applications—components, Frameworks and Semantics. Environmental Modelling & Software 19:219–234. DOI:10.1016/S1364-8152(03)00150-6

Argent, R.M., R.B. Grayson, and S.A. Ewing, 1999. Integrated Models for Environmental Management: Issues of Process and Design. Environment International 25:693–699. DOI: 10.1016/S0160-4120(99)00052-5

Beven, K.J., R. Warren, and J. Zaoui, 1980. SHE: Towards a Methodology for Physically-Based Distributed Forecasting in Hydrology. IAHS Publ 129:133–137.

Buahin, C.A. and J.S. Horsburgh, 2015. Evaluating the Simulation Times and Mass Balance Errors of Component-Based Models: An Application of OpenMI 2.0 to an Urban Stormwater System. Environmental Modelling & Software 72:92–109. DOI: 10.1016/j.envsoft.2015.07.003

Buahin, C. and J. Horsburgh, 2016. From OpenMI to HydroCouple: Advancing OpenMI to Support Experimental Simulations and Standard Geospatial Datasets. International Congress on Environmental Modelling and Software. http://scholarsarchive.byu.edu/iemssconference/2016/Stream-A/11.

Burger, G., R. Sitzenfrei, M. Kleidorfer, and W. Rauch, 2014. Parallel Flow Routing in SWMM 5. Environmental Modelling & Software 53:27–34. DOI:10.1016/j.envsoft.2013.11.002

Buschmann, F., 1996. Pattern-Oriented Software Architecture Volume 1: A System of Patterns. Wiley, Chichester ; New York.

Castronova, A.M. and J.L. Goodall, 2009. Comparing Tightly Coupled and Loosely Coupled Paradigms for Modeling Hydrologic Systems. AGU Fall Meeting Abstracts 11:1043.

Chowdhury, R.K. and S. Eslamian, 2015. Climate Change and Hydrologic Modeling. Handbook of Engineering Hydrology: Modeling, Climate Change, and Variability. CRC Press, p. .

Clark, M.P., B. Nijssen, J.D. Lundquist, D. Kavetski, D.E. Rupp, R.A. Woods, J.E. Freer, E.D. Gutmann, A.W. Wood, L.D. Brekke, J.R. Arnold, D.J. Gochis, and R.M.

Rasmussen, 2015. A Unified Approach for Process-Based Hydrologic Modeling: 1. Modeling Concept: A Unified Approach for Process-Based Hydrologic Modeling. Water Resources Research 51:2498–2514. DOI:10.1002/2015WR017198

Collins, N., G. Theurich, C. Deluca, M. Suarez, A. Trayanov, V. Balaji, P. Li, W. Yang, C. Hill, and A. Da Silva, 2005. Design and Implementation of Components in the Earth System Modeling Framework. Int. J. High Perform. Comput. Appl. 19:341–350. DOI:10.1177/1094342005056120

Cronshey, R., 1986. Urban Hydrology for Small Watersheds. US Dept. of Agriculture, Soil Conservation Service, Engineering Division. http://repositories.tdl.org/tamug-ir/handle/1969.3/24438. Accessed 27 Sep 2014.

David, O., J.C. Ascough, W. Lloyd, T.R. Green, K.W. Rojas, G.H. Leavesley, and L.R. Ahuja, 2013. A Software Engineering Perspective on Environmental Modeling Framework Design: The Object Modeling System. Environmental Modelling & Software 39:201–213. DOI:10.1016/j.envsoft.2012.03.006

David, O., S. Markstrom, K. Rojas, L. Ahuja, and I. Schneider, 2002. The Object Modeling System. L. Ahuja, L. Ma, and T. Howell (Editors). Agricultural System Models in Field Research and Technology Transfer. CRC Press, p. . http://www.crcnetbase.com/doi/abs/10.1201/9781420032413.ch15. Accessed 31 Mar 2014.

Deursen, A. van and A. Financial, 1997. Domain-Specific Languages versus Object-Oriented Frameworks: A Financial Engineering Case Study.

Deursen, A.V., P. Klint, and J. Visser, 2000. Domain-Specific Languages: An Annotated Bibliography. Acm Sigplan Notices 35:26–36.

Eaton, B., J. Gregory, B. Drach, K. Taylor, S. Hankin, J. Caron, R. Signell, P. Bentley, G. Rappa, H. Höck, A. Pamment, and M. Juckes, 2011. NetCDF Climate and Forecast (CF) Metadata Conventions Version 1.6.

Falgout, R.D. and U.M. Yang, 2002. Hypre: A Library of High Performance Preconditioners. P. M. A. Sloot, A. G. Hoekstra, C. J. K. Tan, and J. J. Dongarra (Editors). Computational Science — ICCS 2002,Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 632–641.

Fröhlich, P.H. and M. Franz, 1999. Component-Oriented Programming in Object-Oriented Languages. Information and Computer Science, University of California, Irvine. http://www.inf.utfsm.cl/~mcloud/iwi-253/lecturas/tr-ics-1999-49.pdf. Accessed 5 Mar 2014.

Goodall, J.L., B.F. Robinson, and A.M. Castronova, 2011. Modeling Water Resource

Systems Using a Service-Oriented Computing Paradigm. Environmental Modelling & Software 26:573–582. DOI:10.1016/j.envsoft.2010.11.013

Gregersen, J.B., P. J. A. Gijsbers, and S. J. P. Westen, 2007. OpenMI: Open Modelling Interface. Journal of Hydroinformatics 9:175. DOI:10.2166/hydro.2007.023

Guibas, L. and J. Stolfi, 1985. Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi. ACM Transactions on Graphics (TOG) 4:74–123. DOI: 10.1145/282918.282923

Herring, J.R., 2011. OpenGIS® Implementation Standard for Geographic Information - Simple Feature Access - Part 1: Common Architecture.

Hill, C., C. DeLuca, M. Suarez, and A. Da Silva, 2004. The Architecture of the Earth System Modeling Framework. Computing in Science & Engineering 6:18–28. DOI:10.1109/MCISE.2004.1255817

Hunter, N.M., P.D. Bates, M.S. Horritt, and M.D. Wilson, 2007. Simple Spatially-Distributed Models for Predicting Flood Inundation: A Review. Geomorphology 90:208–225. DOI:10.1016/j.geomorph.2006.10.021

Laniak, G.F., G. Olchin, J. Goodall, A. Voinov, M. Hill, P. Glynn, G. Whelan, G. Geller, N. Quinn, M. Blind, S. Peckham, S. Reaney, N. Gaber, R. Kennedy, and A. Hughes, 2013. Integrated Environmental Modeling: A Vision and Roadmap for the Future. Environmental Modelling & Software 39:3–23. DOI:10.1016/j.envsoft.2012.09.006

Leavesley, G.H., S.L. Markstrom, P.J. Restrepo, and R.J. Viger, 2002. A Modular Approach to Addressing Model Design, Scale, and Parameter Estimation Issues in Distributed Hydrological Modelling. Hydrological Processes 16:173–187. DOI: 10.1002/hyp.344

Moore, R. (Editor)., 2010. The OpenMI Document Series: OpenMI Standard 2 Specification. https://publicwiki.deltares.nl/download/attachments/41549981/OpenMI+Standard+2+Specification.pdf. Accessed 2 Jun 2015.

Moore, R.V. and C.I. Tindall, 2005. An Overview of the Open Modelling Interface and Environment (the OpenMI). Environmental Science & Policy 8:279–286. DOI:10.1016/j.envsci.2005.03.009

Parson, E.A., 1995. Integrated Assessment and Environmental Policy Making: In Pursuit of Usefulness. Energy Policy 23:463–475. DOI:10.1016/0301-4215(95)90170-C

Peckham, S.D., E.W.H. Hutton, and B. Norris, 2013. A Component-Based Approach to Integrated Modeling in the Geosciences: The Design of CSDMS. Computers &

Geosciences 53:3–12. DOI:10.1016/j.cageo.2012.04.002

Rossman, L.A., 2006. Storm Water Management Model, Quality Assurance Report: Dynamic Wave Flow Routing. US Environmental Protection Agency, Office of Research and Development, National Research Management Research Laboratory. http://www.hydrolatinamerica.org/jahia/webdav/site/hydrolatinamerica/shared/Manuals/epaswmm5_qa/SWMM5_QA.pdf. Accessed 7 Sep 2014.

Samuels, P.G., 1990. Cross-Section Location in 1-D Models. International Conference on River Flood Hydraulics. Hydraulics Research Limited by Wiley, pp. 339–350.

Smolders, S., I. Neyskens, P. Willems, G. Vaes, and J. Van Assel, 2008. Bi-Directional Sewer-River Linking through the OpenMI Software. http://web.sbe.hw.ac.uk/staffprofiles/bdgsa/11th_International_Conference_on_Urban_Drainage_CD/ICUD08/pdfs/410.pdf. Accessed 1 Aug 2014.

Syvitski, J.P., Hutton, E.W., Peckham, S, and Slingerland, R.L., 2011. CSDMS—A Modeling System to Aid Sedimentary Research. Sedimentary Record 9:4–9.

Szyperski, C., 2002. Component Software: Beyond Object-Oriented Programming. ACM Press.

Vanecek, S. and R. Moore, 2014. OGC® Open Modelling Interface Interface Standard. https://portal.opengeospatial.org/files/?artifact_id=59022. Accessed 2 Jun 2015.

Voinov, A. and H.H. Shugart, 2013. 'Integronsters', Integral and Integrated Modeling. Environmental Modelling & Software 39:149–158. DOI:10.1016/j.envsoft.2012.05.014

Warmerdam, F., 2008. The Geospatial Data Abstraction Library. G. B. Hall and M. Leahy (Editors). Open Source Approaches in Spatial Data Handling. Springer Berlin Heidelberg, pp. 87–104.

Figures

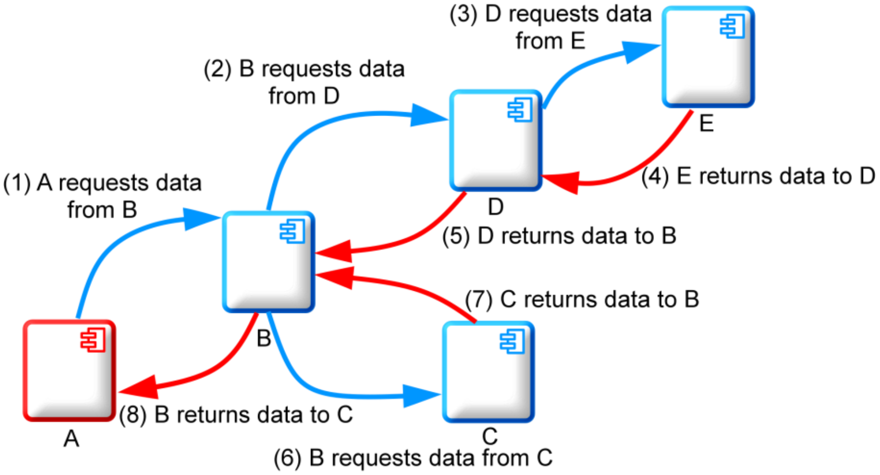

(3) D requests data from E
(2) B requests data from D
E
(4) E returns data to D
(1) A requests data from B
D
(5) D returns data to B
(7) C returns data to B
B
A
(8) B returns data to C
C
(6) B requests data from C

Figure 4.1 OpenMI "request and reply" data exchange mechanism. Component A is the controller/trigger for the simulation.

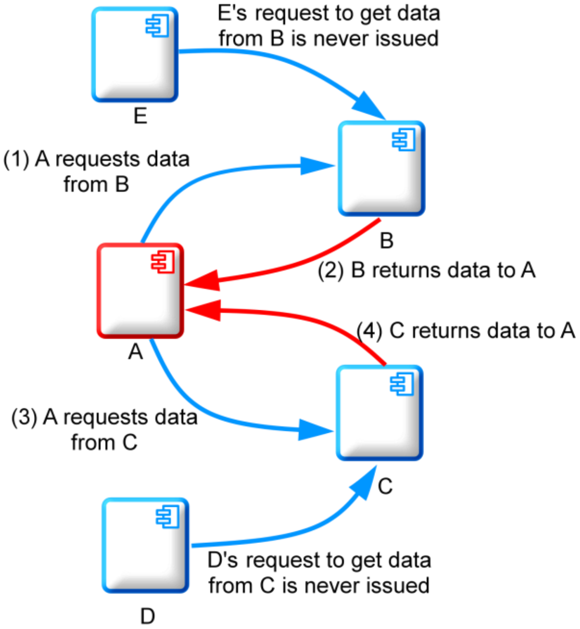

E's request to get data from B is never issued
E
(1) A requests data from B
B
(2) B returns data to A
A
(4) C returns data to A
(3) A requests data from C
C
D's request to get data from C is never issued
D

Figure 4.2 Limitation of the OpenMI "request and reply" data exchange mechanism for compositions with more than one downstream component.

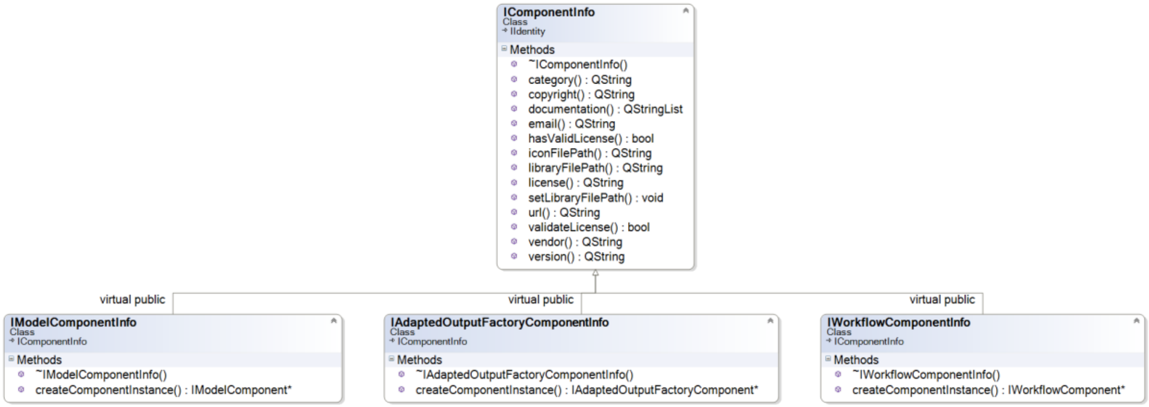


Figure 4.3 UML diagram for the IComponentInfo interface.

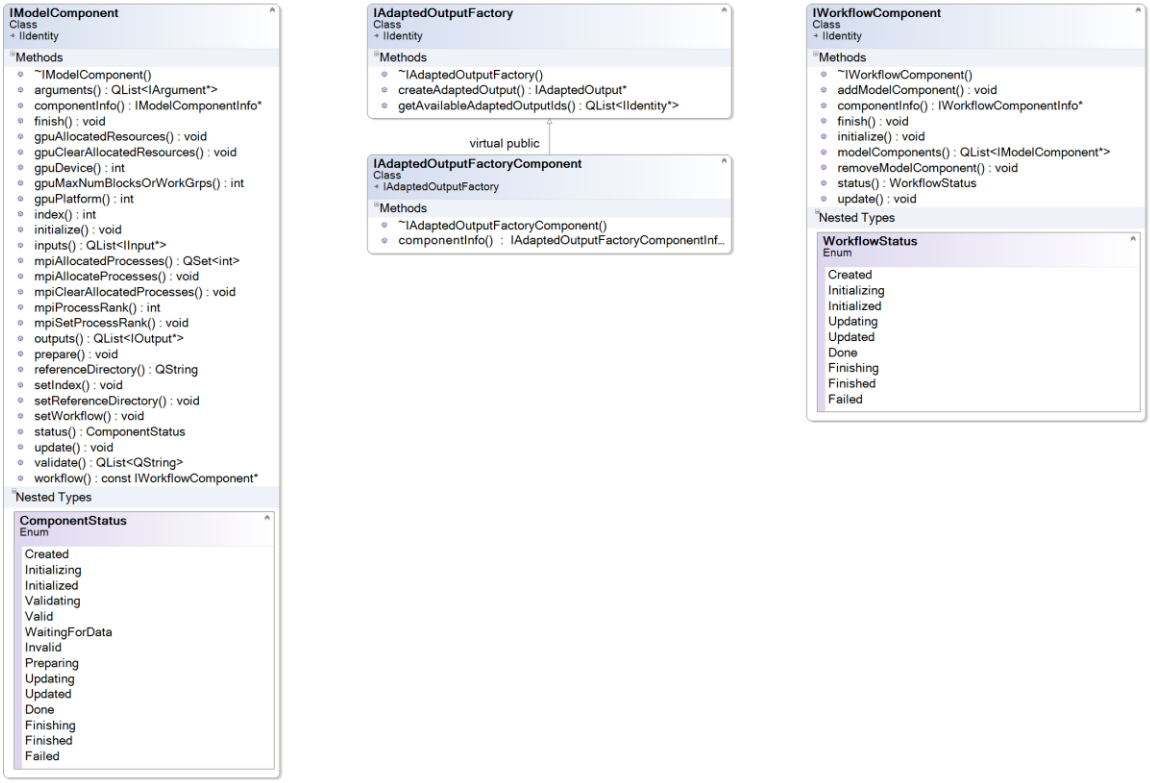

IModelComponent
Class
→ IIdentity
Methods
~IModelComponent()
arguments() : QList<IArgument*>
componentInfo() : IModelComponentInfo*
finish() : void
gpuAllocatedResources() : void
gpuClearAllocatedResources() : void
gpuDevice() : int
gpuMaxNumBlocksOrWorkGrps() : int
gpuPlatform() : int
index() : int
initialize() : void
inputs() : QList<IInput*>
mpiAllocatedProcesses() : QSet<int>
mpiAllocateProcesses() : void
mpiClearAllocatedProcesses() : void
mpiProcessRank() : int
mpiSetProcessRank() : void
outputs() : QList<IOutput*>
prepare() : void
referenceDirectory() : QString
setIndex() : void
setReferenceDirectory() : void
setWorkflow() : void
status() : ComponentStatus
update() : void
validate() : QList<QString>
workflow() : const IWorkflowComponent*
Nested Types
ComponentStatus
Enum
Created
Initializing
Initialized
Validating
Valid
WaitingForData
Invalid
Preparing
Updating
Updated
Done
Finishing
Finished
Failed
IAdaptedOutputFactory
Class
→ IIdentity
Methods
~IAdaptedOutputFactory()
createAdaptedOutput() : IAdaptedOutput*
getAvailableAdaptedOutputIds() : QList<IIdentity*>
virtual public
IAdaptedOutputFactoryComponent
Class
→ IAdaptedOutputFactory
Methods
~IAdaptedOutputFactoryComponent()
componentInfo() : IAdaptedOutputFactoryComponentInf...
IWorkflowComponent
Class
→ IIdentity
Methods
~IWorkflowComponent()
addModelComponent() : void
componentInfo() : IWorkflowComponentInfo*
finish() : void
initialize() : void
modelComponents() : QList<IModelComponent*>
removeModelComponent() : void
status() : WorkflowStatus
update() : void
Nested Types
WorkflowStatus
Enum
Created
Initializing
Initialized
Updating
Updated
Done
Finishing
Finished
Failed

Figure 4.4 UML diagram for HydroCouple’s new component types.

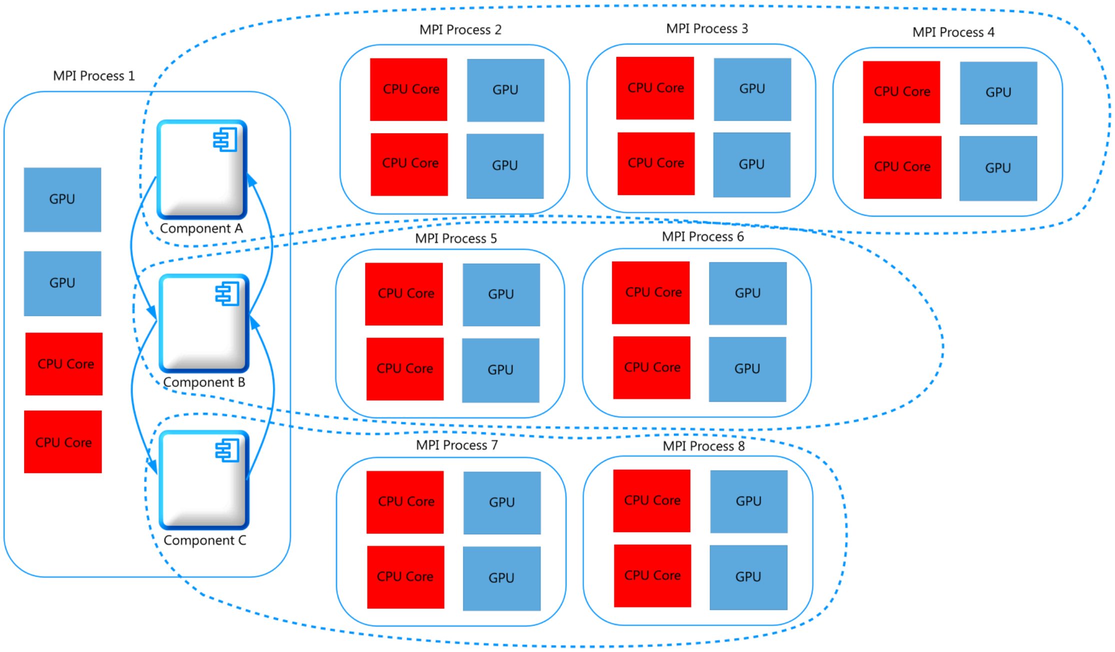
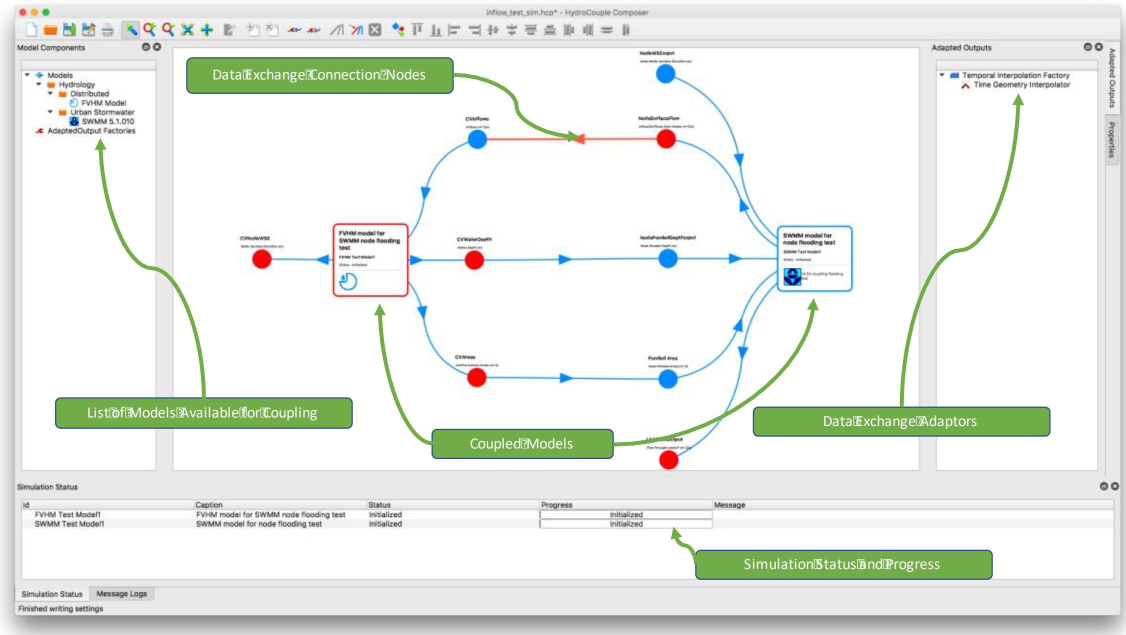


Figure 4.5 Example HydroCouple configuration on high performance heterogeneous computing infrastructure where each component is assigned a number of GPU (blue) and CPU (red) resources.



Simulation Status

Id	Caption	Status	Progress	Message
FVHM Test Model1	FVHM model for SWMM node flooding test	Initialized	Initialized	
SWMM Test Model1	SWMM model for node flooding test	Initialized	Initialized	

Finished writing settings

Figure 4.6 HydroCouple Composer graphical user interface.

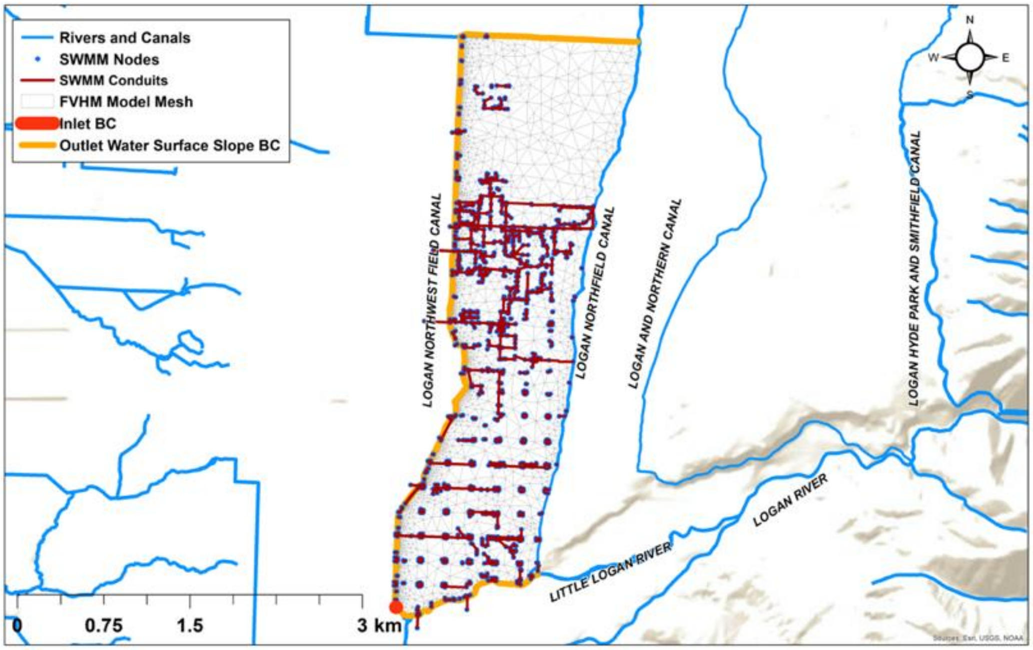


Figure 4.7 1D-2D coupled model boundaries.

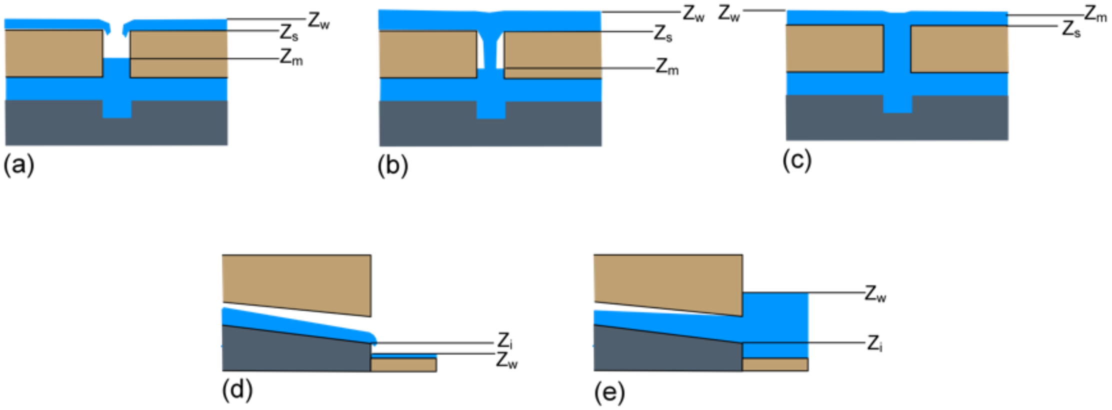


Figure 4.8 1D-2D Model interaction scenarios: (a) 1D Inlet water surface elevation (Z_m) less than 2D cell bottom elevation (Z_s) and 2D cell water depth less than A_{in}/w; (b) 1D Inlet water surface elevation (Z_m) less than 2D cell bottom elevation (Z_s) and 2D cell water depth greater than A_{in}/w; (c) 1D Inlet water surface elevation (Z_m) greater than 2D cell bottom elevation (Z_s); (d) Water surface elevation of 2D coupling cell (Z_w) less than outfall invert elevation (Z_i) of 1D model; (e) Water surface elevation of 2D cell greater than outfall invert elevation of 1D cell (Z_i).

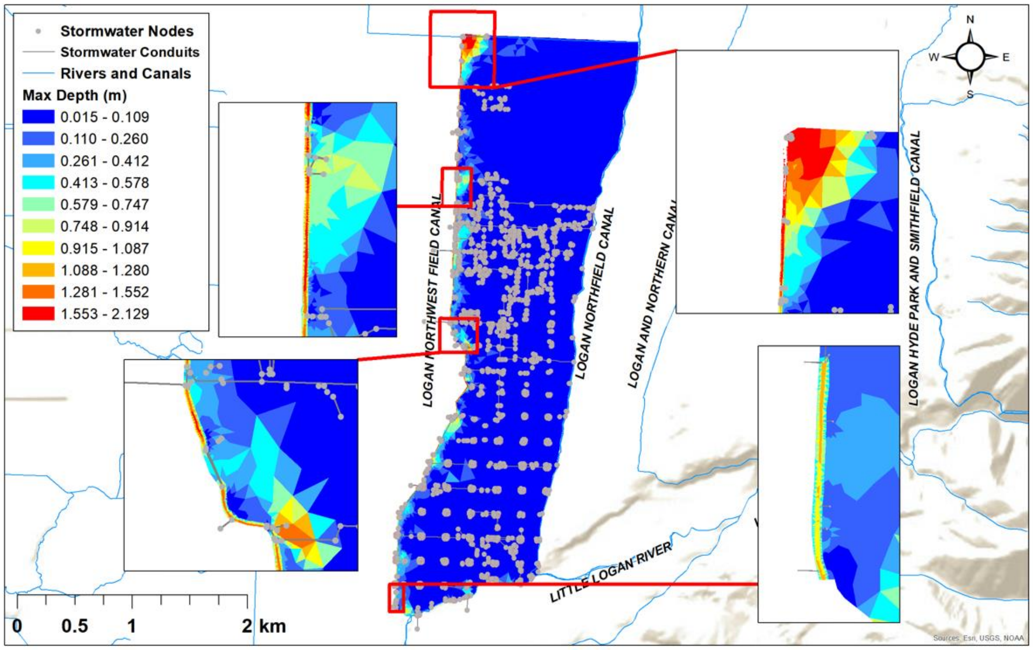


Figure 4.9 Map of maximum inundation depths for the simulation period.

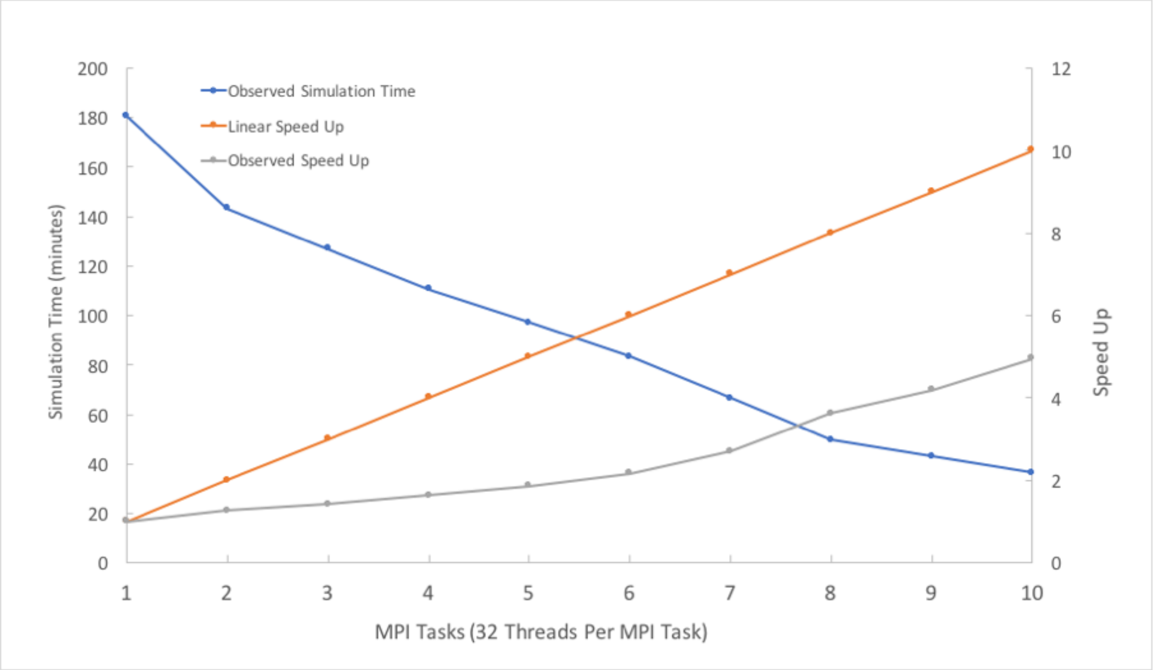


Figure 4.10 Speed up in simulation time versus increases in the number of MPI tasks.

CHAPTER 5

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

The research presented in this dissertation seeks to address the growing need to integrate models that simulate various hydrologic processes as well as models from other related earth system science disciplines so that complex environmental and natural resources management can be tackled more holistically. Efforts to combine or couple models for integrated water resources assessment are not new but have traditionally been accomplished using the tightly coupled approach, where models are stitched together and compiled into a monolithic executing unit. While this approach may lead to efficient codes and can be fine-tuned to address specific problems, there is an emerging consensus that a more systematic approach will be more helpful in achieving the outlined goals of model flexibility, re-use, and maintainability.

This consensus around the need for new approaches has also arisen because of the increasingly complex water resources management challenges society faces, which can no longer be investigated from the perspective of a single research discipline, with a model developed to primarily simulate a single process, or a model developed to address a single question. Model integration approaches that promote flexibility of process formulation selection, re-use of model formulations, and ease of maintainability of model codebases are, therefore, required to make these challenges more tractable. The significance of the work presented in this dissertation is that it quantifies the computational costs of different model integration approaches, directs modelers to methods for minimizing these costs, identifies coupling approaches and configurations that are most appropriate for any coupling scenarios (e.g., weir and orifice coupling configurations for 1D-2D coupled modeling scenario using the component-based modeling approach), and provides advancements to the software that supports integrated water resources modeling. All of these are necessary for realizing the promises of integrated and component-based modeling and for building a next generation of more modular, flexible, and customizable water resources models.

The workflow approach and the component-based modeling approach demonstrated through the research presented in this dissertation are two such approaches. The workflow approach is suitable for processing large datasets and for long running model simulations that are typically mediated by a database. For example, in the data centric workflow example described in Chapter 2, the weather, surface runoff, and streamflow forecasts used in developing the flood inundation delineations were stored on an iRODS data system. For models that need to communicate frequently, the latency from communicating through the database can lead to significant increases in computation times. In contrast, the component-based modeling approach involves direct, in-memory data exchange between model components at runtime, minimizing the latency related increases in computation times. However, the size of the data that can be exchanged between model components is limited by the hardware and operating system specifications of the computer on which the model is executed. While the component-based modeling approach can be adapted to take on the tasks undertaken by workflows, the opposite is not always feasible. The component-based modeling approach has, therefore, emerged as the preferred approach to achieve model integration in the earth systems modeling field.

Resolving the temporal and spatial scale mismatches that arise frequently between coupled hydrologic and hydraulic models must be done efficiently while ensuring that conservation laws are obeyed across the coupling boundaries between models. Since the spatial representation of hydrologic features in models do not typically change during a simulation, a technique that is often used to achieve computational efficiency is to map corresponding spatial features between coupled models at the initialization stage. This approach was used in the downscaling process described in Chapter 2 to transform the runoff from the coarsely gridded land surface model into streamflow in corresponding river reaches. This pre-mapping process involved deriving a matrix of grid cells versus the river reach catchments where each value in the matrix represents the fraction of the runoff grid cell that overlaps a sub-catchment of a river reach. This matrix of interpolation coefficients was then applied at every time step to transform runoff into streamflow. This same technique was also applied in the coupling of the 1D and 2D hydraulic models described in Chapter 4, where cells in the 2D model were mapped to corresponding nodes in the 1D model at the initializing stage so that the mapping could be used at each time step.

While the spatial interpolation technique used in Chapter 2 for downscaling runoff to streamflow performed well in forecasting streamflows, local effects were not captured for smaller watersheds. It is, therefore, important that a modeler investigate different spatial interpolation, downscaling, or up-scaling approaches that are most appropriate for any coupling application. It is also important that the software available for developing coupled models support multiple types of spatial interpolation and scaling approaches to ensure that model components can be connected in appropriate ways. The work presented in this dissertation directly addresses some of this need. The HydroCouple software provides support for various means for resolving spatial mismatches and temporal interpolation methods including the linear, quadratic, and cubic interpolation methods.

Results from Chapter 3 and 4 demonstrate that the coupling configuration adopted at the coupling interface between models plays an important role in model convergence for hydraulic modeling applications. Depending on the flow regime (i.e., critical or sub-critical) at the boundaries, a one-way or bi-directional data exchange between models may be required. Furthermore, for urban stormwater coupled modeling applications, where a 1D model is coupled to a 2D model, the relative water surface elevation difference between the two models at the coupling interface determines whether the weir or orifice equation needs to be adopted. Thus, the internal state of the coupled model components may dictate the required coupling configuration, requiring the type of adaptability demonstrated by the model components developed and described in Chapter 4.

For the component-based model coupling approach we adopted in Chapter 3, we discovered that the programming language mismatch between the OpenMI coupling framework written using C# and the SWMM model written using C led to data marshalling costs, which led to increases in simulation times. We anticipate that similar computational costs would be observed in any coupled modeling use case that requires marshalling data across programming languages. Additionally, we found that the data structures available in OpenMI 2.0 were not suitable for the more complex 1D-2D hydraulic coupled modeling study we subsequently undertook in Chapter 4. These challenges led to the development of the HydroCouple component-based modeling framework. While the HydroCouple interface definitions are language agnostic, the HydroCouple implementation was developed using C/C++ so HydroCouple can better support the large number of legacy hydrologic models that were developed using similar programming languages that are compiled to run directly on the operating system. Additionally, HydroCouple explicitly provides data structures that are widely used in the water resources modeling arena and supports simulations that harness high performance computing resources.

One way to achieve all the benefits promised by the component-based modeling approach is for practitioners to adopt open software frameworks and standards. Open software development efforts have been shown to be successful at spurring a sense a community among practitioners so that anyone can leverage the most innovative advancements contributed by members. The HydroCouple modeling framework and its suite of model components has, therefore, been made freely available on a Github repository (www.github.com). We have also been in communication with the larger community of coupled model practitioners and with the OpenMI association in particular, to create awareness of some of the advancements that we have implemented so that they can be adopted by the larger community.

While the challenges tackled in this dissertation are crucial, many areas of further research remain for integrated water resources modeling. At the fundamental level of challenges plaguing the use of component-based modeling frameworks are difficulties that arise from ontological differences between not only scientific disciplines, but also between science, management, and broader social views of the natural environment (Argent, 2004). This lack of shared understanding and a unified framework for model component metadata limits interoperability between modeling communities and the reuse of models across modeling frameworks due to ambiguity about models and their capabilities. Research efforts are needed to develop a comprehensive ontology that can be adopted to form the foundation for a component-based modeling framework. Some work has already been done along these lines in standardizing naming conventions and metadata for models, datasets, and their variables (e.g., Peckham, 2014; Peckham, 2016; Morsy et al., 2017), but consensus has not yet been achieved within the modeling community. Because ontologies are founded on logical languages, further work may enable automated reasoning to be employed to ensure model consistency and ontological-compliance (Rizzoli et al., 2008). This will ensure that the assemblage of components is not only correct from a technical standpoint but also from the scientific knowledge standpoint (Athanasiadis et al., 2006; Voinov and Shugart, 2013).

Another active area for further research is in exposing model components as web services, Remote Procedure Calls (RPC), and OGC’s Web Processing Services (WPS) (Čurn, 2007; Castronova et al., 2013; Bulatewicz et al., 2014). Most existing efforts have focused on integrating time series data available via standards-based web services as inputs to model components. Advancing the design of existing component-based modeling frameworks to integrate not only time series data but also more complex datasets including gridded climate and land change projection data into models through web-services are important advancements that need to be explored. On this front, the Hydroshare effort (Horsburgh et al., 2016), which provides an avenue for scientists to share hydrologic datasets and models as social objects can serve as a venue for achieving this goal. Within Hydroshare, scientists could collaboratively develop coupled model compositions that leverage other datasets within the system as inputs, allocate high performance computing resources for the simulation, and retrieve, analyze, and publish model results. This suggests a new paradigm in modeling where the model itself, input data, and outputs never exist on the modeler’s personal computer, but rather all of the data storage, execution, and analysis happen on remote computational resources made available through systems like HydroShare. Benefits of this approach include removal of the need for modelers to install specialized software to execute a model, easier access to high performance computational resources, and the ability to publish reproducible modeling workflows that can be executed in the cloud.

Hydrologic models often contain parameters that cannot be measured directly, but can only be meaningfully inferred through calibration to a historical record of input-output data (Vrugt et al., 2005). This inevitably leads to uncertain parameter estimates (and consequently to uncertain forecasts) due to structural errors in the model schematization, errors in the input and output measurement, and errors in initial conditions of state variables (Mantovan and Todini, 2006; Beven and Binley, 2014).

These challenges increase in a component-based modeling framework, as there are more models with a potentially large number of parameters that may have feedbacks with each other (Voinov and Shugart, 2013). The component-based modeling framework provides a natural avenue for testing different model structures by allowing model developers to swap models that simulate the same processes using different formulations within a coupled model system. Rizzoli et al. (2008) argue that automated parameter estimation, model validation, calibration, and other experiments carried out by modelers can be regarded as components in their own right, to which hierarchical decomposition and object-orientation can equally be applied. Further research needs to be conducted to investigate the feasibility of using the component-based modeling framework to improve process representation and investigation model uncertainty. One area of research interest where the component-based modeling approach might be applied to investigate model structure is the investigation of the best way to represent the basic spatial units of hydrologic models used in land surface models. Land surface models average spatial heterogeneity over large grid cells on the order of ~10 – 100 km. This averages out effects of topography, aspect, and hydrogeological properties, which are important in determining the locations where water flows and accumulates as well as processes including energy, water, and carbon flux exchanges with the atmosphere. A component-based modeling framework could allow investigators to couple hydrologic models having various spatial discretization strategies to atmospheric, oceanographic, and sea-ice models. The most appropriate discretization strategies for a coupled model composition could then be evaluated through a performance evaluation model component.

References

Argent, R.M., 2004. An Overview of Model Integration for Environmental Applications—components, Frameworks and Semantics. Environmental Modelling & Software 19:219–234. DOI: 10.1016/S1364-8152(03)00150-6

Athanasiadis, I.N., A.E. Rizzoli, M. Donatelli, and L. Carlini, 2006. Enriching Software Model Interfaces Using Ontology-Based Tools. Third Biennial Meeting of the Int'l Environmental Modelling and Software Society. Burlington, VT, USA. http://www.iemss.org/iemss2006/papers/s5/284_Athanasiadis_1.pdf. Accessed 8 May 2015.

Beven, K. and A. Binley, 2014. GLUE: 20 Years on: GLUE: 20 YEARS ON. Hydrological Processes 28:5897–5918. DOI:10.1002/hyp.10082

Bulatewicz, T., D. Andresen, S. Auvenshine, J. Peterson, and D.R. Steward, 2014. A Distributed Data Component for the Open Modeling Interface. Environmental Modelling & Software 57:138–151. DOI:10.1016/j.envsoft.2014.02.017

Castronova, A.M., J.L. Goodall, and M.M. Elag, 2013. Models as Web Services Using the Open Geospatial Consortium (OGC) Web Processing Service (WPS) Standard. Environmental Modelling & Software 41:72–83. DOI:10.1016/j.envsoft.2012.11.010

Čurn, J., 2007. Distribution for Open Modelling Interface and Environment. Masters Thesis, Univerzita Karlova v Praze, Prague, Czech Republic.

Horsburgh, J.S., M.M. Morsy, A.M. Castronova, J.L. Goodall, T. Gan, H. Yi, M.J. Stealey, and D.G. Tarboton, 2016. HydroShare: Sharing Diverse Environmental Data Types and Models as Social Objects with Application to the Hydrology Domain. JAWRA Journal of the American Water Resources Association 52:873–889. DOI:10.1111/1752-1688.12363

Mantovan, P. and E. Todini, 2006. Hydrological Forecasting Uncertainty Assessment: Incoherence of the GLUE Methodology. Journal of Hydrology 330:368–381. DOI: 10.1016/j.jhydrol.2006.04.046

Morsy, M.M., J.L. Goodall, A.M. Castronova, P. Dash, V. Merwade, J.M. Sadler, M.A. Rajib, J.S. Horsburgh, and D.G. Tarboton, 2017. Design of a Metadata Framework for Environmental Models with an Example Hydrologic Application in HydroShare. Environmental Modelling & Software 93:13–28. DOI:10.1016/j.envsoft.2017.02.028

Peckham, S., 2014. The CSDMS Standard Names: Cross-Domain Naming Conventions for Describing Process Models, Data Sets and Their Associated Variables.

International Congress on Environmental Modelling and Software. http://scholarsarchive.byu.edu/iemssconference/2014/Stream-A/12.

Peckham, S., 2016. Towards Standard Variable Names for Environmental Chemistry: Semantic Mediation and Extensions to the CSDMS Standard Names. International Congress on Environmental Modelling and Software. http://scholarsarchive.byu.edu/iemssconference/2016/Stream-A/1.

Rizzoli, A.E., G. Leavesley, J.C. Ascough II, R.M. Argent, I.N. Athanasiadis, V. Brilhante, F.H.A. Claeys, O. David, M. Donatelli, P. Gijsbers, D. Havlik, A. Kassahun, P. Krause, N.W.T. Quinn, H. Scholten, R.S. Sojda, and F. Villa, 2008. Integrated Modelling Frameworks for Environmental Assessment and Decision Support. A. A. V. A. E. R. and S. H. C. A.J. Jakeman (Editor). Developments in Integrated Environmental Assessment, Environmental Modelling, Software and Decision Support. Elsevier, pp. 101–118. DOI: 10.1016/S1574-101X(08)00607-8

Voinov, A. and H.H. Shugart, 2013. “Integronsters”, Integral and Integrated Modeling. Environmental Modelling & Software 39:149–158. DOI:10.1016/j.envsoft.2012.05.014

Vrugt, J.A., C.G.H. Diks, H.V. Gupta, W. Bouten, and J.M. Verstraten, 2005. Improved Treatment of Uncertainty in Hydrologic Modeling: Combining the Strengths of Global Optimization and Data Assimilation. Water Resources Research 41:W01017. DOI:10.1029/2004WR003059

APPENDICES

Appendix A The Finite Volume Hydrologic Model Component Formulations

The FVHM HydroCouple model solves the shallow water equations (SWE) (Equations A.1 and A.2) using the finite volume approximation over a triangular irregular network mesh:

$$\frac{\partial \zeta}{\partial t} + \nabla \cdot \left(h\vec{V}\right) = q \tag{A.1}$$

$$\frac{\partial h\vec{V}}{\partial t} + \nabla \cdot \left(h\vec{V}\vec{V}\right) = -gh\nabla\zeta - \frac{\tau_b}{\rho} + \nabla \cdot \left(\Gamma h\nabla\vec{V}\right) + Fh \tag{A.2}$$

where ζ is the water surface elevation; t is time; h is the water depth; $\vec{V}$ is the velocity vector; q is the sum of external fluxes; g is the acceleration due to gravity; $\tau_b$is the bed shear stress (friction) vector; ρ is water density; $\Gamma$is the sum of the kinematic (ν) and eddy viscosities (ν_t); and F is the vector sum of external forces. The bed shear stress is calculated using the Manning's roughness equation shown in Equation 3:

$$\begin{pmatrix}\tau_{bx}\\ \tau_{by}\end{pmatrix} = \frac{\rho g n^2}{\sqrt[3]{h}}\begin{pmatrix}u\\ v\end{pmatrix}\sqrt{u^2 + v^2} \tag{A.3}$$

where n is the Manning's roughness coefficient and u and v are the velocities in the x and y direction respectively. The FVHM component was developed to especially handle hydrologic modeling applications that often involve prolonged periods of dry spells with non-existent or small water depths, which is necessary to simulate areas with climate comparable Utah and the intermountain western U.S.

The finite volume approximation estimates the average value of a conserved quantity in an arbitrarily shaped control volume using an integral version of partial differential shallow water equations. The theorem underlying the finite volume method is Gauss's divergence theorem (Equation 4), which may be interpreted physically as the integral of the divergence of a vector (i.e., **a**) in a control volume (i.e., CV) is equal to the sum of the components of the vector normal (i.e., **n**) to surfaces of area of the control volume (i.e., A) (Versteeg and Malalasekera, 2007).

$$\int_{CV}(\nabla \cdot a)\, dV = \int_A n \cdot a\, dA \tag{A.4}$$

To illustrate how the Gauss theorem is used in the derivation of the finite volume numerical approximations for the shallow water equations in the FVHM Component, we apply it to the transport of velocity $\vec{V}$ in the control volume P surrounded by neighboring control volumes N1, N2, and N3 as depicted in Figure A.1. In Figure A.1, a, b, and c represent the nodes of the triangle for the control volume P; $\nabla\eta$, $\boldsymbol{e_\eta}$, $\boldsymbol{e_n}$, C and represent the length, unit vector, unit normal vector, and centroid of the common edge between control volume P and its neighboring control volume N1, respectively; $\nabla\xi$ and e_ξ represent the length and unit vector for the distance between the centroids of the control volumes P and N1 respectively; and $\mathbf{r}_{PC}$ and $\mathbf{r}_{CN}$ represent the vector distances between the centroids of the control volumes P and N1 with the centroid C of their common edge respectively.

In FVHM, the collocated grid arrangement used by several investigators including Peric (1985) and Lai (2009) is adopted. This grid arrangement involves calculating the control volume velocities, depths, and water surface elevations at the centroid of each control volume. Applying the Gauss theorem to the momentum conservation equations yields Equation 5.

$$\int_A \frac{\partial h\vec{V}}{\partial t} dA + \int_\eta e_n \cdot \left(h\vec{V}\vec{V}\right) d\eta = \int_A (-gh\nabla\zeta) dA - \int_A \left(\frac{\tau_b}{\rho}\right) dA + \int_\eta e_n \cdot \left(\Gamma h \nabla\vec{V}\right) d\eta + \int_A F dA \tag{A.5}$$

where A is the area of the control volume. The numerical approximations for the terms in Equation 5 are as follows:

$$\int_{\mathrm{A}} \frac{\partial \mathrm{h}\vec{\mathrm{V}}}{\partial \mathrm{t}} \mathrm{dA} = \frac{\left(\mathrm{h}^{\mathrm{n+1}}\vec{\mathrm{V}}^{\mathrm{n+1}} - \mathrm{h}^{\mathrm{n}}\vec{\mathrm{V}}^{\mathrm{n}}\right)\mathrm{A}}{\Delta \mathrm{t}} \quad \text{(A.6)}$$

$$\int_{\mathrm{A}} \left(\frac{\tau_{\mathrm{b}}}{\rho}\right) \mathrm{dA} = \frac{\tau_{\mathrm{b}}}{\rho} \mathrm{A} \quad \text{(A.7)}$$

$$\int_{A} (-gh\nabla\zeta) dA = -gh \begin{pmatrix} \frac{\partial \zeta}{\partial x} \\ \frac{\partial \zeta}{\partial y} \end{pmatrix} A \quad \text{(A.8)}$$

$$\int_{A} FhdA = FhA \quad \text{(A.9)}$$

where the superscript n+1 and n represents the current timestep and the previous timestep respectively, and Δt is the current timestep.

The water surface elevation spatial gradients $\frac{\partial \zeta}{\partial x}, \frac{\partial \zeta}{\partial y}$, and all spatial gradients in FHVM are estimated using the least-squares gradient reconstruction approach from cell centered values of neighboring cells in the previous time step/previous iteration. Referring to Figure A.1, the water surface elevation of a neighboring cell N (i.e., ζ_N) surrounding cell P can be estimated from the water surface gradient at P using Equation 10:

$$\zeta_N = \zeta_P + \left.\frac{\partial \zeta}{\partial x}\right|_P (\Delta x_{PN}) + \left.\frac{\partial \zeta}{\partial y}\right|_P (\Delta y_{PN}) \quad \text{(A.10)}$$

where Δx_{PN} and Δy_{PN} represent the distances in the x and y directions from the centroid of the control volume P to the centroid of the neighboring control volume N. Assembling Equation 10 for all neighboring cells into a linear system of equations yields Equation 11, which is solved using the QR decomposition method in FVHM.

$$\begin{bmatrix} \Delta x_{PN1} & \Delta y_{PN1} \\ \Delta x_{PN2} & \Delta y_{PN2} \\ \Delta x_{PN3} & \Delta y_{PN3} \end{bmatrix} \begin{bmatrix} \left.\frac{\partial \zeta}{\partial x}\right|_P \\ \left.\frac{\partial \zeta}{\partial y}\right|_P \end{bmatrix} = \begin{bmatrix} \zeta_1 - \zeta_P \\ \zeta_2 - \zeta_P \\ \zeta_3 - \zeta_P \end{bmatrix} \tag{A.11}$$

The derivation of the numerical approximations of the more complex diffusion and advection terms in the momentum equation are provided in the following sections.

A.1 Discretization of the Diffusion Term

The diffusion term in the momentum equation is discretized as follows:

$$\int_\eta e_n \cdot \left(\Gamma h \nabla \vec{V}\right) d\eta = \sum_{all\ sides} \Gamma_c h_c e_n \cdot \left(\frac{\partial \vec{V}}{\partial n} e_n + \frac{\partial \vec{V}}{\partial \eta} e_\eta\right) \Delta\eta \tag{A.12}$$

where Γ_c and h_c are the viscosity and the water depth at the centroid of the common edge between the control volume P and it neighboring cell. These values are estimated using gradients calculated from the gradients derived from the least-squares gradients reconstruction method described earlier.

It can be shown from trigonometry as detailed in Versteeg and Malalasekera (2007) that the direct gradient and cross diffusion terms of Equation A.12 for each neighboring cell can be represented by Equation A.13:

$$e_n \cdot \left(\frac{\partial \vec{V}}{\partial n} e_n + \frac{\partial \vec{V}}{\partial \eta} e_\eta\right) = \underbrace{\frac{e_n \cdot e_n}{e_n \cdot e_\xi} \frac{\vec{V}_N - \vec{V}_P}{\Delta \xi}}_{Direct\ Gradient} + \underbrace{\frac{e_\xi \cdot e_\eta}{e_n \cdot e_\xi} \frac{\vec{V}_b - \vec{V}_a}{\Delta \eta}}_{Cross-diffusion} \tag{A.13}$$

where $\vec{V}_P$ and $\vec{V}_N$ are the cell velocities for P and N respectively, and $\vec{V}_b$ and $\vec{V}_a$ are cell P's interpolated nodal velocities for the shared edge between cells P and N.

Two approaches are available in FVHM for computing the cell turbulent eddy viscosity. The first is the parabolic eddy viscosity model (Equation A.14):

$$\nu_t = c_t U_* h \tag{A.14}$$

where c_t is theoretically equal to $\frac{\kappa}{6}$, with κ being the von Kármán constant (Wu *et al.*, 2014). The second eddy viscosity model is the Smagorinsky–Lilly model (Smagorinsky, 1963) shown in Equation A.15:

$$\nu_t = c_s A \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \frac{1}{2}\left(\frac{\partial \bar{u}}{\partial y} + \frac{\partial \bar{v}}{\partial x}\right)^2} \tag{A.15}$$

where c_s is the Smagorinsky constant, with values that are usually between 0.1 – 0.2.

A.2 Discretization of the Advection Term

Following Lai (2009), the advection term in the momentum equation is discretized as follows:

$$\int_\eta e_n \cdot \left(h\vec{V}\vec{V}\right) d\eta = \sum_{all\ sides} (h_c V_c)^{n+1,\#} \left(\vec{V}_c\right)^{n+1} \Delta\eta \tag{A.16}$$

where the superscript *n*+1,# refers to the previous iteration for the current timestep; V_c is the normal velocity to the current edge at its centroid, where a positive value indicates an outward flow from the control volume and a negative value indicates inflow into the control volume; and $\vec{V}_c$ is the velocity vector at the centroid of the current edge that is to be calculated at the current timestep.

$\vec{V}_c$ is estimated as a function of the velocity of the current cell P and its neighboring upstream and downstream cell velocities using Equation A.17:

$$\vec{V}_c = \vec{V}_P + \frac{1}{2}\psi(r)\left(\vec{V}_D + \vec{V}_P\right) \tag{A.17}$$

where $\vec{V}_D$ is the velocity of the control volume downstream of the current cell P, and $\psi(r)$ is a flux limiting function of r. The variable r is the upwind ratio of consecutive gradients of velocity defined using Equation A.18:

$$r = \frac{\vec{V}_P - \vec{V}_U}{\vec{V}_D - \vec{V}_P} \tag{A.18}$$

where $\vec{V}_U$ is the velocity of the control volume upstream of the current cell P. Equation A.17 assumes that the centroid of the edge C is equidistant from the centroids of the bounding control volumes, hence the multiplication factor 0.5. However, for unstructured grids this may not be true. Therefore, an inverse distance interpolation weighting factor L is applied in Equation A.19 instead of the 0.5 as recommended by (Denner and van Wachem, 2015):

$$L = \frac{r_{PC}}{r_{PC} + r_{CN}} \tag{A.19}$$

Assuming the direction of flow is from P to N1, it is easy to locate the downstream control volume with velocity $\vec{V}_D$ in Figure 4.2 that is to be used in calculating r. However, finding the upstream control volume $\vec{V}_U$ is not straightforward. To overcome this challenge, Darwish and Moukalled (2003) derived Equation A.20 for calculating r:

$$r = \frac{2\nabla\vec{V}_P \cdot r_{PN}}{\vec{V}_D - \vec{V}_U} \tag{A.20}$$

where r_{PN} is the vector distance between the centroids of the control volumes P and N.

Using $\psi(r)$ = 0 leads to the edge velocity being the same as the cell velocity representing the upwind differencing scheme. While the upwind differencing scheme is stable and results in smooth solutions, it is only first-order accurate. $\psi(r)$ = 1 represents the central differencing scheme, which, while second-order accurate, can lead to spurious oscillations with problems that exhibit sharp discontinuities in velocities as is common with higher-order schemes. To obtain stable and non-oscillatory solutions for higher-order schemes, the function $\psi(r)$ must be monotonicity-preserving. Monotonicity-preserving schemes ensure that solutions do not create local extrema. Additionally, the value of a local minimum must be non-decreasing and the value of a local maximum must be non-increasing (Versteeg and Malalasekera, 2007). Monotonicity-preserving schemes have a property that the total variation (TV) (i.e., Equation A.21) of the discrete solutions should be total variation diminishing (TVD). TVD schemes are characterized with TV values that decrease with time as shown in Equation A.22.

$$TV = \int \left|\frac{\partial u}{\partial x}\right| dx \tag{A.21}$$

$$TV(u^{n+1}) \leq TV(u^{n}) \tag{A.22}$$

Sweby (1984) provides the necessary and sufficient conditions for $\psi(r)$ to be TVD in terms of a relationship between r and $\psi(r)$. Several $\psi(r)$ functions that meet these conditions are provided in FVHM, including those shown in Table A.1.

A.3 Pressure Velocity Coupling

The discretization for the momentum equations provided can be organized into a linearized system of equations for all control volumes based on the control volume center values and can be solved implicitly for new velocities using Equation A.23:

$$a_P\vec{V}_P = \sum_{N=1}^{3} a_N\vec{V}_N - ghA\nabla\zeta + \vec{S}\,hA \tag{A.23}$$

where a_P is the coefficient of the velocity of the current cell P, a_N are the coefficients of the neighboring cells, and $\vec{S}$ is the sum of the external forces and constants acting on the control volume. We seek to solve Equation A.23 for control volume velocities as well the water surface elevations. The momentum equation is non-linear because it involves the multiplication of two velocity terms. Additionally, for incompressible flows, cell velocities and pressures are coupled in a non-linear fashion through the momentum and continuity equations. In FVHM, the Semi-Implicit Method for Pressure Linked Equations (SIMPLE; Patankar and Spalding, 1972) or alternatively SIMPLE-Consistent (SIMPLEC; Doormaal and Raithby, 1984) iterative solution procedures are adopted to deal with these nonlinearities.

First, it is important to note that in the collocated grid arrangement, where velocities and water surface elevation values are estimated for the centroid of each control volume, a highly non-uniform water surface elevation field can act like a uniform field when the gradients of the water surface elevation fields are calculated numerically. This may lead to the well-known “checker-board” pressure field effect, which, in turn, leads to non-physical results (Versteeg and Malalasekera, 2007). To overcome this problem, Rhie and Chow (1983) proposed the momentum interpolation equation shown in Equation A.24 to calculate the edge normal velocity V_c:

$$V_c = \left((1.0-L)\vec{V}_P + (L)\vec{V}_N\right)\cdot e_n - \frac{1.0}{r_\xi \cdot e_n}\left(\frac{(1.0-L)A_P}{a_P} + \frac{(L)A_N}{a_N}\right)\left(gh_c(\zeta_N\ -\ \zeta_P) + \right.$$

$$\left.(gh_P(1.0-L)(\nabla\zeta)_P + gh_N(L)(\nabla\zeta)_N)\cdot r_\xi\right) \tag{A.24}$$

The SIMPLE and SIMPLEC iterative solution procedure begins by using the initial or previous iteration water surface elevation values ζ^* to solve the momentum equation (i.e., Equation A.23), for an intermediate velocity field $\vec{V}^*$ as shown in Equation A.25.

$$a_P\vec{V}_P{}^* = \sum_{N=1}^{3} a_N\vec{V}_N{}^* - ghA\nabla\zeta^* + \vec{S} \qquad \textit{(A.25)}$$

Since the initial water surface elevation used is a guess from the previous iteration or time step, the computed velocities are likely not correct. A water surface elevation ζ' that corrects the water surface elevation ζ^* is, therefore, defined as shown in Equation A.26. Similarly, a new velocity $\vec{V}'$that corrects the calculated intermediate velocity $\vec{V}^*$ is also defined.

$$\zeta_P = \zeta^*{}_P + \zeta'{}_P \qquad \text{(A.26)}$$

$$\vec{V}_P = \vec{V}^*{}_P + \vec{V}'{}_P \qquad \text{(A.27)}$$

Subtracting Equation A.25 from 23 yields:

$$a_P\vec{V}_P{}' = \sum_{N=1}^{3} a_N\vec{V}_N{}' - ghA\nabla\zeta' \qquad \text{(A.28)}$$

The velocity correction is then calculated from Equation A.28 as:

$$\vec{V}_P{}' = \frac{\sum_{N=1}^{3} a_N\vec{V}_N{}' - gh\nabla\zeta'}{a_P} \qquad \text{(A.29)}$$

Ignoring the minor terms in Equation A.29 and inserting it into Equation A.27 yields:

$$\vec{V}_P = \vec{V}^*{}_P + \frac{-ghA\nabla\zeta'}{a_P} \qquad \text{(A.30)}$$

The velocities estimated in Equation A.25 do not satisfy the continuity equation. Therefore, the water surface elevation correction equation (i.e., Equation A.26) and the correction velocity equation (i.e., Equation A.30) are used in the continuity equation to derive the water surface elevation correction values ζ'. These computations form the basis for the SIMPLE method. To illustrate it, we derive the finite volume approximation of the continuity equation as:

$$\left(\frac{{\zeta^{n+1}}_P - {\zeta^{n}}_P}{\Delta t} A_P\right) = \sum_{N=1}^{3} \Delta\eta h_c V_c + qA_P \tag{A.31}$$

where q represents external inflows. Inserting the water surface elevation correction equation and the Rhie-Chow interpolated control volume edge velocities from the velocity correction equation into the continuity equation yields:

$$\left(\frac{\left({\zeta^*}_P + {\zeta'}_P\right)^{n+1} - {\zeta^{n}}_P}{\Delta t} A_P\right) = \sum_{N=1}^{3} \left[\Delta\eta\, {h^{n+1}}_c \left(V_c - \frac{1.0}{r_{\xi}.e_n}\left(\frac{(1.0-L)A_P}{a_P} + \frac{(L)A_N}{a_N}\right) g h_c\left({\zeta'}_N - {\zeta'}_P\right)\right)^{n+1}\right] + qA_P \tag{A.32}$$

In the SIMPLEC method, the minor terms ignored in Equation A.30 are included in the continuity equation to estimate that water surface elevation correction values.

To recap, for each time step, the solution process begins by using the initial or previous iteration values of water surface elevations and velocities to calculate intermediate velocity values $\vec{V}^*$ for each control volume. The water surface correction equation (Equation A.32) is then solved to obtain correction values ζ' for each control volume. The water surface values and velocities are then corrected using Equations 26 and 27, respectively. If convergence is not achieved, the whole process is repeated.

All the systems of equations generated by FVHM are solved using the algebraic, multigrid, preconditioned, generalized, minimal residual method (GMRES) from the hypre software library (Falgout and Yang, 2002), which solves large, sparse linear systems of equations on massively parallel computers. Even though the Courant-

Friedrichs-Lewy (CFL) condition for stability does not apply for the implicit time-marching approach adopted in FVHM, the use of an excessively large time step can lead to inaccurate estimates (Durran, 2013). An adaptive time step approach was, therefore, adopted using a user specified maximum Courant number as a controlling variable. The time step at the beginning of each iteration is estimated by dividing the maximum user specified Courant number (C_o) by the maximum Courant number for the control volumes at the current time step as shown in Equation A.33.

$$\Delta t = \frac{C_o}{\left(\sum \frac{Outgoing\ fluxes}{Volume}\right)_{max}} \qquad \text{(A.33)}$$

A.4 Wetting and Drying

Tracking of the wetting front at the boundaries between wet and dry cells in hydraulic models is important because of the numerical instabilities that would arise from the unrealistically high velocities that would be calculated by dividing volumetric fluxes by the small water depths in dry cells (Kim *et al.*, 2012). Many approaches have been proposed for the proper treatment of wetting and drying cells in hydraulic models, including the thin film, element removal, depth extrapolation, and negative depth algorithms as discussed by (Medeiros and Hagen, 2013). A common feature for the treatment of wetting and drying cells in many hydraulic models involves first classifying cells as wet, partially wetted, or dry depending on the number of cell nodes that are submerged. A wet cell has a water surface elevation that submerges all the nodes of the cell by a certain small threshold value (e.g., 1e-7 m). A partially wetted cell has a water surface elevation that submerges at least one node of a cell by a certain small threshold value. A dry cell has a water surface elevation that does not submerge any of the nodes of a cell. For each time step, the momentum equations are only solved for wet cells, and velocities for dry cells are set to zero. Velocities and water surface elevations for partially submerged cells are then extrapolated from neighboring wet cells.

The application of this approach to hydrological simulations that solve the full dynamic wave model, however, introduces some challenges. Hydrological simulations often involve long periods with small or no runoff generation. Additionally, hydrological models often involve large areas. Coarse and often steeply sloped computational cells are often employed for computational efficiency. This leads to the frequent occurrence of partially wetted cells that cause a no-flow phenomenon where water is unable to leave a cell because of ponding in the lowest corner of a cell where the water surface elevation is below the two nearest cell edge midpoints (Kim et al., 2012) (Begnudelli et al., 2008). Also, the typical assumption made in many hydraulic models that the water surface elevation at the centroid is equal to the water depth plus the elevation of the centroid of the cell does not hold for partially wetted cells. To address this challenge, we adopted the volume-free surface relationship (VFR) proposed by Begnudelli and Sanders (2006) in FVHM to deal with partially wetted cells. The VFR relationship makes a distinction between the free water surface elevation at the centroid and the depth at the centroid by assuming sheet-flow for partially wetted cells. This is done by calculating the flow depth as a ratio between the fluid volume in the cell and the area of the cell. The VFR approach provides equations to quickly transform water surface elevation to depths and vice versa for triangular cells to support modeling.

In FVHM, the momentum equations are solved for wet cells and partially wetted cells with depths above a specified threshold while setting velocities for dry cells to zero. The mass balance equations are then solved for all cells in the model domain.

A.5 FVHM Component Verification

The FVHM component was verified using test problems 1 and 6 from (MacDonald, 1996). Problem 1 involves subcritical flow through a channel with a rectangular cross-section. Problem 6 involves flow through a channel with a rectangular cross-section where there is a transition from subcritical flow to critical flow and then a hydraulic jump (i.e., transcritical) near the end of a channel. The attributes for the two test problems are shown in Table A.2.

References

van Albada, G.D., B. van Leer, and W.W. Roberts Jr., 1982. A Comparative Study of Computational Methods in Cosmic Gas Dynamics. Astronomy and Astrophysics 108:76–84.

Begnudelli, L. and B.F. Sanders, 2006. Unstructured Grid Finite-Volume Algorithm for Shallow-Water Flow and Scalar Transport with Wetting and Drying. Journal of Hydraulic Engineering 132:371–384. DOI:10.1061/(ASCE)0733-9429(2006)132:4(371)

Darwish, M.S. and F. Moukalled, 2003. TVD Schemes for Unstructured Grids. International Journal of Heat and Mass Transfer 46:599–611. DOI:10.1016/S0017-9310(02)00330-7

Denner, F. and B.G.M. van Wachem, 2015. TVD Differencing on Three-Dimensional Unstructured Meshes with Monotonicity-Preserving Correction of Mesh Skewness. Journal of Computational Physics 298:466–479. DOI:10.1016/j.jcp.2015.06.008

Doormaal, J.P.V. and G.D. Raithby, 1984. Enhancements of the Simple Method for Predicting Incompressible Fluid Flows. Numerical Heat Transfer 7:147–163. DOI:10.1080/01495728408961817

Durran, D.R., 2013. Numerical Methods for Wave Equations in Geophysical Fluid Dynamics. Springer Science & Business Media.

Falgout, R.D. and U.M. Yang, 2002. Hypre: A Library of High Performance Preconditioners. P. M. A. Sloot, A. G. Hoekstra, C. J. K. Tan, and J. J. Dongarra (Editors). Computational Science — ICCS 2002,Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 632–641.

Kim, J., A. Warnock, V.Y. Ivanov, and N.D. Katopodes, 2012. Coupled Modeling of Hydrologic and Hydrodynamic Processes Including Overland and Channel Flow. Advances in Water Resources 37:104–126. DOI:10.1016/j.advwatres.2011.11.009

Lai, Y.G., 2009. Two-Dimensional Depth-Averaged Flow Modeling with an Unstructured Hybrid Mesh. Journal of Hydraulic Engineering 136:12–23. DOI:10.1061/(ASCE)HY.1943-7900.0000134

van Leer, B., 1974. Towards the Ultimate Conservative Difference Scheme. II. Monotonicity and Conservation Combined in a Second-Order Scheme. Journal of Computational Physics 14:361–370. DOI:10.1016/0021-9991(74)90019-9

Leonard, B.P., 1988. Simple High-Accuracy Resolution Program for Convective

Modelling of Discontinuities. International Journal for Numerical Methods in Fluids 8:1291–1318. DOI:10.1002/fld.1650081013

Lien, F.S. and M.A. Leschziner, 1994. Upstream Monotonic Interpolation for Scalar Transport with Application to Complex Turbulent Flows. International Journal for Numerical Methods in Fluids 19:527–548. DOI:10.1002/fld.1650190606

MacDonald, I., 1996. Analysis and Computation of Steady Open Channel Flow. University of Reading Reading, UK. http://www.henley.reading.ac.uk/web/FILES/maths/I_macdonald-thesis.pdf. Accessed 2 Oct 2016.

Medeiros, S.C. and S.C. Hagen, 2013. Review of Wetting and Drying Algorithms for Numerical Tidal Flow Models. International Journal for Numerical Methods in Fluids 71:473–487. DOI:10.1002/fld.3668

Patankar, S.V. and D.B. Spalding, 1972. A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows. International Journal of Heat and Mass Transfer 15:1787–1806. DOI:10.1016/0017-9310(72)90054-3

Peric, M., 1985. A Finite Volume Method for the Prediction of Three-Dimensional Fluid Flow in Complex Ducts. Ph.D. Dissertation, Imperial College London (University of London). http://hdl.handle.net/10044/1/7601. Accessed 30 Jun 2017.

Rhie, C.M. and W.L. Chow, 1983. Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation. AIAA Journal 21:1525–1532. DOI:10.2514/3.8284

Roe, P.L., 1985. Some Contributions to the Modelling of Discontinuous Flows. Large-scale computations in fluid mechanics; Proceedings of the Fifteenth Summer Seminar on Applied Mathematics, La Jolla, CA. pp. 163–193.

Smagorinsky, J., 1963. General Circulation Experiments with the Primitive Equations. Monthly Weather Review 91:99–164. DOI:10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2

Sweby, P., 1984. High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws. SIAM Journal on Numerical Analysis 21:995–1011. DOI:10.1137/0721062

Versteeg, H.K. and W. Malalasekera, 2007. An Introduction to Computational Fluid Dynamics: The Finite Volume Method. Pearson Education Ltd., Harlow, England; New York.

Warnock, A., J. Kim, V. Ivanov, and N.D. Katopodes, 2014. Self-Adaptive Kinematic-Dynamic Model for Overland Flow. Journal of Hydraulic Engineering 140:169–

181. DOI:10.1061/(ASCE)HY.1943-7900.0000815

Wu, W., P. Wang, and N. Chiba, 2014. Comparison of Five Depth-Averaged 2-D Turbulence Models for River Flows. Archives of Hydro-Engineering and Environmental Mechanics 51:183–200.

Tables

Table A.1 TVD flux limiters.

Name	Limiter function $\boldsymbol{\psi(r)}$	Source
Van Leer	$\frac{r+\lvert r\rvert}{1+r}$	van Leer (1974)
Van Albada	$\frac{r+r^2}{1+r^2}$	van Albada *et al.* (1982)
UMIST	$\max\left[0, min\left(2r, \frac{1+3r}{4}, \frac{3+r}{4}, 2\right)\right]$	Lien and Leschziner (1994)
QUICK	$\max\left[0, min\left(2r, \frac{3+r}{4}, 2\right)\right]$	Leonard (1988)
Min-Mod	$\begin{cases} \min(r,1) & if\ r > 0 \\ 0 & if\ r \leq 0 \end{cases}$	Roe (1985)

Table A.2 Properties for (MacDonald, 1996) test problems.

Problem	Channel Width(m)	Channel Length (m)	Manning's Roughness (n)	Inlet Flow (m^3/s)	Outlet Water Surface Elevation (m)
1	10	1000	0.03	20	0.800054
6	10	150	0.03	20	1.700225

Figures

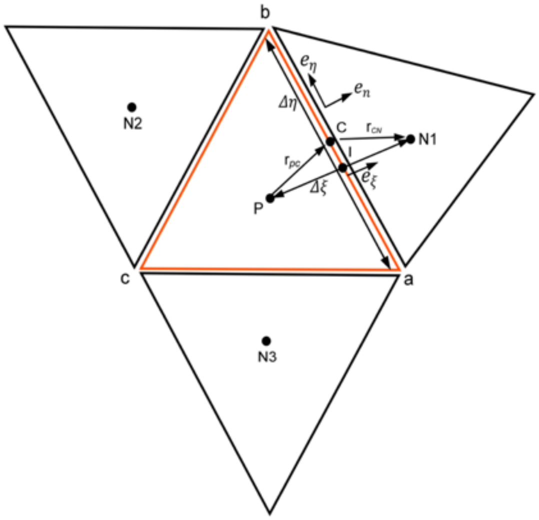


Figure A.1 Control volume P surrounded by neighboring control volumes N1, N2, and N3.

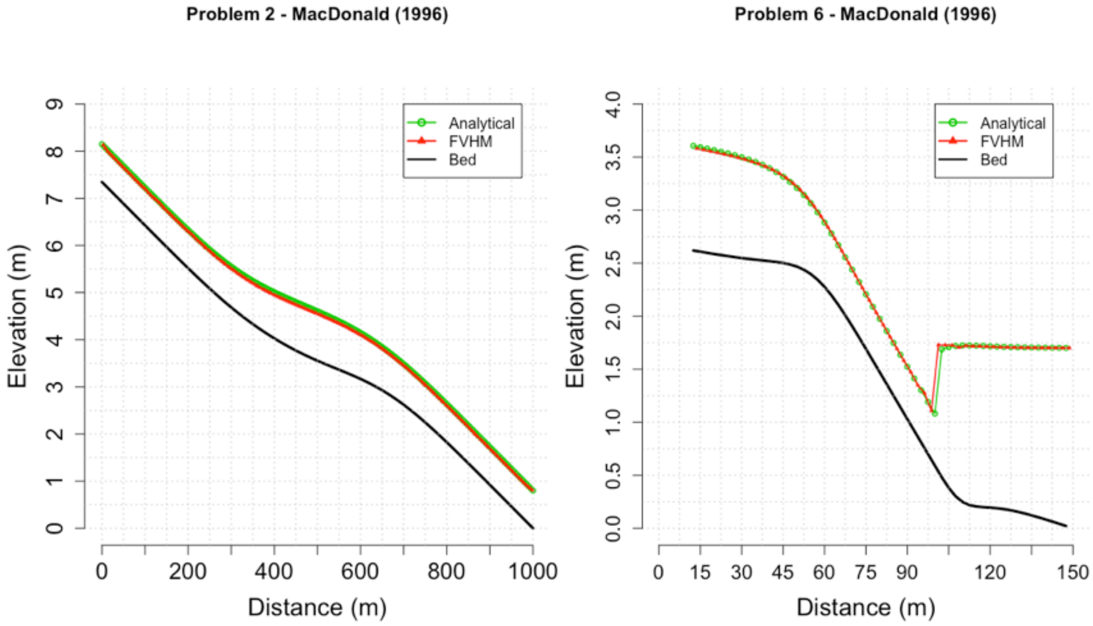
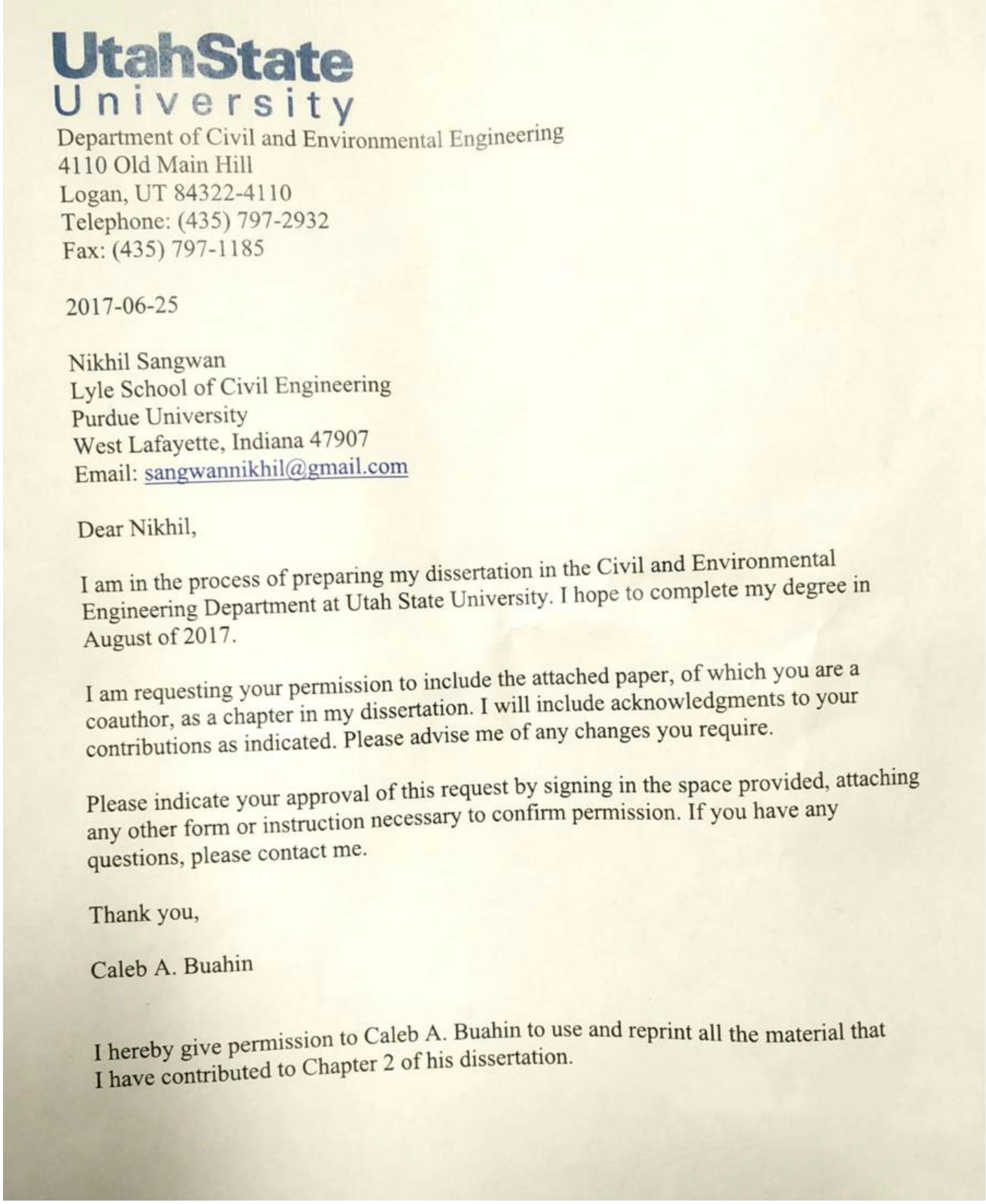


Figure A.2 Comparison of analytical results from MacDonald (1996) test problems with FVHM.

Appendix B Coauthor Approval Letters



UtahState
University

Department of Civil and Environmental Engineering
4110 Old Main Hill
Logan, UT 84322-4110
Telephone: (435) 797-2932
Fax: (435) 797-1185

2017-06-25

Nikhil Sangwan
Lyle School of Civil Engineering
Purdue University
West Lafayette, Indiana 47907
Email: sangwannikhil@gmail.com

Dear Nikhil,

I am in the process of preparing my dissertation in the Civil and Environmental Engineering Department at Utah State University. I hope to complete my degree in August of 2017.

I am requesting your permission to include the attached paper, of which you are a coauthor, as a chapter in my dissertation. I will include acknowledgments to your contributions as indicated. Please advise me of any changes you require.

Please indicate your approval of this request by signing in the space provided, attaching any other form or instruction necessary to confirm permission. If you have any questions, please contact me.

Thank you,

Caleb A. Buahin

I hereby give permission to Caleb A. Buahin to use and reprint all the material that I have contributed to Chapter 2 of his dissertation.

UtahState
University

Department of Civil and Environmental Engineering
4110 Old Main Hill
Logan, UT 84322-4110
Telephone: (435) 797-2932
Fax: (435) 797-1185

2017-06-25

Cassandra Fagan
Department of Civil, Architectural, and Environmental Engineering
University of Texas
Austin, Texas 78712
Email: fagan.cassandra@gmail.com

Dear Cassandra,

I am in the process of preparing my dissertation in the Civil and Environmental Engineering Department at Utah State University. I hope to complete my degree in August of 2017.

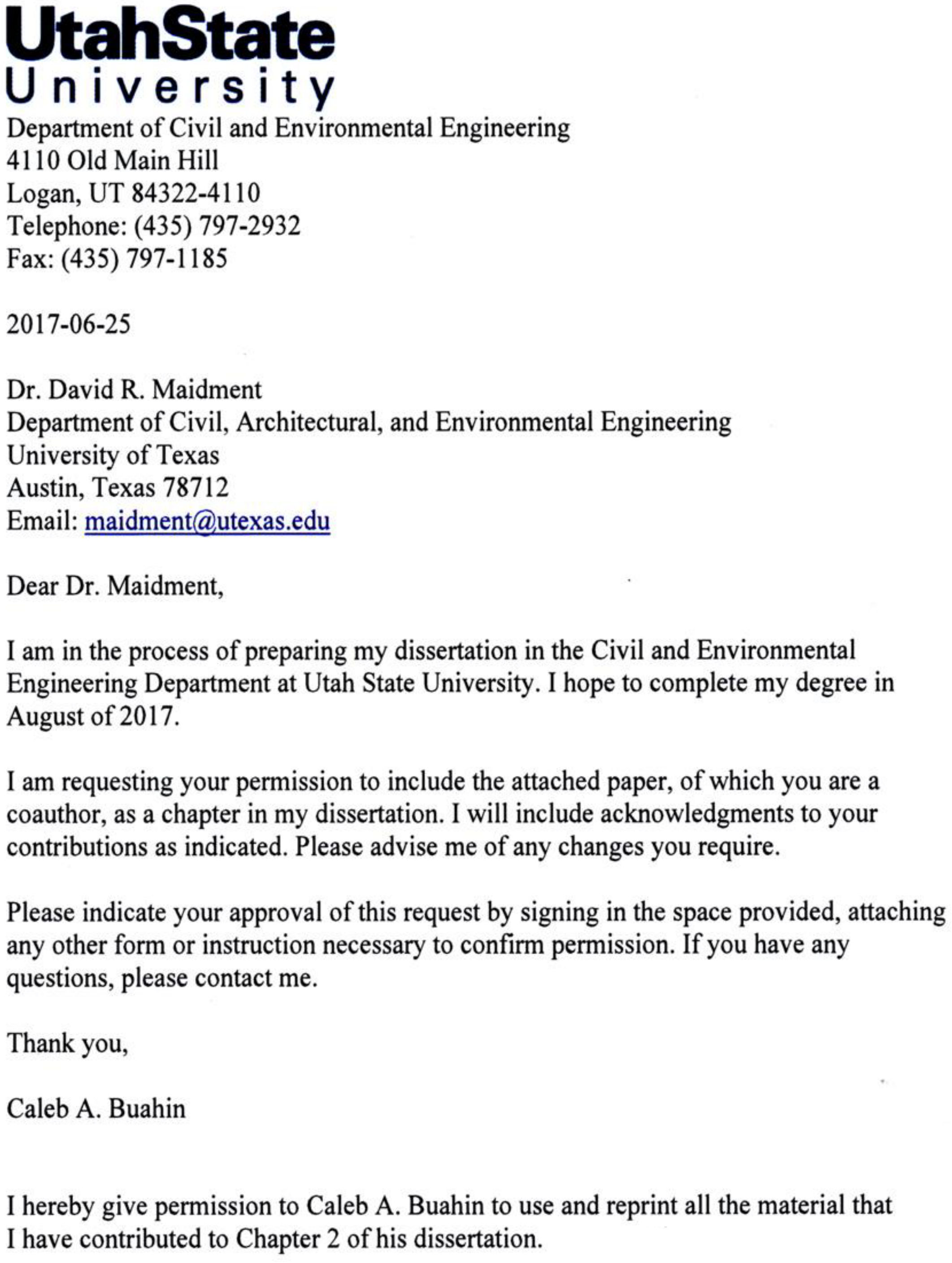
I am requesting your permission to include the attached paper, of which you are a coauthor, as a chapter in my dissertation. I will include acknowledgments to your contributions as indicated. Please advise me of any changes you require.

Please indicate your approval of this request by signing in the space provided, attaching any other form or instruction necessary to confirm permission. If you have any questions, please contact me.

Thank you,

Caleb A. Buahin

I hereby give permission to Caleb A. Buahin to use and reprint all the material that I have contributed to Chapter 2 of his dissertation.



UtahState University

Department of Civil and Environmental Engineering
4110 Old Main Hill
Logan, UT 84322-4110
Telephone: (435) 797-2932
Fax: (435) 797-1185

2017-06-25

Dr. David R. Maidment
Department of Civil, Architectural, and Environmental Engineering
University of Texas
Austin, Texas 78712
Email: maidment@utexas.edu

Dear Dr. Maidment,

I am in the process of preparing my dissertation in the Civil and Environmental Engineering Department at Utah State University. I hope to complete my degree in August of 2017.

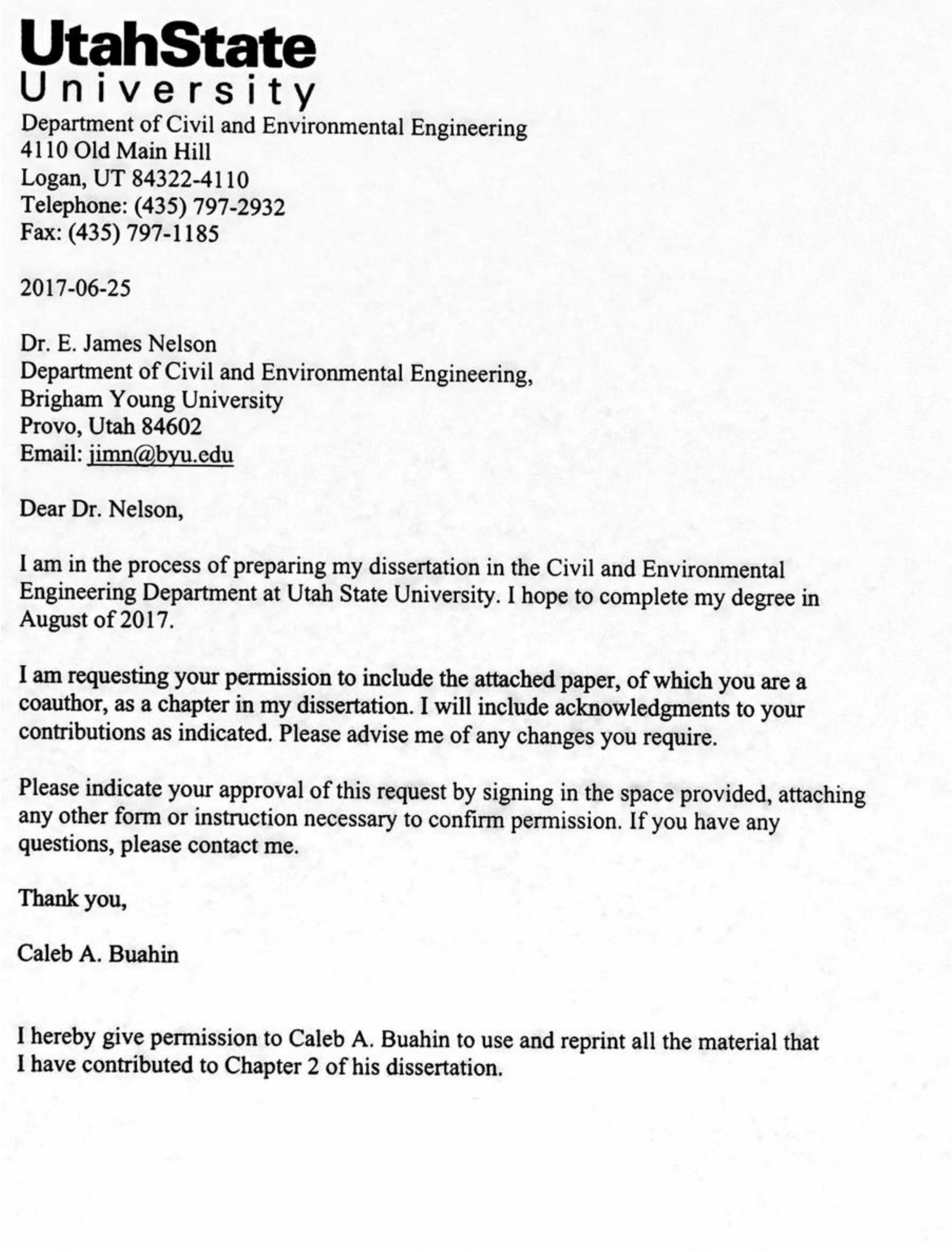
I am requesting your permission to include the attached paper, of which you are a coauthor, as a chapter in my dissertation. I will include acknowledgments to your contributions as indicated. Please advise me of any changes you require.

Please indicate your approval of this request by signing in the space provided, attaching any other form or instruction necessary to confirm permission. If you have any questions, please contact me.

Thank you,

Caleb A. Buahin

I hereby give permission to Caleb A. Buahin to use and reprint all the material that I have contributed to Chapter 2 of his dissertation.



UtahState University

Department of Civil and Environmental Engineering
4110 Old Main Hill
Logan, UT 84322-4110
Telephone: (435) 797-2932
Fax: (435) 797-1185

2017-06-25

Dr. E. James Nelson
Department of Civil and Environmental Engineering,
Brigham Young University
Provo, Utah 84602
Email: jimn@byu.edu

Dear Dr. Nelson,

I am in the process of preparing my dissertation in the Civil and Environmental Engineering Department at Utah State University. I hope to complete my degree in August of 2017.

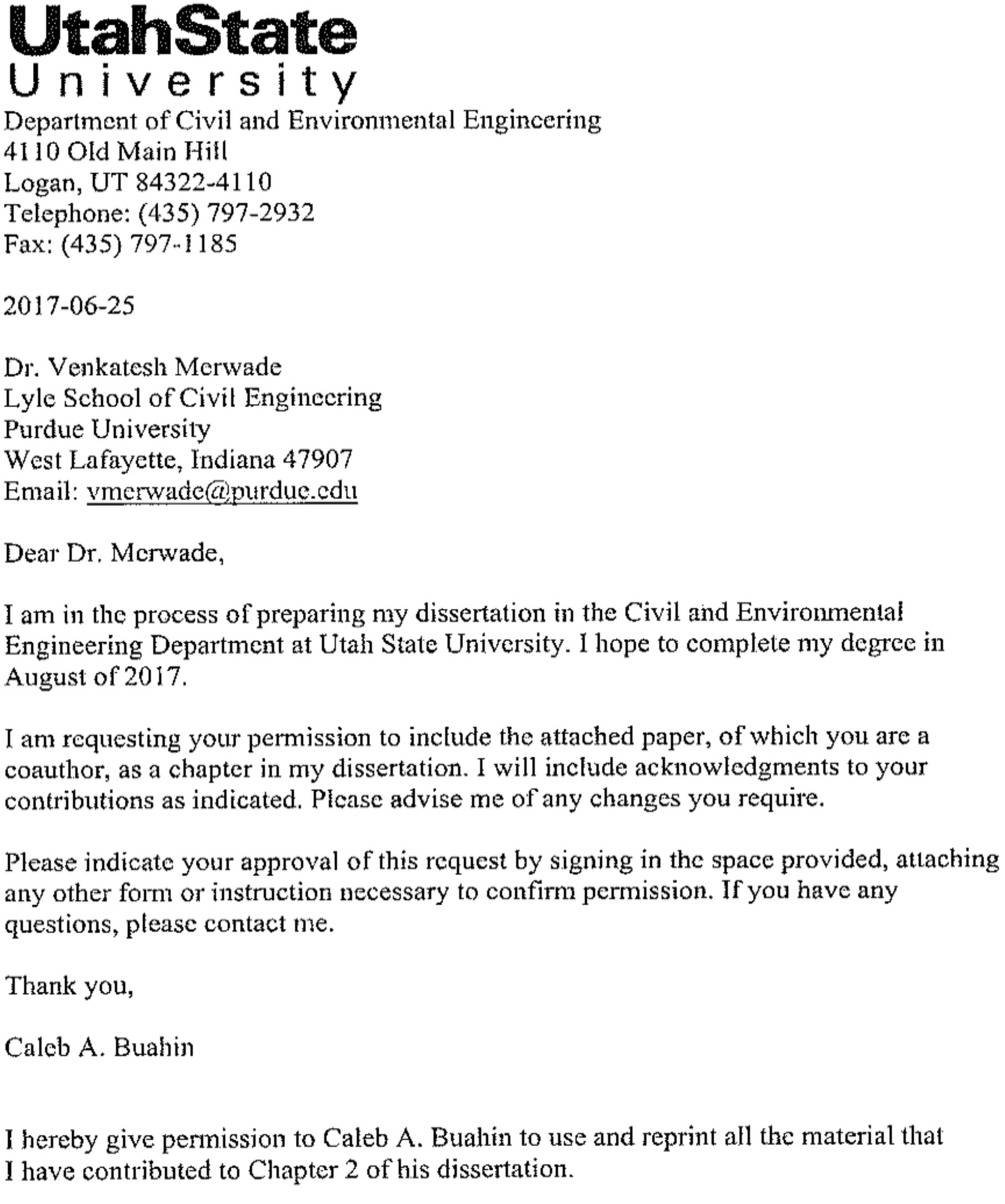
I am requesting your permission to include the attached paper, of which you are a coauthor, as a chapter in my dissertation. I will include acknowledgments to your contributions as indicated. Please advise me of any changes you require.

Please indicate your approval of this request by signing in the space provided, attaching any other form or instruction necessary to confirm permission. If you have any questions, please contact me.

Thank you,

Caleb A. Buahin

I hereby give permission to Caleb A. Buahin to use and reprint all the material that I have contributed to Chapter 2 of his dissertation.



UtahState
University

Department of Civil and Environmental Engineering
4110 Old Main Hill
Logan, UT 84322-4110
Telephone: (435) 797-2932
Fax: (435) 797-1185

2017-06-25

Dr. Venkatesh Merwade
Lyle School of Civil Engineering
Purdue University
West Lafayette, Indiana 47907
Email: vmerwade@purdue.edu

Dear Dr. Merwade,

I am in the process of preparing my dissertation in the Civil and Environmental Engineering Department at Utah State University. I hope to complete my degree in August of 2017.

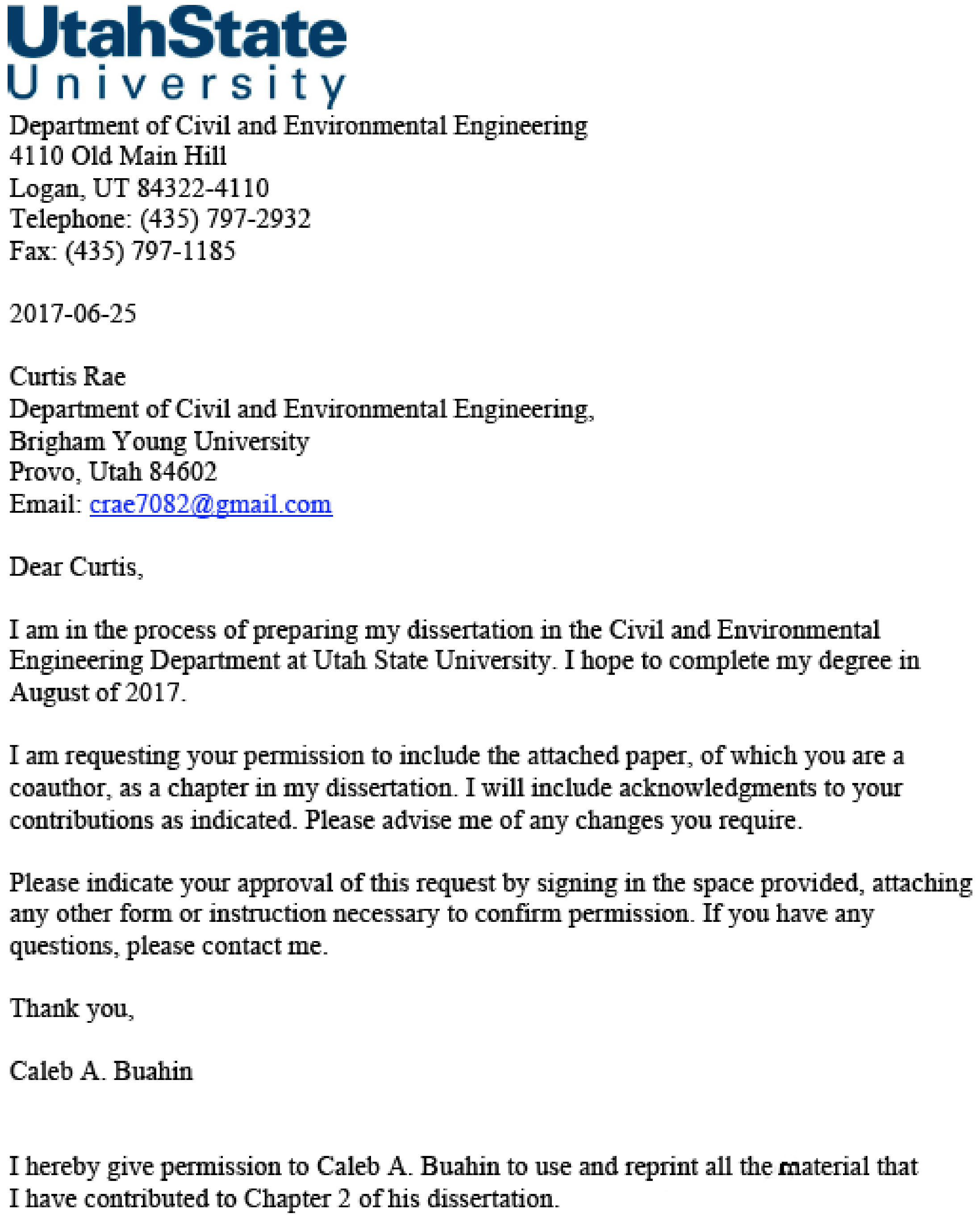
I am requesting your permission to include the attached paper, of which you are a coauthor, as a chapter in my dissertation. I will include acknowledgments to your contributions as indicated. Please advise me of any changes you require.

Please indicate your approval of this request by signing in the space provided, attaching any other form or instruction necessary to confirm permission. If you have any questions, please contact me.

Thank you,

Caleb A. Buahin

I hereby give permission to Caleb A. Buahin to use and reprint all the material that I have contributed to Chapter 2 of his dissertation.



UtahState
University

Department of Civil and Environmental Engineering
4110 Old Main Hill
Logan, UT 84322-4110
Telephone: (435) 797-2932
Fax: (435) 797-1185

2017-06-25

Curtis Rae
Department of Civil and Environmental Engineering,
Brigham Young University
Provo, Utah 84602
Email: crae7082@gmail.com

Dear Curtis,

I am in the process of preparing my dissertation in the Civil and Environmental Engineering Department at Utah State University. I hope to complete my degree in August of 2017.

I am requesting your permission to include the attached paper, of which you are a coauthor, as a chapter in my dissertation. I will include acknowledgments to your contributions as indicated. Please advise me of any changes you require.

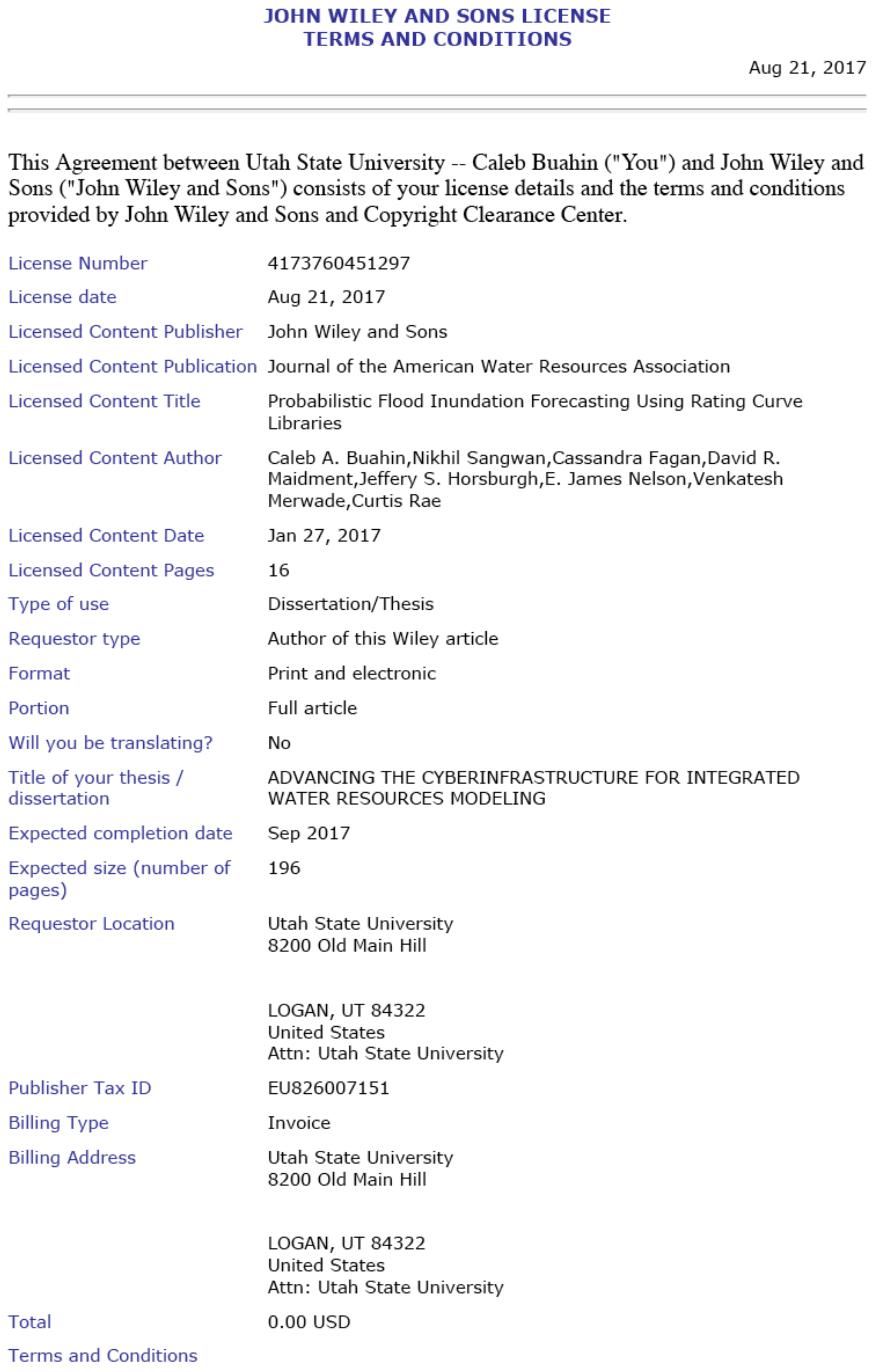
Please indicate your approval of this request by signing in the space provided, attaching any other form or instruction necessary to confirm permission. If you have any questions, please contact me.

Thank you,

Caleb A. Buahin

I hereby give permission to Caleb A. Buahin to use and reprint all the material that I have contributed to Chapter 2 of his dissertation.

Appendix C Permission to Reprint Chapter 2



JOHN WILEY AND SONS LICENSE
TERMS AND CONDITIONS

Aug 21, 2017

This Agreement between Utah State University -- Caleb Buahin ("You") and John Wiley and Sons ("John Wiley and Sons") consists of your license details and the terms and conditions provided by John Wiley and Sons and Copyright Clearance Center.

License Number	4173760451297
License date	Aug 21, 2017
Licensed Content Publisher	John Wiley and Sons
Licensed Content Publication	Journal of the American Water Resources Association
Licensed Content Title	Probabilistic Flood Inundation Forecasting Using Rating Curve Libraries
Licensed Content Author	Caleb A. Buahin,Nikhil Sangwan,Cassandra Fagan,David R. Maidment,Jeffery S. Horsburgh,E. James Nelson,Venkatesh Merwade,Curtis Rae
Licensed Content Date	Jan 27, 2017
Licensed Content Pages	16
Type of use	Dissertation/Thesis
Requestor type	Author of this Wiley article
Format	Print and electronic
Portion	Full article
Will you be translating?	No
Title of your thesis / dissertation	ADVANCING THE CYBERINFRASTRUCTURE FOR INTEGRATED WATER RESOURCES MODELING
Expected completion date	Sep 2017
Expected size (number of pages)	196
Requestor Location	Utah State University 8200 Old Main Hill LOGAN, UT 84322 United States Attn: Utah State University
Publisher Tax ID	EU826007151
Billing Type	Invoice
Billing Address	Utah State University 8200 Old Main Hill LOGAN, UT 84322 United States Attn: Utah State University
Total	0.00 USD
Terms and Conditions	

TERMS AND CONDITIONS

This copyrighted material is owned by or exclusively licensed to John Wiley & Sons, Inc. or one of its group companies (each a"Wiley Company") or handled on behalf of a society with which a Wiley Company has exclusive publishing rights in relation to a particular work (collectively "WILEY"). By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the billing and payment terms and conditions established by the Copyright Clearance Center Inc., ("CCC's Billing and Payment terms and conditions"), at the time that you opened your RightsLink account (these are available at any time at http://myaccount.copyright.com).

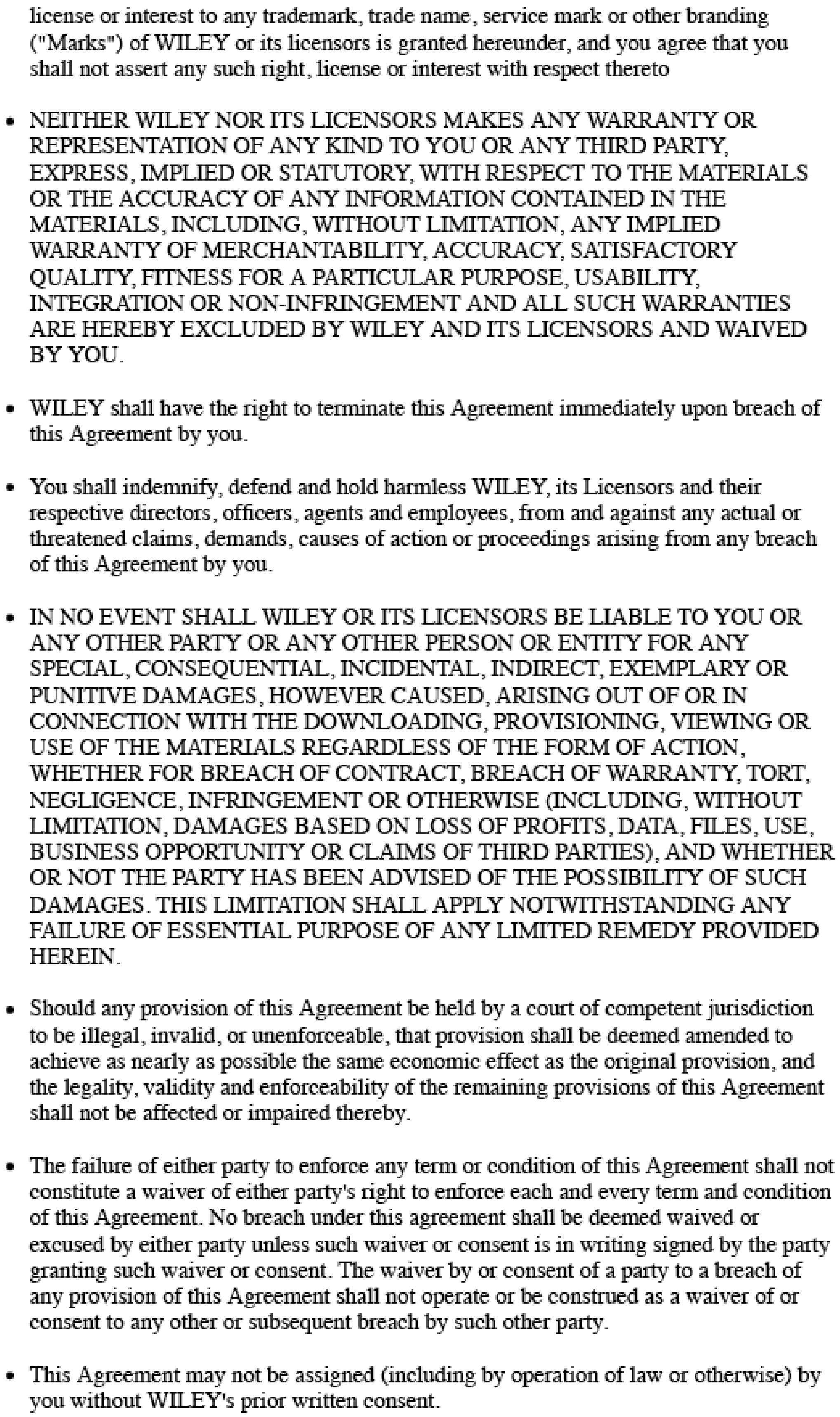
Terms and Conditions

- The materials you have requested permission to reproduce or reuse (the "Wiley Materials") are protected by copyright.

- You are hereby granted a personal, non-exclusive, non-sub licensable (on a stand-alone basis), non-transferable, worldwide, limited license to reproduce the Wiley Materials for the purpose specified in the licensing process. This license, **and any CONTENT (PDF or image file) purchased as part of your order,** is for a one-time use only and limited to any maximum distribution number specified in the license. The first instance of republication or reuse granted by this license must be completed within two years of the date of the grant of this license (although copies prepared before the end date may be distributed thereafter). The Wiley Materials shall not be used in any other manner or for any other purpose, beyond what is granted in the license. Permission is granted subject to an appropriate acknowledgement given to the author, title of the material/book/journal and the publisher. You shall also duplicate the copyright notice that appears in the Wiley publication in your use of the Wiley Material. Permission is also granted on the understanding that nowhere in the text is a previously published source acknowledged for all or part of this Wiley Material. Any third party content is expressly excluded from this permission.

- With respect to the Wiley Materials, all rights are reserved. Except as expressly granted by the terms of the license, no part of the Wiley Materials may be copied, modified, adapted (except for minor reformatting required by the new Publication), translated, reproduced, transferred or distributed, in any form or by any means, and no derivative works may be made based on the Wiley Materials without the prior permission of the respective copyright owner.**For STM Signatory Publishers clearing permission under the terms of the STM Permissions Guidelines only, the terms of the license are extended to include subsequent editions and for editions in other languages, provided such editions are for the work as a whole in situ and does not involve the separate exploitation of the permitted figures or extracts,** You may not alter, remove or suppress in any manner any copyright, trademark or other notices displayed by the Wiley Materials. You may not license, rent, sell, loan, lease, pledge, offer as security, transfer or assign the Wiley Materials on a stand-alone basis, or any of the rights granted to you hereunder to any other person.

- The Wiley Materials and all of the intellectual property rights therein shall at all times remain the exclusive property of John Wiley & Sons Inc, the Wiley Companies, or their respective licensors, and your interest therein is only that of having possession of and the right to reproduce the Wiley Materials pursuant to Section 2 herein during the continuance of this Agreement. You agree that you own no right, title or interest in or to the Wiley Materials or any of the intellectual property rights therein. You shall have no rights hereunder other than the license as provided for above in Section 2. No right,



license or interest to any trademark, trade name, service mark or other branding ("Marks") of WILEY or its licensors is granted hereunder, and you agree that you shall not assert any such right, license or interest with respect thereto

- NEITHER WILEY NOR ITS LICENSORS MAKES ANY WARRANTY OR REPRESENTATION OF ANY KIND TO YOU OR ANY THIRD PARTY, EXPRESS, IMPLIED OR STATUTORY, WITH RESPECT TO THE MATERIALS OR THE ACCURACY OF ANY INFORMATION CONTAINED IN THE MATERIALS, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY, ACCURACY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, USABILITY, INTEGRATION OR NON-INFRINGEMENT AND ALL SUCH WARRANTIES ARE HEREBY EXCLUDED BY WILEY AND ITS LICENSORS AND WAIVED BY YOU.

- WILEY shall have the right to terminate this Agreement immediately upon breach of this Agreement by you.

- You shall indemnify, defend and hold harmless WILEY, its Licensors and their respective directors, officers, agents and employees, from and against any actual or threatened claims, demands, causes of action or proceedings arising from any breach of this Agreement by you.

- IN NO EVENT SHALL WILEY OR ITS LICENSORS BE LIABLE TO YOU OR ANY OTHER PARTY OR ANY OTHER PERSON OR ENTITY FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, INDIRECT, EXEMPLARY OR PUNITIVE DAMAGES, HOWEVER CAUSED, ARISING OUT OF OR IN CONNECTION WITH THE DOWNLOADING, PROVISIONING, VIEWING OR USE OF THE MATERIALS REGARDLESS OF THE FORM OF ACTION, WHETHER FOR BREACH OF CONTRACT, BREACH OF WARRANTY, TORT, NEGLIGENCE, INFRINGEMENT OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA, FILES, USE, BUSINESS OPPORTUNITY OR CLAIMS OF THIRD PARTIES), AND WHETHER OR NOT THE PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.

- Should any provision of this Agreement be held by a court of competent jurisdiction to be illegal, invalid, or unenforceable, that provision shall be deemed amended to achieve as nearly as possible the same economic effect as the original provision, and the legality, validity and enforceability of the remaining provisions of this Agreement shall not be affected or impaired thereby.

- The failure of either party to enforce any term or condition of this Agreement shall not constitute a waiver of either party's right to enforce each and every term and condition of this Agreement. No breach under this agreement shall be deemed waived or excused by either party unless such waiver or consent is in writing signed by the party granting such waiver or consent. The waiver by or consent of a party to a breach of any provision of this Agreement shall not operate or be construed as a waiver of or consent to any other or subsequent breach by such other party.

- This Agreement may not be assigned (including by operation of law or otherwise) by you without WILEY's prior written consent.

- Any fee required for this permission shall be non-refundable after thirty (30) days from receipt by the CCC.

- These terms and conditions together with CCC's Billing and Payment terms and conditions (which are incorporated herein) form the entire agreement between you and WILEY concerning this licensing transaction and (in the absence of fraud) supersedes all prior agreements and representations of the parties, oral or written. This Agreement may not be amended except in writing signed by both parties. This Agreement shall be binding upon and inure to the benefit of the parties' successors, legal representatives, and authorized assigns.

- In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall prevail.

- WILEY expressly reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.

- This Agreement will be void if the Type of Use, Format, Circulation, or Requestor Type was misrepresented during the licensing process.

- This Agreement shall be governed by and construed in accordance with the laws of the State of New York, USA, without regards to such state's conflict of law rules. Any legal action, suit or proceeding arising out of or relating to these Terms and Conditions or the breach thereof shall be instituted in a court of competent jurisdiction in New York County in the State of New York in the United States of America and each party hereby consents and submits to the personal jurisdiction of such court, waives any objection to venue in such court and consents to service of process by registered or certified mail, return receipt requested, at the last known address of such party.

WILEY OPEN ACCESS TERMS AND CONDITIONS

Wiley Publishes Open Access Articles in fully Open Access Journals and in Subscription journals offering Online Open. Although most of the fully Open Access journals publish open access articles under the terms of the Creative Commons Attribution (CC BY) License only, the subscription journals and a few of the Open Access Journals offer a choice of Creative Commons Licenses. The license type is clearly identified on the article.

The Creative Commons Attribution License

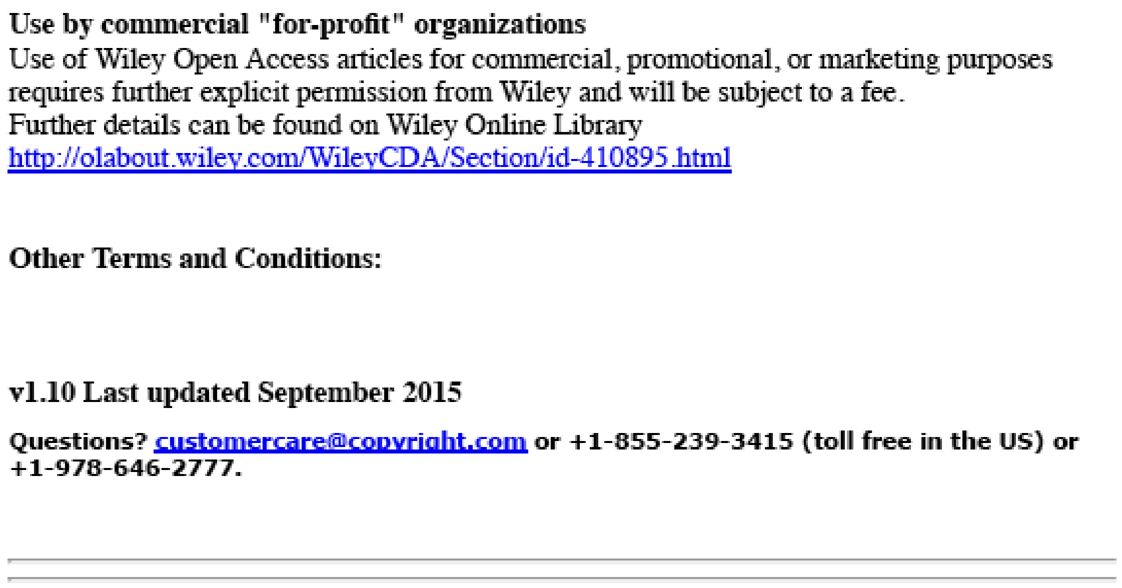
The Creative Commons Attribution License (CC-BY) allows users to copy, distribute and transmit an article, adapt the article and make commercial use of the article. The CC-BY license permits commercial and non-

Creative Commons Attribution Non-Commercial License

The Creative Commons Attribution Non-Commercial (CC-BY-NC)License permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.(see below)

Creative Commons Attribution-Non-Commercial-NoDerivs License

The Creative Commons Attribution Non-Commercial-NoDerivs License (CC-BY-NC-ND) permits use, distribution and reproduction in any medium, provided the original work is properly cited, is not used for commercial purposes and no modifications or adaptations are made. (see below)



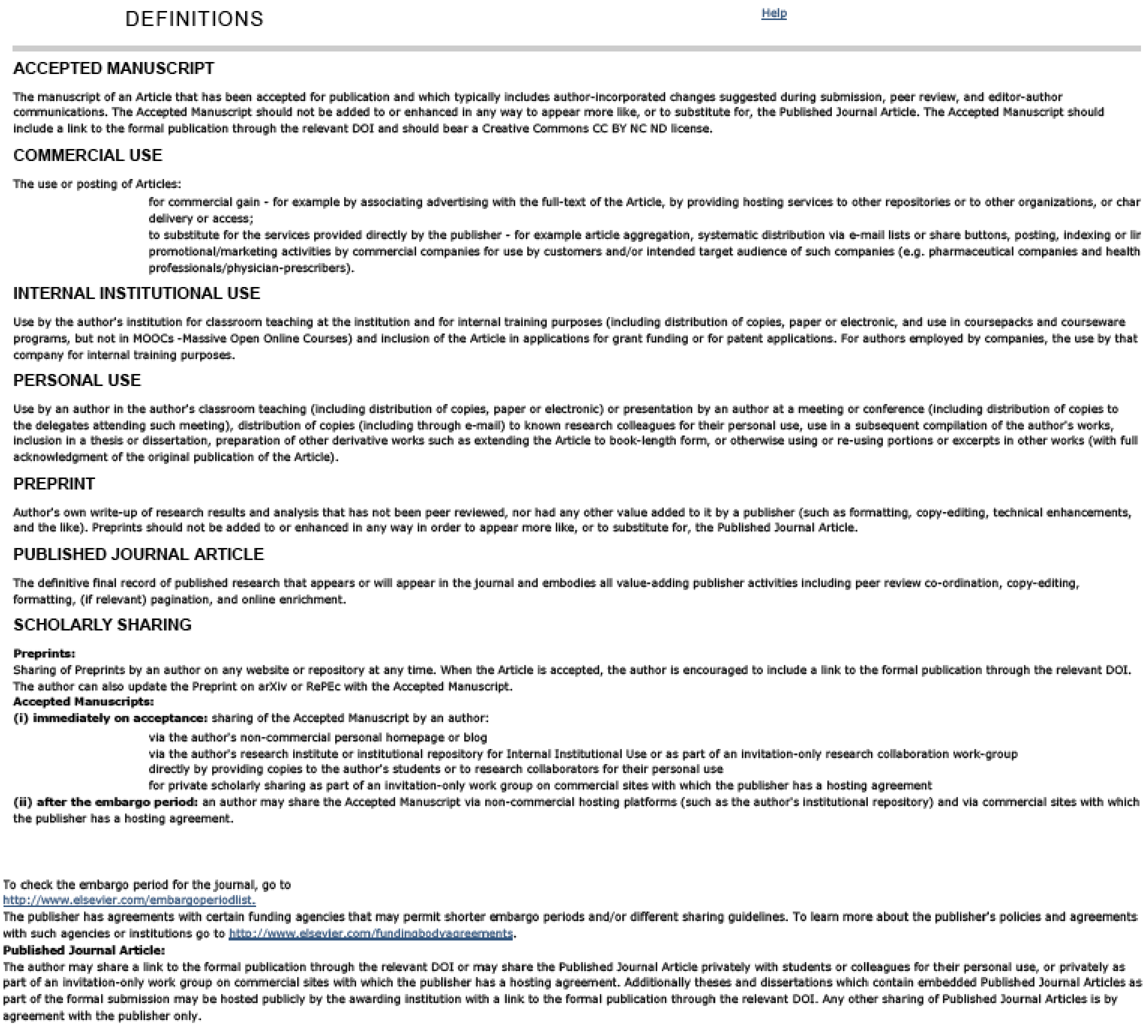
Use by commercial "for-profit" organizations
Use of Wiley Open Access articles for commercial, promotional, or marketing purposes requires further explicit permission from Wiley and will be subject to a fee.
Further details can be found on Wiley Online Library
http://olabout.wiley.com/WileyCDA/Section/id-410895.html

Other Terms and Conditions:

v1.10 Last updated September 2015

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

Appendix D Permission to Reprint Chapter 3



DEFINITIONS

ACCEPTED MANUSCRIPT

The manuscript of an Article that has been accepted for publication and which typically includes author-incorporated changes suggested during submission, peer review, and editor-author communications. The Accepted Manuscript should not be added to or enhanced in any way to appear more like, or to substitute for, the Published Journal Article. The Accepted Manuscript should include a link to the formal publication through the relevant DOI and should bear a Creative Commons CC BY NC ND license.

COMMERCIAL USE

The use or posting of Articles:

for commercial gain - for example by associating advertising with the full-text of the Article, by providing hosting services to other repositories or to other organizations, or char delivery or access;

to substitute for the services provided directly by the publisher - for example article aggregation, systematic distribution via e-mail lists or share buttons, posting, indexing or lir promotional/marketing activities by commercial companies for use by customers and/or intended target audience of such companies (e.g. pharmaceutical companies and health professionals/physician-prescribers).

INTERNAL INSTITUTIONAL USE

Use by the author's institution for classroom teaching at the institution and for internal training purposes (including distribution of copies, paper or electronic, and use in coursepacks and courseware programs, but not in MOOCs -Massive Open Online Courses) and inclusion of the Article in applications for grant funding or for patent applications. For authors employed by companies, the use by that company for internal training purposes.

PERSONAL USE

Use by an author in the author's classroom teaching (including distribution of copies, paper or electronic) or presentation by an author at a meeting or conference (including distribution of copies to the delegates attending such meeting), distribution of copies (including through e-mail) to known research colleagues for their personal use, use in a subsequent compilation of the author's works, inclusion in a thesis or dissertation, preparation of other derivative works such as extending the Article to book-length form, or otherwise using or re-using portions or excerpts in other works (with full acknowledgment of the original publication of the Article).

PREPRINT

Author's own write-up of research results and analysis that has not been peer reviewed, nor had any other value added to it by a publisher (such as formatting, copy-editing, technical enhancements, and the like). Preprints should not be added to or enhanced in any way in order to appear more like, or to substitute for, the Published Journal Article.

PUBLISHED JOURNAL ARTICLE

The definitive final record of published research that appears or will appear in the journal and embodies all value-adding publisher activities including peer review co-ordination, copy-editing, formatting, (if relevant) pagination, and online enrichment.

SCHOLARLY SHARING

Preprints:
Sharing of Preprints by an author on any website or repository at any time. When the Article is accepted, the author is encouraged to include a link to the formal publication through the relevant DOI. The author can also update the Preprint on arXiv or RePEc with the Accepted Manuscript.
Accepted Manuscripts:
(i) immediately on acceptance: sharing of the Accepted Manuscript by an author:

via the author's non-commercial personal homepage or blog
via the author's research institute or institutional repository for Internal Institutional Use or as part of an invitation-only research collaboration work-group
directly by providing copies to the author's students or to research collaborators for their personal use
for private scholarly sharing as part of an invitation-only work group on commercial sites with which the publisher has a hosting agreement

(ii) after the embargo period: an author may share the Accepted Manuscript via non-commercial hosting platforms (such as the author's institutional repository) and via commercial sites with which the publisher has a hosting agreement.

To check the embargo period for the journal, go to
http://www.elsevier.com/embargoperiodlist.
The publisher has agreements with certain funding agencies that may permit shorter embargo periods and/or different sharing guidelines. To learn more about the publisher's policies and agreements with such agencies or institutions go to http://www.elsevier.com/fundingbodyagreements.
Published Journal Article:
The author may share a link to the formal publication through the relevant DOI or may share the Published Journal Article privately with students or colleagues for their personal use, or privately as part of an invitation-only work group on commercial sites with which the publisher has a hosting agreement. Additionally theses and dissertations which contain embedded Published Journal Articles as part of the formal submission may be hosted publicly by the awarding institution with a link to the formal publication through the relevant DOI. Any other sharing of Published Journal Articles is by agreement with the publisher only.

CURRICULUM VITAE

Caleb A. Buahin
Department of Civil and Environmental Engineering
Utah Water Research Laboratory
Utah State University, 8200 Old Main Hill, Logan, UT 84322-8200
Phone: (801) 897-8517
Email: caleb.buahin@gmail.com

Education

Ph.D. Civil and Environmental Engineering December 2017
Utah State University, Logan, Utah
Dissertation: Advancing the Cyberinfrastructure for Integrated Water Resources Modeling.
Advisor: Jeffery S. Horsburgh

M.S. Civil and Environmental Engineering December 2010
Brigham Young University, Provo, Utah
Project: Dynamic Multidimensional Visualization of Water Quality Data for River Networks.
Advisor: James E. Nelson

B.S. Civil and Environmental Engineering 2010
Brigham Young University, Provo, Utah

Professional Experience

Graduate Teaching Assistant 2015 – 2016
Department of Civil and Environmental Engineering, Utah State University, Logan, Utah

Graduate Research Assistant 2013 – 2017
Utah Water Research Laboratory, Department of Civil and Environmental Engineering, Utah State University, Logan, Utah

Project Engineer 2011 – 2013
Environmental Resources Management Inc., Exton, Pennsylvania

Software Engineering Intern 2010
Aquaveo LLC, Provo, Utah

Civil Engineering Intern 2010
United Research Corporation, Salt Lake City, Utah

Graduate Research Assistant 2009 – 2010
The Environmental Modeling Research Lab, Department of Civil and Environmental Engineering, Brigham Young University, Provo, Utah

Research Interests

My research experience and interests are centered generally on the hydroinformatics field with an emphasis on developing and applying models for surface and sub-surface hydrologic, hydraulic, and hydrodynamic modeling studies. My current research is focused on developing and testing standardized modeling interfaces and frameworks that enable earth systems and environmental modelers to couple models in a "plug-and-play" fashion to support more holistic evaluations of human-natural water systems.

Awards

Best Student Paper and Presentation 2016
International Environmental Modeling and Software Society Biennial Congress

Martin Luther King Fellowship 2016
Utah State University, Office of Research and Studies

Graduate Scholar Award 2016
Utah State University, College of Engineering

Visiting Scholar 2015
National Flood Interoperability Experiment Summer Institute, National Water Center, University of Alabama, Tuscaloosa, Alabama

Doctoral Research Fellowship 2013 – 2017
iUTAH and the Utah Water Research Laboratory, Utah State University, Logan, Utah

Graduate Student Finalist 2010
Paul J. Riley Student Conference and Paper Competition, American Water Resources Association, Utah Section

Undergraduate Student Winner 2009
Paul J. Riley Student Conference and Paper Competition, American Water Resources Association, Utah Section

Professional Affiliations

American Society of Civil Engineers.
Environmental Water Research Institute.
American Water Resources Association.

Publications and Presentations

Journal Papers in Print or Press

1. Buahin, C. A., Sangwan, N., Fagan, C., Rae, C., Maidment, D. R., Nelson, J. E., Horsburgh, J. S., Merwade, V. (2017). Probabilistic Flood Inundation Delineation Using a Rating Curve Library Approach, *Journal of the American Water Resources Association (JAWRA)*, doi:10.1111/1752-1688.12500.

2. Buahin, C. A. and J.S. Horsburgh (2015). Evaluating the Simulation Times and Mass Balance Errors of Component-Based Models: An Application of OpenMI 2.0 to an Urban Stormwater System. *Environmental Modelling & Software* 72:92–109. doi:10.1016/j.envsoft.2015.07.003.
3. Hale, R.L., A. Armstrong, M.A. Baker, S. Bedingfield, D. Betts, C. A. Buahin, M. Buchert, T. Crowl, R.R. Dupont, J.R. Ehleringer, J. Endter-Wada, C. Flint, J. Grant, S. Hinners, J.S. Horsburgh, D. Jackson-Smith, A.S. Jones, C. Licon, S.E. Null, A. Odame, D.E. Pataki, D. Rosenberg, M. Runburg, P. Stoker, and C. Strong (2015). iSAW: Integrating Structure, Actors, and Water to Study Socio-Hydro-Ecological Systems. *Earth's Future*. doi:10.1002/2014EF000295.
4. Williams, G.P., O. Obregon, E.J. Nelson, W. Miller, M.B. Borup, and C. A. Buahin (2014). Sensitivity of Water Quality Indicators in a Large Tropical Reservoir to Selected Climate and Land-Use Changes. *Lakes & Reservoirs: Research & Management* 19:293–305. doi:10.1111/lre.12079.

Journal Papers in Preparation and Review

1. Buahin, C.A., J.S. Horsburgh (2017). HydroCouple: A Component-Based Modeling Framework for Integrated Water Resources Modeling. *Environmental Modelling & Software*

Conference Proceedings Papers

1. Buahin, C. A., Horsburgh, J. S. **(**2016). From OpenMI to HydroCouple: Advancing OpenMI to Support Experimental Simulations and Standard Geospatial Datasets, In: Proceedings of the 8th International Congress on Environmental Modelling & Software, 11-14 July, Toulouse, France.
2. Buahin, C. A., Nelson, E. J., Obregon, O., and Williams, G. P. (2011). Dynamic Multidimensional Visualization for Water Quality Data in Rivers. World Environmental and Water Resources Congress 2011, American Society of Civil Engineers, 4811–4819. http://ascelibrary.org/doi/abs/10.1061/41173%28414%29499.
3. Buahin, C. A., R. Hila, T. Rabadi, O. Obregon, R. Chilton, A. Childers, G. Williams, and E.J. Nelson (2010). ArcGIS Tools for Storing and Analyzing Reservoir Vertical Profile Data. AWRA 2010 Spring Specialty Conference. Orlando, FL.
4. Buahin, C.A. (2010). "Spatial Interpolation Techniques for Dynamic Isopleth Map Generation in Assessing Water Quality in Rivers." J. Paul Riley Student Conference and Paper Competition, AWRA Utah Section.

Conference Presentations, Posters, and Abstracts

1. Buahin, C.A. and J.S. Horsburgh (2017). Parallel Optimization Simulations Using the HydroCouple Component-Based Modeling Framework. Fourth Workshop on Coupling Technologies for Earth System Models (CW2017). Princeton, NJ. https://www.earthsystemcog.org/projects/cw2017/abstracts#buahin.
2. Buahin, C.A. and J.S. Horsburgh (2016). From OpenMI to HydroCouple: Advancing OpenMI to Support Experimental Simulations and Standard Geospatial Datasets. Environmental Modelling and Software for Supporting a Sustainable Future.

Toulouse, France, pp. 153–160. http://scholarsarchive.byu.edu/iemssconference/2016/Stream-A/11.

3. Buahin, C. A. and J. Horsburgh (2015). Computational Penalties of Component Based Models: An Urban Stormwater Component-Based Modeling Application Using OpenMI. Spring Runoff Conference. Utah State University. http://digitalcommons.usu.edu/runoff/2015/2015Posters/38.

Teaching Experience

Courses

Geographic Information Systems for Civil Engineers 2015– 2016
Utah State University, Logan, Utah
Undergraduate and Graduate Level Course in Civil and Environmental Engineering

Reviewer for the Following Journals

Environmental Modelling & Software

Journal of the American Water Resources Association

Lakes & Reservoirs: Research and Management