2008-11-19

Foundational Data Repository for Numeric Engine Validation

Jason Michael Hollingsworth
Brigham Young University - Provo

FOUNDATIONAL DATA REPOSITORY FOR NUMERIC

ENGINE VALIDATION

by

Jason M. Hollingsworth

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Civil and Environmental Engineering

Brigham Young University

December 2008

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Jason M. Hollingsworth

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	E. James Nelson, Chair

Date	Alan K. Zundel

Date	Rollin H. Hotchkiss

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Jason M. Hollingsworth in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

_____   _____
Date   E. James Nelson
                                          Chair, Graduate Committee

Accepted for the Department

                                          _____
                                          Steven E. Benzley
                                          Department Chair

Accepted for the College

                                          _____
                                          Alan R. Parkinson
                                          Dean, Ira A. Fulton College of Engineering
                                          and Technology

ABSTRACT

FOUNDATIONAL DATA REPOSITORY FOR NUMERIC

ENGINE VALIDATION

Jason M. Hollingsworth

Department of Civil and Environmental Engineering

Master of Science

Many different numeric models have been created to address a variety of hydraulic and hydrologic engineering applications. Each utilizes formulations and numeric methods to represent processes such as contaminant transport, coastal circulation, and watershed runoff. Although one process may be adequately represented by a model, this does not guarantee that another process will be represented even if that process is similar. For example, a model that computes subcritical flow does not necessarily compute supercritical flow. Selecting an appropriate numeric model for a situation is a prerequisite to obtaining accurate results.

Current policies and resources do not provide adequate guidance in the model selection process. Available resources range from approved lists to guidelines for performing calculations to technical documentation of candidate numeric models. Many of these resources are available only from the developers of the numeric models. They

focus on strengths with little or no mention of weaknesses or limitations. For this reason, engineers must make a selection based on publicity and/or familiarity rather than capability, often resulting in inappropriate application, frustration, and/or incorrect results.

A comprehensive selection tool to aid engineers needs to test model capabilities by comparing model output with analytical solutions, laboratory tests, and physical case studies. The first step in building such a tool involves gathering and categorizing robust data the can be used for such model comparisons. A repository has been designed for this purpose, created, and made available to the engineering community. This repository can be found at http://verification.aquaveo.com. This allows engineers and regulators to store studies with assigned characteristics, as well as search and access studies based on a desired set of characteristics. Studies with characteristics similar to a desired project can help identify appropriate numeric models.

ACKNOWLEDGMENTS

TABLE OF CONTENTS

x

LIST OF TABLES

LIST OF FIGURES

1 Introduction

Numeric hydraulic engines have been coded to solve a range of engineering studies ranging from groundwater to watershed to coastal issues. No single numeric engine addresses all aspects of hydrologic engineering. Several numeric engines may produce reasonably accurate results. Selecting an appropriate engine for a situation helps to ensure better results for a project. This report discusses a framework to assist engineers and regulators in the numeric engine selection process.

1.1 Definitions

In the modeling world, it is common to refer to the numeric engine performing the computations as a "model" and also the input data as a "model". To avoid confusion in this paper, the computation engine will be referred to as an "engine" while the input data will be referred to as a "model".

1.2 Background

Numeric engines provide engineers assigned to perform analysis and design of complex engineering problems a powerful toolset. Each numeric engine is a distinct tool designed for a specific purpose. The number of numeric engines available increases continually. This makes the task of selecting the best engine for a study difficult and the

possibility of selecting an inappropriate tool more likely. Appropriate engines produce results similar to those observed in the field. The best engine for the study produces the closest results to the observations.

The ability to calibrate or verify a model with real world situations is a key, and often overlooked, step in any modeling process. The challenge facing the user of the engine commences with selecting an engine that performs the necessary computations and also reproduces the observed conditions.

1.2.1 Numeric Engines

Numeric engines exist to compute just about anything. In the world of water, engines have been created to compute, among other things:

- Hydraulics (flow directions and water depth)
- Wave state (height or amplitude, period, and direction)
- Hydrologic routing (reservoir and riverine)
- Transport

Each of these categories contains multiple complexities and specializations. For example, hydraulic computations vary from:

- Analytical equations such as the Manning's equation
- Lumped parameter 1D engines
- Riverine channel/floodplain interaction
- Coastal zone, bay, estuary, and inlet engines
- Regional to global ocean engines
- Computational fluid dynamics (CFD) engines

2

Each of these engines attempt to represent reality. Their formulations come from empirical or theoretical relationships and include inherent assumptions. Simple analytical solutions solve the problem directly. Results from an analytical solution can be obtained quickly, but, the restricting assumptions often prohibit accurate results for all but the simplest problems. These solutions produce quick estimates, but should not be applied for complex analysis/design projects. Lumped parameter engines assume that flow proceeds in one direction. Other dimensions are assumed constant at any given position along the dominating dimension. Two-dimensional engines assume the vertical or horizontal flow components are negligible. Three-dimensional engines take all components (longitudinal, lateral, and vertical) into account while performing calculations, but still include limiting assumptions and numerics.

Numeric engines perform calculations at finite locations in space. The discretization of the model determines where to perform computations and output solutions. Types of discretizations in numerical engines include:

- Finite-element

- Finite-difference

- Finite-volume

After choosing the engine type, complexity, and discretization, the engineer will generally still have a plentiful selection of engines. Each of these engines will produce varied results. Two engines may produce nearly identical or dramatically different results. For this reason it is also critical to understand the limitations of each numerical engine. Only with this understanding can an engineer select an appropriate engine to use for a study.

1.2.2   Trusting Model Results

An engineer should always exercise caution before accepting the results of a numeric model as valid. All results should be verified. In ideal situations, this process, sometimes called calibration or validation, increases confidence that the engine produces useable results. This step of the modeling process is often overlooked or minimized, as obtaining real world data can be time consuming, dangerous, expensive, and sometimes impossible.

"Calibrating a model", involves comparing model results with real world data, and then making small changes to the input parameters of the model to make the results more accurate. It is important to understand that these changes should always stay within the acceptable range for the data type. These changes should never be made to physically measured values such as bathymetry/topography, flow rates, or water surface elevations. The calibration process should generally be performed one variable at a time. This ensures that the changes being made to variables do not interact. The changes made to variables should remain consistent with physical conditions. When changing input variables, it is important to make changes that are justifiable. For example, changing the roughness value for a section of channel when the bed material and surface (e.g. plane bed, ripples, dunes, etc) do not change would be an unjustifiable change. As input values are changed, the output of the model will also change.

The sensitivity of an engine to parameters can be measured by changing the input parameters and observing the change in the outputs. Understanding the sensitivity of an engine to various parameters can help speed up the calibration process. A sensitivity analysis can help determine which input parameters have the largest influence on the

output of the engine. Sensitivity can be performed on any input, including bathymetry/topography. A sensitivity analysis can help determine the amount of change that can occur before undesirable effects occur.

When performing a sensitivity analysis it is important to remember that the output from the engine reflects the impact of changing the input parameters. To perform this type of analysis, a baseline must be established. Changes and resulting differences will determine the sensitivity of an engine to a certain parameter, but this in no way reflects performance. The only true way to determine the performance of an engine is its ability to match observed conditions. In the absence of calibration data, a sensitivity analysis may help bracket input values.

1.3 NCHRP Project 24-24

In July of 2004, National Cooperative Highway Research Program (NCHRP) Project 24-24 was commissioned as an attempt to provide tools to aid engineers in the numeric engine selection process. According to the Transportation Research Board (TRB), eight tasks were to be completed in two phases. (National Academy of Sciences [NAS] 2008)

> **PHASE I** (1.) Conduct a literature review to identify commonly used one-dimensional and two-dimensional numeric modeling software that can be used in the analysis of bridge openings in riverine and tidal systems. Identify available data sets from actual bridge sites for use in analyzing the different modeling software. The data sets should include not only input data for the modeling software but also, if available, observed event data for use in validating the model results. Identify and characterize site conditions and design requirements that can affect the selection of a hydraulic model. (2.) Segregate the Task 1 data sets, site conditions, and design requirements into categories, including but not limited to embankment skew, skewed and/or complex pier configurations, complex floodplain geometry, curvilinear flow, valley slope, multiple openings, and

variable flow rates. Additionally, categorize identified modeling software based on numeric approach, capabilities, and ease of use. (3.) Using theoretical data and commonly used numeric modeling software identified in Task 1, develop one- and two-dimensional conceptual numeric models to evaluate the effects of site conditions including but not limited to embankment skew, skewed and/or complex pier configurations, complex floodplain geometry, curvilinear flow, valley slope, multiple openings, and variable flow rates. (4.) Based on the results of Tasks 1, 2 and 3, develop a preliminary decision analysis tool (e.g. decision tree) for use by practicing engineers in selecting the most appropriate numeric modeling software for use in a given situation and design stage. (5.) Within 10 months of contract award, submit an interim report documenting the information developed in Tasks 1 through 4. The interim report shall contain, as a separate appendix, an updated work plan for completing Phase II of the research. Meet with the NCHRP panel to discuss the interim report, proposed data sets for use in Phase II, and the updated work plan. Work on Phase II will not begin until the interim report and updated work plan are approved by the NCHRP.

PHASE II (6.) Using the Phase I data sets agreed on during the interim meeting, validate the decision analysis tool for selecting the most appropriate numeric modeling software for use in a given situation and design stage. Revise and finalize the decision analysis tool as necessary based on the validation results. (7.) Develop guidelines to assist hydraulic engineers in applying the decision analysis tool. (8.) Submit a final report documenting the entire research effort. The report shall include an appendix that fully describes the decision analysis tool and guidelines and provides illustrative examples for use of each. (NAS 2008)

The defined objective of the NCHRP research is the key element examined here: "develop a decision analysis tool and guidelines to assist hydraulic engineers in selecting the most appropriate numeric modeling software" (NAS 2008). The NCHRP was primarily concerned with riverine and tidal bridges.

The desired outcome of the project was to obtain a better understanding of numeric engines available for analyzing bridge crossings. The first step involved surveying practicing engineers to determine which engines were used and if case studies were available for testing the engines. With that information, the engines were to be

tested in order to develop a tool to aid engineers in the process of selecting a numeric engine.

The end result of the project was a decision matrix which could be filled out and used to select whether to use a one-dimensional or two-dimensional engine. The decision matrix still required the engineer to fill out the effectiveness of the one- and two-dimensional engines. This information was not made available in the report and thus the tool could not be used effectively (Chapter 2 discusses this report in more detail). The objective of the study was a large and worthwhile undertaking. The time constraints placed on the project (10 months for the first phase), made this a nearly impossible task. As such the outcome did not result in a completed objective and verifiable selection tool, but provided a basic guideline for future work.

1.4   Project Scope

This project attempts to begin an ongoing task to accomplish the objective set forth by the TRB. In order to effectively develop a tool to aid engineers in the selection of the most appropriate numeric engine, it is necessary to determine the capabilities and limitations of the specific numeric engines. As discussed previously, the best method of measuring engine performance is its ability to be calibrated and match observed conditions. Calibrating a model requires case studies with calibration data. These cases could be real world, laboratory, or analytical. Storing the case studies in a repository would ensure the cases could be easily accessed. This requirement was the primary focus of my research. The main objectives of my research were:

1.   Research available tools and resources for building a case study repository

2. Select and use tools to develop a repository which could be accessed by the public

3. Contact companies and individuals to obtain case studies for the repository

4. Begin populating the repository and illustrate its applicability

The above objectives focus on a particular portion of the NCHRP study, namely gathering and categorizing data. Supporting the above objectives required that the repository perform a set of specific functions. The repository needed to be able to:

- Store site characteristics (i.e. supercritical flow conditions, flow contraction, super elevation around a bend) included in the studies

- Store whether the site was a real site, a laboratory study, or an analytical case

- Allow users to search the repository for cases that met certain criteria

- Allow users to search for cases that are similar to a given site

- Store and search different types of sites (i.e. riverine, costal, watershed)

Once the repository has been populated, it may be used in two modes. The case studies in the repository may be used to test the capabilities and accuracy of a single numeric engine. This would in essence be running the numeric engine though a test suite. Another use involves searching the repository for similar sites and evaluating various engines that work well for the specified conditions. Both modes use the compiled case study repository to more accurately determine which numeric engines perform best for given situations, by determining which engines can match the observed data within reasonable tolerance.

2 Existing Policies and Resources

Current policies and resources available to engineers regarding numeric engines do not provide much guidance for selecting an engine. Available resources range from lists of accepted engines, such as a Federal Emergency Management Agency (FEMA) approved list, to guidelines for performing calculations, similar to those found in publications such as the Hydraulic Engineering Circulars, which are discussed in Section 2.2. Today's engineers must choose numeric engines based simply on familiarity rather than the performance of the engine resulting in hit or miss success. This often results in frustration to the engineer, inappropriate engine selection, and incorrect or less correct model results. By using existing resources and ideas, a better selection tool for numeric engines can be constructed for use by engineers.

This section will discuss some of the currently available guidelines/resources to engineers. Each of these provides some help to engineers when selecting an engine, but no complete tool or method exists. Using ideas from each resource, a better, more complete selection tool can be built.

2.1   Federal Emergency Management Agency

FEMA maintains a list of accepted numeric engines for use in flood insurance map revision. This list has been compiled in accordance with Title 44 of the Code of Federal Regulations (CFR) 65.6(a)(6), which reads:

> (6) Any computer program used to perform hydrologic or hydraulic analyses in support of a flood insurance map revision must meet all of the following criteria:
> (i) It must have been reviewed and accepted by a governmental agency responsible for the implementation of programs for flood control and/or the regulation of flood plain lands. For computer programs adopted by non-Federal agencies, certification by a responsible agency official must be provided which states that the program has been reviewed, tested, and accepted by that agency for purposes of design of flood control structures or flood plain land use regulation.
> (ii) It must be well-documented including source codes and user's manuals.
> (iii) It must be available to FEMA and all present and future parties impacted by flood insurance mapping developed or amended through the use of the program. For programs not generally available from a Federal agency, the source code and user's manuals must be sent to FEMA free of charge, with fully-documented permission from the owner that FEMA may release the code and user's manuals to such impacted parties. (U.S. Government Printing Office 2008)

This policy specifically applies to flood insurance studies, but the policy contains multiple shortcomings. The policy states that any numeric engine which is used must be reviewed and accepted by a government agency responsible for implementing programs for flood control and/or regulation of flood plain studies. As these agencies generally develop their own numeric engines for such studies, there is generally no motivation for the agencies to consider other numeric engines. FEMA has stated that they "would prefer not to [review and test engines] in the future, for this is a time-consuming and expensive procedure" (Buckley 1999). This statement would be true for any of the agencies which

have been approved to test and accept numeric engines, as such this would be a great deterrent to use engines developed by an outside entity.

The policy leaves the testing and review procedure to the discretion of the government agency. There is no motivation to thoroughly review an engine. There is no standard in engine testing procedure or performance. The definition of adequate documentation does not exist.

The FEMA list is simply one of numeric engines in use by various government agencies. Engineers are given no guidance as to the performance of the engines, or which engine should be used for specific types of applications. This may result in the selection of an accepted, but inappropriate engine. For example, both RMA2 and FESWMS are listed as accepted engines. An engineer may look at the list and decide to use RMA2 for a study because of familiarity with the engine. If the study contains areas of supercritical flow, the engineer should have selected FESWMS, as RMA2 does not handle supercritical flow regimes. The FEMA policy, however, gives no guidance as to the capabilities and limitations of the various engines.

2.2   Hydraulic Engineering Circulars

The Hydraulic Engineering Circulars are a series of publications by the Federal Highways Administration (FHWA) which give guidelines for design and evaluation of hydraulic structures. These circulars provide guidelines accepted by FHWA for various practices and are used for many engineering projects. Some of the topics discussed in the circulars include:

- Evaluating scour at bridges

- Drainage of highway pavements

- Urban drainage design

- Hydraulic design of improved inlets for culverts

Software programs such as the Watershed Modeling System (Aquaveo, LLC. 2008) sometimes implement these procedures in order to aid in the design and analysis of various projects.

The guidelines presented are based on many observations and are meant to give engineers easier ways to achieve answers to otherwise difficult problems. The results may not always be correct. However, when faced with a problem that does not have an exact solution, empirical equations offer a better solution than simply estimating.

While the circulars generally do not provide case study data, the methods presented can still be utilized in the engine verification process. The formulae and methods contained in the reports can be used to build analytical case studies. These case studies, while not real observed data, can expand the studies available for testing.

2.3   Bridge Backwater Atlases

In the 1960's and 1970's the United States Geological Survey (USGS) in cooperation with the Federal Highway Administration Department of Transportation, the Alabama State Highway Department, the Louisiana Department of Transportation and Development, and the Mississippi State Highway Department collected backwater data where wide vegetated flood plains were crossed by highway embankments. Data were collected for 35 different floods over 22 sites. (U.S. Geological Survey [USGS] n.d.)

These data were collected in order to aid in the development of numeric engines for predicting the response to highway crossings at streams. The results from the study showed that methods in use in 1970 were inaccurate (USGS n.d.). These case studies would provide an excellent tool for verification of currently used numeric engines.

The atlases are available on the Mississippi USGS website as a series of images. These images contain the information from the original reports of the study. Chapter 4 contains an example of an atlas and discusses the content in further detail. In order to use the studies each image needs to be georeferenced and digitized. This process can take hours per atlas.

2.4   National Cooperative Highway Research Program Project 24-24

As discussed in Chapter 1, the objective of NCHRP 24-24 was to design a tool to aid engineers in selecting a numeric engine to use for a case study. Similar to the bridge backwater atlases, this study focused on bridge crossings. The final report for the project (Sheppard 2007) explained the procedures used.

The first task involved determining the engines used by engineers and gather case studies. This was accomplished by conducting a survey among the practicing engineers at the Federal Highways Administration and state Departments of Transportation. The survey showed the numerical engines being used were:

Table 2-1: Commonly Used Numeric Engines for Bridge Sites

One-dimensional	Two-dimensional
HEC-2	RMA2
UNET	FESWMS
WSPRO	ADCIRC
HEC-RAS	MIKE21
Ad-ICPR	Delft-FLS
SWMM	
MIKE11	

The study selected two engines on which to perform the various tests. The most commonly used engine of each dimensional category was selected. According to the survey results, these were HEC-RAS and FESWMS. The assumption made was that one engine from each of the categories, one- and two-dimensions, represented the complete population of engines. This assumption fails to take into account known limitations with the engines, e.g., RMA2 does not support supercritical flow regimes.

The other reason for the survey involved gathering data sets for numeric engine testing. "53% of the respondents indicated they did in fact have data sets available" (Sheppard 2007) according to the survey. However, these data sets were not used in the characteristic definition, testing, or verification processes. The reason for not using these data sets was not clearly stated in the report.

In order to determine the performance of the engines, conceptual cases were constructed for 10 different site characteristics commonly found at bridge crossings. A comparative analysis, between the one- and two-dimensional results, was performed for each of these characteristics. The characteristics studied include:

- Multiple openings

- Bridges located on river bends

- Bridges near confluences

- Bridges with significant constrictions

- Overtopping flow

- Embankment skew

- Bridges over meandering rivers

- Bridges with asymmetric floodplains

- Bridges with large piers/high blockage

- Tidal hydraulics

When performing the comparative analysis, the two-dimensional engine was assumed to always be correct. There were no calibration/verification data for the various cases. The results presented show the difference between the one-dimensional engine results and two-dimensional engine results. The models were not calibrated to observed data, nor were any attempts made to make the model results match. In essence, the results showed how the results from HEC-RAS and FESWMS differed with the same inputs. This information results in very little useful data, as any two engines will give different results due to inherent assumptions and underlying formulations.

From the results, a decision matrix was designed in order to aid engineers in selecting whether to use a one- or two-dimensional engine. One of the main, and highly flawed, assumptions used for producing the decision matrix, is that all one-dimensional engines produce results similar to HEC-RAS and all two-dimensional engines produce results similar to FESWMS.

An example decision matrix for a site containing a bridge near a confluence with multiple openings, overtopping flow, embankment skew, and asymmetric floodplains can

be seen in Table 2-2. The user would fill in the weight column with the importance of each site condition. This value should be determined based on the results presented in the final report of the NCHRP 24-24 study and observations made by the engineer. The score column represents the engine's performance in each of the criteria. Once again this judgment must be made by the engineer based on results presented in the report. The report, for reasons unstated, did not provide default values for the engine performance. This means that two individuals could fill in the same set of engine performance scores with different values. The shortcoming of this being that the two-dimensional engine will always receive a higher score as it was used as the baseline. It would seem that this being the case, a two-dimensional engine would always be selected. However, notice the "Other Considerations" section. This provides the means to change the decision. This section includes items such as the engineer's experience, time, and data availability. In the report, the one-dimensional engine always scored higher in these categories, despite the fact that the two-dimensional models always took less time to construct. The justification was that one-dimensional models are easier to set up, take less modeling experience, and require less data. In essence, the decision matrix determines whether the inaccuracy of the one-dimensional engine is outweighed by the inexperience of the engineer with the two-dimensional engine.

Once a decision is made on whether to use a one- or two-dimensional engine, the engineer still has a suite of engines to select from, and this tool gives no guidance in making such a decision. The outlined process for selecting a numeric engine is far too subjective. The decisions must be made by each engineer based on observations from the report and prior knowledge about the engines. As the results presented in the study are

not based on real observations, the conclusions drawn about engine performance are inaccurate. (Sheppard 2007)

Table 2-2: Example Decision Matrix Produce by NCHRP Project 24-24

Design Criteria	Weight	One-dimensional model		Two-dimensional model	
		Score	Weight x Score	Score	Weight x Score
Site Conditions	(1-10)				
Multiple openings					
Bridge near confluences					
Overtopping flow					
Embankment skew					
Bridges with asymmetric floodplains					
Other Considerations	(1-10)				
Modeler Experience					
Scheduling					
Data Availability					
Totals (Sum of Weight x Score)					

2.5 "Verification and Validation of 3D Free Surface Flow Models"

Dr. Sam Wang of the University of Mississippi has been working on a book titled "Verification and Validation of 3D Free Surface Flow Models". The book promises to contain methods and data for verifying numeric engines. The data which were collected for use in the study cannot be released until the book is published. This project has been in the works for many years and will hopefully prove beneficial in the engine verification and selection process. (Sam Wang, personal communication, October 8, 2008)

2.6   Review Summary

Each of these resources has the potential to aid engineers in selecting a numerical engine for a study. However, major gaps exist in providing a real selection tool.

The FEMA policy provides the idea of maintaining a list of accepted numeric engines from which to choose. This list includes only engines that have been, or at least claimed to have been, reviewed and tested. The testing process could be improved upon by making the process uniform. In other words, every engine on the list would have been required to pass a suite of case studies within reasonable tolerance. Additionally, the list should include a way to indicate suitability for specific applications.

The Hydraulic Engineering Circulars suggests guidelines for acceptable analytical solutions. As it is not always possible to collect verification data for all cases, these circulars would aid in the process of designing analytical case studies with calibration data. This would enable the testing of a wider range of case studies.

The hydrologic backwater atlases provide 35 case studies over 22 different sites. As this data was collected to aid in the development of more accurate numeric engines, it makes it ideal for verification case studies. In its current state, however, this data can not easily be used for its intended purpose.

The NCHRP Project 24-24 contributes an initial method for the development of a selection tool. The tool however needs to include more components and be more developed. It must incorporate calibration of model results against observed data. Only in this way can the performance of a numeric engine be determined. Each one- and two-dimensional engine must be tested, as each will perform differently. This will help

determine engine strengths and allow the selection to be made on an application specific basis.

All these resources attempt to aid in the engine selection process, but none provides real world assistance. The need for a better selection tool for numeric engines still exists. This tool would be used by engineers to select an engine based on observed site characteristics.

2.7   Developing a New Selection Tool

A satisfactory tool to aid in the selection of an appropriate numeric engine does not exist. The development and use of a new selection tool should proceed as follows:

1. Collect and categorize case studies with calibration data

2. Build and calibrate models for various numeric engines

3. Determine performance of engine by comparing results to observed data and calibration parameters

4. Compile numeric engine results based on site characteristics

5. Use selection tool

    a) Engineer selects site characteristics present

    b) Numeric engine results are queried and returned

    c) Based on results suggestion is made for numeric engine(s) to use

    d) Engineer reviews results and makes decision

This new tool will select engines based on results for case studies with calibration data. The performance of an engine will be judged by its ability to match the calibration data while keeping calibration parameters within a reasonable range (discussed in

Chapter 1) for each application. These results will be compiled into a database which will be queried by the decision tool. When selecting an engine, the engineer will input the site characteristics present for the project. The tool will return how the tested numeric engines performed when case studies containing those characteristics were run. Based on these results, the selection tool can suggest one or more appropriate engines. The engineer still makes the final selection.

The case studies contained in the repository can be used in the future to standardize the review and testing procedure suggested by FEMA. This will more easily allow new engines to be tested and verified. The results can be used to evaluate how new engines could be applied in future studies.

The effectiveness of this tool pivots on the ability to compile meaningful case studies to test the capabilities of the numeric engines. I have designed a repository which allows engineers to store and share case studies, which can be used to test the performance of numeric engines. The end success of the selection tool will ultimately be determined by the involvement of the engineering community in filling the repository with case studies.

3   Verification Repository

The first step in building the selection tool involves gathering and categorizing case studies with calibration data. This project creates a repository designed for this purpose, and offers it to the public for continued enhancement. The process of adding case studies to the repository must be a continual process. As the development of new engines continues, case studies testing the capabilities of these engines must be added to the repository. This chapter discusses the methods used to design and build the repository, which can be found at http://verification.aquaveo.com.

3.1   Sponsors

Aquaveo sponsored the design and development of the repository tool. Currently they also sponsor the maintenance and data storage fees. In the future, agencies such as the U.S. Army Corp of Engineers and FEMA may sponsor the repository. Many of the numeric engines in use are developed by the Army Corp and the repository would prove a valuable tool for verifying the engines. FEMA maintains a list of accepted engines for flood studies. This repository would help in the verification process of those engines. With the completion of the selection tool, the list of acceptable models would be returned by the tool.

3.2   Design Requirements

This section discusses the design requirements for a repository to allow engineers to share case studies with verification data.

The primary requirement of the repository involved categorizing the case studies by observed site characteristics. Each site may have multiple characteristics. These categories need to be linked to each study without storing the study in multiple places. The site characteristics should be searchable in order to find studies that included one or many specific characteristics.

Ideally, each case study also includes information about the submitter. In the case of policy changes regarding the repository, these individuals could be notified. The contact information should be accessible only by those administering the repository.

Maintaining and administering the repository should require minimal effort and time. The repository should verify as much information as possible and require minimal human interaction to verify data.

The repository must be accessible to the public. Allowing public access presents a security vulnerability, thus basic measures should be taken to prevent inappropriate use of the provided tools. A simple and easy user interface should allow users/contributors to meet the previously stated requirements.

Based on the stated design requirements, a web based interface was the most appropriate form for the repository. A web interface allows global access to users in a media familiar to a wide audience.

3.3 Design Tools

The repository uses a number of existing tools to fulfill the design specifications. As stated, the repository needed a web based interface which could store and query information about case studies. The web interface needed to be generated dynamically, as the public would be contributing to the repository. The following tools and services were used to accomplish the design:

- PHP: Hypertext Preprocessor (PHP)

- MySQL

- Amazon Simple Storage Service (Amazon S3)

PHP is an open source scripting language which allows for dynamic generation of HTML code. The language itself resembles the C coding language and incorporates functionality for interacting with databases, such as MySQL. (The PHP Group 2008)

The MySQL database, also open source, was chosen to store the necessary information about the cases and users. MySQL provides a variety of database engines, each with their own set of features. The InnoDB engine was used for the repository largely for its ability to link attributes from various tables (i.e. link a user entry to a case entry). The database access is both fast and flexible, which made this an excellent choice for the database storage. (Sun Microsystems, Inc. 2008)

The Amazon S3 service provides scalable web storage. Unlike PHP and MySQL, the Amazon S3 service is not open source. The service allows for the storage and access of data. Users are charged only for the amount storage and data transfers used. The ability to distribute transfer loads, gives the S3 service an advantage over normal web hosting services. Any number of users can access and download the same data without the

bottleneck of server access. Using this service allows the repository to allow a large number of users to upload and download data simultaneously without being affected by slower transfer speeds. Further details about the Amazon S3 service will not be discussed, as it acts only as a storage center for data. (Amazon.com 2008)

These tools allow for the dynamic generation of the web pages used by the repository. As users will be contributing to the repository, the ability to generate web pages dynamically is essential. The web interface gives users the tools necessary to interact with a database in a method familiar to them. There is no need for the average user to learn how to query and add items to the database in order to contribute studies to the repository.

3.4   Underlying Database Design

The MySQL database stores all the information for the users and case studies except for the data files. The database design needed to create links between users, case studies, and site attributes. In addition to storing this information, the database design allows for the storage of various numeric engine types (i.e. riverine, coastal, watershed, etc). Creating the database tables in the right manner minimizes the data which must be stored and allows for easier queries. The various table designs will be discussed in this section. Brief descriptions of the data being stored and links to other tables will be presented. Figure 3-1 provides a graphical overview of the database design.
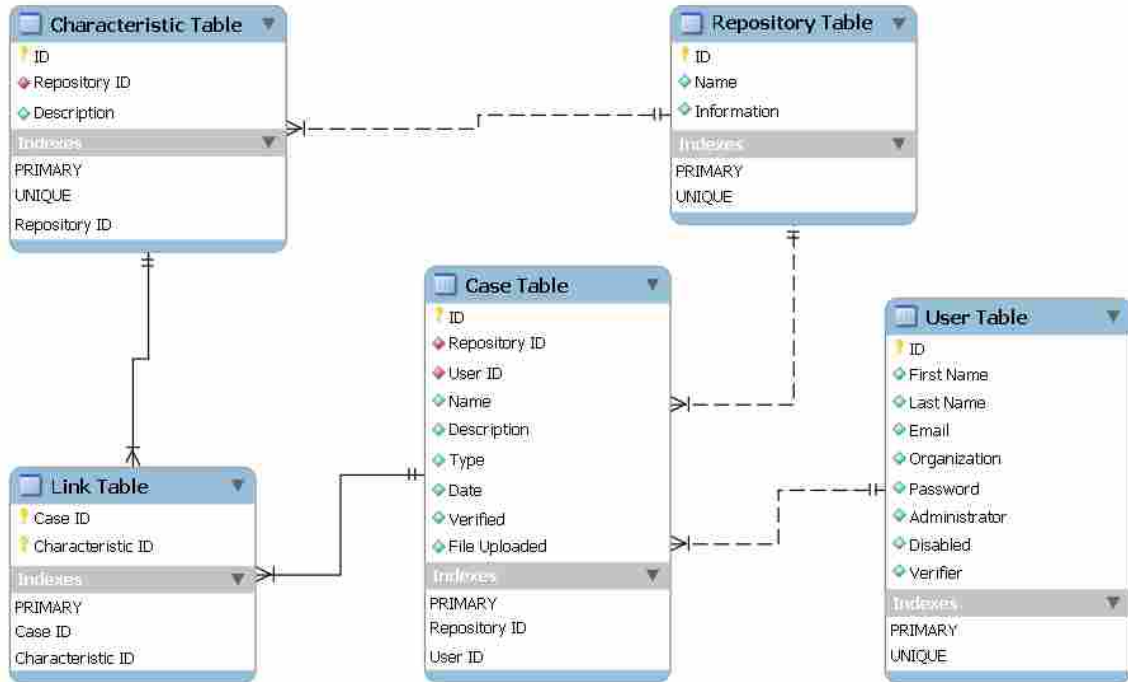
Figure 3-1: Database Tables and Relationships

3.4.1 User Table

The users table stores the information for each user registered with the repository. All registered users have rights to add cases to the repository. The values stored in the database for each user include:

- ID – a unique identifier assigned to quickly identify each user.

- First and Last Name – the user's name displayed in the web interface.

- Email – the contact information for the user, the email address is never made available to non-administrative users.

- Organization – an optional field specifying the company/organization for which the user works.

- Password – a one-way hash (the output of a function which takes an input string and computes a random-looking output string) of the password specified by the user. The password is never stored in plain text for security reasons.

- Administrator – a flag specifying whether the user has administrative rights.

- Verifier – a flag specifying whether the user can mark cases as verified.

- Disabled – a flag specifying whether the account has been disabled. This should be set only if users have used the repository inappropriately.

In addition to the users table, there are two tables which help in the activation and password reset process. Each of these tables stores the user ID and a unique string which must be provided to perform the requested action. As these tables are for convenience only, they will not be discussed.

3.4.2   Repository Table

The repositories table stores the information necessary for each individual repository. In this sense, an individual repository consists of a set of case studies grouped by numeric engine type (i.e. riverine, coastal, etc). This table provides the ability to support multiple engine types without confusion. The attributes stored in this table include:

- ID – a unique identifier assigned to quickly identify each repository.

- Name – the name of the repository. This name should describe the numeric engine type.

- Information – this text block is displayed on the information page for the repository.

Each engine type repository contains a set of case studies. The information for each individual case study contains a link to a single repository. As each case contains site characteristics specific to the type of engine, a case can only be contained in a single repository. For a case to be contained in multiple repositories, it must be added to each one individually.

3.4.3 Case Table

This table stores the information associated with each case study submitted to the repository. A single case table contains all the information for all repositories. The user provides much of the information contained in this table when submitting a study. This table works in conjunction with the characteristics and links table (discussed in Sections 3.4.4 and 3.4.5) to store, query, and display information about the cases in the repository. The values stored in the cases table include:

- ID – a unique identifier assigned to each case.

- Repository ID – references the ID of the repository to which the case belongs. This enables all cases to be stored in a single table regardless of the repository.

- User ID – references the ID of the user who submitted the case. Only one user can be linked to each case.

- Name – a descriptive name assigned to the case.

- Description – this text block contains a more detailed description of the study. This may include information about location, reason for collecting the data, etc.

- Type – specifies the type of study. Current options for this field include analytical, laboratory, or real world.

- Date – stores the date the case was submitted to the repository.

- Verified – a flag specifying whether the case has been verified by a user with verifier privileges.

- File uploaded – a flag specifying whether the files associated with the case have been uploaded. This saves accessing the Amazon S3 to determine whether the files exist.

The verification flag enables users to see whether a case has been verified for content. If the case files have not been checked for content and completeness, a message displays on the case page. This message warns that downloading the case files should be performed at the risk of the user as the content is currently unknown.

The characteristics for the cases may vary in number, which makes them difficult to store in the case database. The characteristics and links table create the necessary links to specify the site characteristics for each case.

3.4.4 Characteristic Table

This table stores the available characteristics to choose from for each numeric engine type. These characteristics can be linked to a case study which helps define the site of the study. The attributes in this table are:

- ID – a unique identifier assigned to easily indentify the characteristic.

- Repository ID – references the repository ID to which the characteristic is assigned.

- Description – a brief description of the characteristic. This should be short and general (i.e. Supercritical flow, Flow constriction, etc).

A single table stores the characteristics for the various repositories. The repository ID attribute allows association of each characteristic to a single repository. Combing this data into one table makes tables easier to manage and eliminates the need for dynamic table creation. A link between the characteristics in this table and a single case in the cases table completes the case specification.

3.4.5 Link Table

The links table provides the means to link site characteristics to specific case studies. This table allows the database to store multiple characteristics for each case. The table columns include:

- Case ID – an ID for a specific case.

- Characteristic ID – an ID for a characteristic.

By creating a table that stores links between ID in other tables, a one-to-many relationship can easily be established. This eliminates the need to store data in a bit field or list, which result in calculations or string parsing.

3.5   Web Interface

The web interface provides tools to the users to interact with the database described in the previous section. The site takes advantage of the PHP scripting language to dynamically create menus, display and search cases, and validate form input. The site also requires that all pages be loaded through a single PHP file. This ensures that all the navigation menus, forms, and page layout are uniform throughout the site. This section will discuss the navigation and main aspects of the site.

3.5.1   Navigation

Navigation menus are located on the side and top of every page. These menus provide links to jump to different sections of the web site with ease. A majority of the pages are accessible from these menus, eliminating the need to navigate pages in a specified order to locate the desired page.

The top navigation menu contains items which are somewhat unrelated to the repository data. The links found on the menu include: Contact Us, Help, and Log in/off. These items are self explanatory and, with the exception of Log in/off, are provided for the help and convenience of the users.

The main navigation menu for the site is located on the left side of every page. This menu provides links to the main page, account information, and various engine repositories. Samples of the navigation menu are displayed in Figure 3-2 and Figure 3-3. This navigation menu provides the functionality to create accounts, reset passwords, activate accounts, search repositories, and add studies to repositories.

Figure 3-2: Example Navigation Menu with User Logged Out

The account section of the menu differs depending on the logged in/out status of the user. With no user logged in the menu provides the ability to create an account, reset a password, or request activation of an account. With a user logged in, the menu contains a link to display user account information. From the user information page, a user can change their password. Below this link a section containing whether files are missing from submitted cases is found. This appears only if the user submitted a case and failed to upload the necessary files corresponding to the case. The number next to the repository name gives the number of cases with missing files. Clicking on this link will perform a search which returns the case(s) in question.

As more engine type repositories are added to the database, the navigation menu will grow to include these repositories.
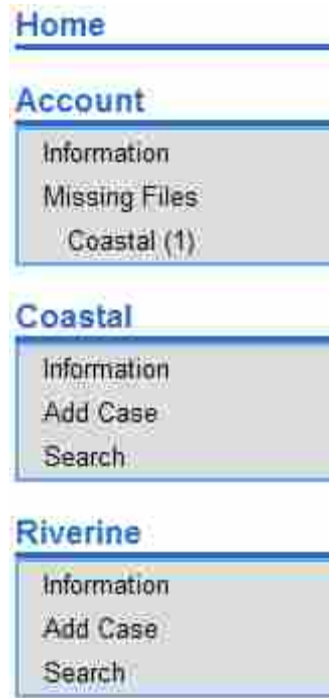
Figure 3-3: Example Navigation Menu with User Logged In

3.5.2   Web Forms

Web forms provide a means for users to submit information to a web server. The forms on the repository web site have been designed to provide useful help and information to the users. A sample form is shown in Figure 3-4. This form prompts the user for the necessary information to create an account. An asterisk and bold font mark the fields which must be provided by the user before submitting the form. Additionally, notes providing guidance to the users are provided in caption boxes on the right side of sections.

After a form has been submitted, the web server validates the input. If incorrect values have been entered into the fields, the form is displayed again with error messages to help the user correct the problems. Figure 3-5 shows sample errors messages which

may be encountered due to incorrect input values. Once all errors have been resolved, the server processes the data and directs the user to the desired page.

Figure 3-4: Create Account Form

3.5.3   Compatibility

Many web sites contain informational messages alerting the user that some aspects of the site may not function correctly due to the web browser. Care was taken to use only standard HTML code and styles. JavaScript, which allows for client side scripting, can allow web developers to validate forms and perform other operations. As some users disable JavaScript for security reason, it was avoided at all costs. Following these rules ensures the web pages on this site display properly for all users and browsers.

Figure 3-5: Create Account Form with Error Messages

3.6 Security

A large concern with web design is security. A large number of bots exist which create accounts at various sites and maliciously add content to the sites. There are many methods implemented to help prevent the use of bots and reduce malicious use. Some of the common methods include captchas and confirmation emails. This site uses confirmation emails to ensure the email addresses provided are valid. In order to activate an account the user must visit a link provided to them in an email.

3.7   Web Site Layout

The web site design requires access to all web pages through the index page. PHP scripting allows the index to add similar portions of the page and then add customized content according to the requested page. Designing a site in this manner allows some validation and error handling to take place in one location. As many site elements use the same PHP scripts, the site maintains a uniform look and feel throughout.

This site takes advantage of the HTTP GET method, which allows parameters to be passed though the web address. Figure 3-6 illustrates how the GET method is applied to load web pages. The address bar displays the web address as "http://verification.aquaveo.com/?do=create_account". The GET parameters can be seen after the "?" in the address, "do=create_account". PHP parses these parameters and loads the body for the "create_account" page into the main page. Before the main page loads the "create_account" body, it verifies that the page is legal to load. In this case, the page requires a user be logged into the site. If a user manages to reach a page that cannot legally be loaded, an error message displays explaining the reason.

Using the HTTP GET method to load web pages from a single access point (i.e. main web page) allows the checking of legality to be centralized. For this site, certain pages can only be loaded if a user is logged in or an engine type repository has been selected. These checks take place in once location and an error displays if the user reaches an invalid page. This method helps prevent curious browser from visiting pages which they do not have rights to view.
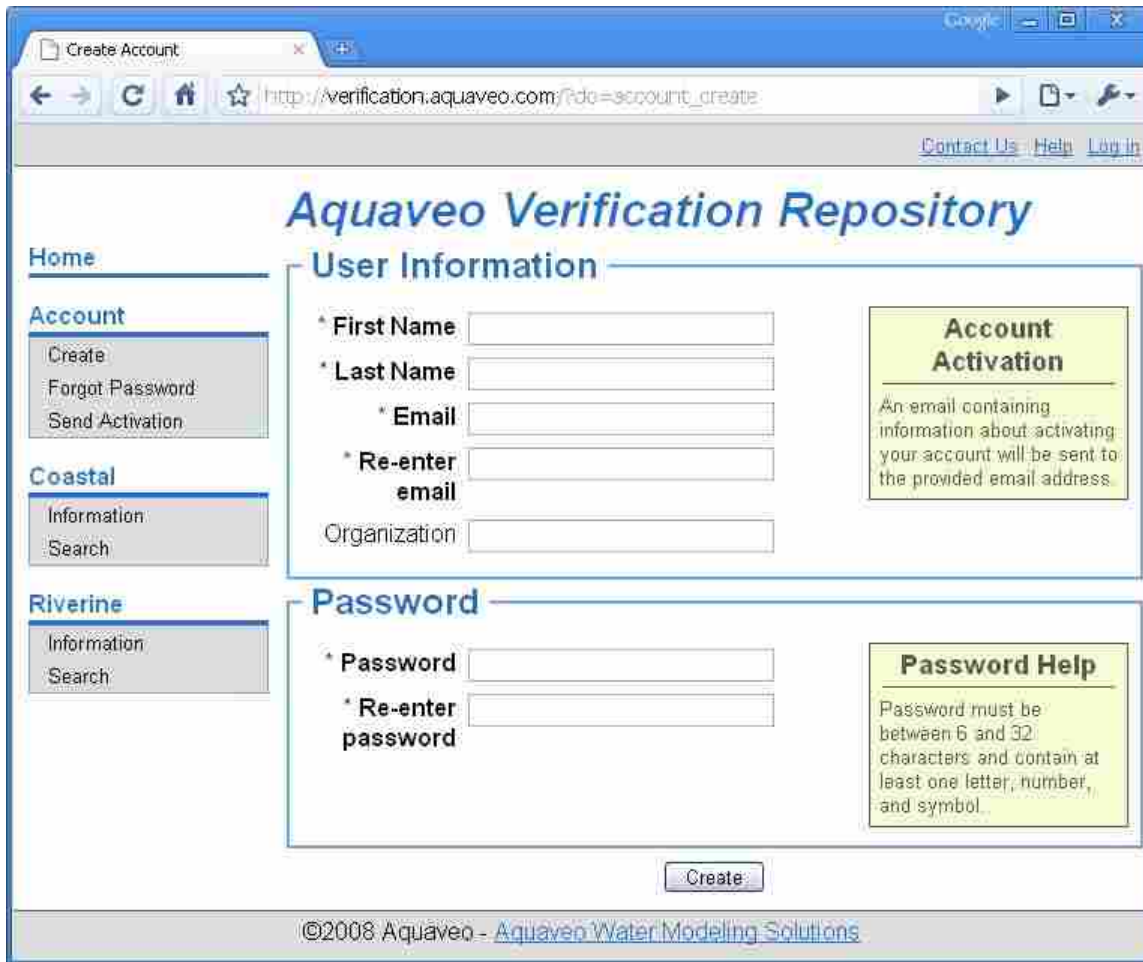
Figure 3-6: Sample Web Page

3.8   Summary

This web site utilizes a number of tools to create an interface for the repository. A MySQL database stores the information relating to user accounts and submitted case studies. PHP queries the database and generates the HTML code necessary to display the web pages. Web forms allow users to input information, after validation by PHP scripts, into the database. Chapter 4 discusses a case study and further use of the web site to submit the data to the repository.

4　Whitewater Creek Case Study

In the 1970's the U.S. Geological Survey, in cooperation with the Federal Highway Administration Department of Transportation, the Alabama State Highway Department, the Louisiana Department of Transportation and Development, and the Mississippi State Highway Department, collected data at bridge crossings of wide, vegetated flood plains. Data were collected for 35 different floods at 22 sites throughout the states of Alabama, Mississippi, and Louisiana. The purpose of creating this collection was to aid in the development of improved numeric engines. The emphases of the data were primarily bridge crossings, but the studies can be applied to riverine engines in general. One of the sites included in this collection is Whitewater Creek. This chapter discusses the process of digitizing, preparing, and submitting the data for Whitewater Creek to the verification repository. (USGS n.d.)

4.1　Background

Whitewater Creek lies near the city of Tarentum, Alabama. On March 2, 1972 a flood event occurred. The peak discharge for the event was 158 cubic meters per seconds, which corresponded to a 6 year recurrence interval. Flood plain and channel cross sections were collected after the event, along with bridge geometry. During the event, water surface elevations and discharge measurements were collected. The discharge

measurements were taken near the bridge crossings at three different times during the flood. (USGS n.d.)

4.2   Available Data

The data, cross section, discharge, and water surface elevation, are currently available as a series of reports in map form. The USGS Mississippi Water Science Center scanned the maps and posted them on their website as compressed MrSID image files. The Whitewater Creek study includes three maps, shown in Figure 4-1 through Figure 4-3. Figure 4-1 illustrates the data layer for the atlas. This page contains background, cross section, and discharge measurement data. Figure 4-2 contains topographic data with the cross section locations shown. Figure 4-3 contains a series of observation points covering the area of study. These three maps provide the necessary data to construct a numeric model simulation.

4.3   Digitizing the Data

Although all the data to build a model simulation exists, the data in image format requires conversion before being used. In order to construct a two-dimensional model, each cross section must be made into a series of three dimensional points. Water level and flow rates must be assigned at appropriate locations. The process of turning data from an image into usable modeling data is referred to as digitizing. The process is both tedious and time consuming. The Surface Water Modeling System (SMS) and Microsoft Excel provided tools to aid in the digitizing process.

Figure 1—Index map of study sites in the bridge backwater investigation project, Alabama, Louisiana, and Mississippi.

Figure 2—Index map showing study reach, Whitewater Creek near Tarentum, Ala.

TABLE 1—VALLEY CROSS SECTION DATA FOR WHITEWATER CREEK NEAR TARENTUM, ALABAMA. ZERO STATION IS AT THE LEFT EDGE OF THE VALLEY (FACING DOWNSTREAM).

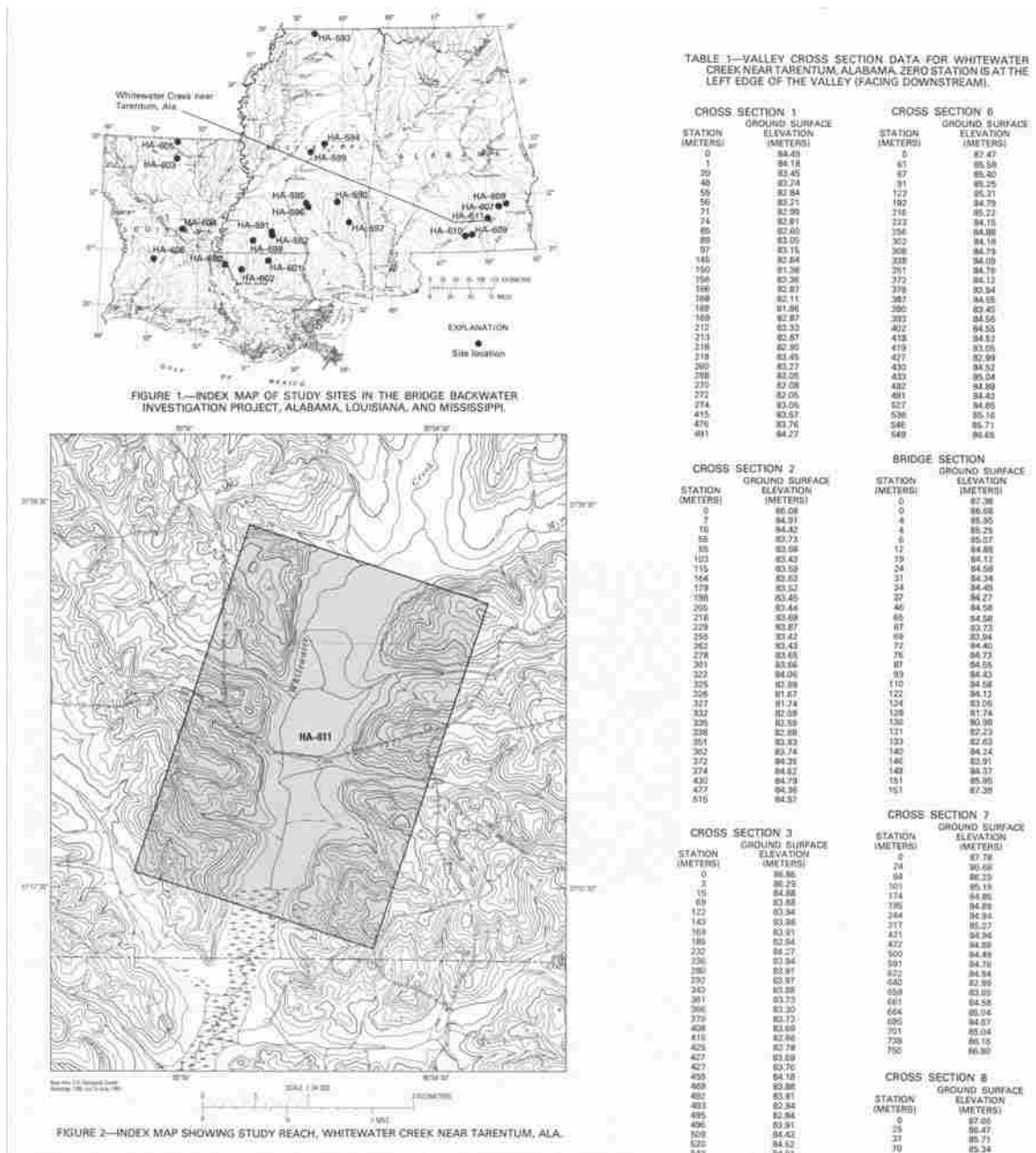**Figure 4-1: Sample Whitewater Creek Background/Discharge Data (USGS n.d.)**

The first step in digitizing the Whitewater Creek case study involved georeferencing the provided image files. Georeferencing an image involves associating particular points on the picture with world locations to position the data in a specific coordinate projection or distances to create appropriate length scales in a local system.

39

For this case, the georeferencing used only distances. The scale bar at the bottom of the image provided the means necessary to associate pixel sizes with real distances. The end result of this process was a map that could be used to determine relative positions between various features on the image. Georeferencing the image using only distances meant that it was not possible to import other datasets and line up the data. However, since the images contain all the necessary information, this was not a requirement.

With the image georeferenced, the next step in the digitizing process was creating the cross sections. Upon comparing the locations provided in Figure 4-2 with the cross section data in Figure 4-1, it was discovered that the point set provided was incomplete. Microsoft Excel was utilized in order to translate the cross section data into world locations. The cross section data were entered manually into the spreadsheet using the values found in Figure 4-1. Table 4-1 shows a sample of a single cross section input. Visual Basic (VB) functions, found in Appendix A, were written to parse the spreadsheet in order to convert the cross sections. The "BEGIN" card marked the beginning of a new cross section. The four values sharing the row with the "BEGIN" card were the x, y world location of the first cross section point and any other x, y world location on the cross section. These two points were obtained within SMS using the georeferenced cross section image, Figure 4-2. This process involved creating a scatter set by clicking the point locations on the map. The points were clicked out by selecting the location corresponding to the first point in the cross section, then clicking another point along the cross section. This was repeated for each cross section. A tabular data file containing the x, y locations was exported from SMS and then imported into the spreadsheet using VB functions. The first point specified the translation, while the second point provided the

information necessary to rotate the cross section. The "END" card specified the end of the cross section. The rows in between the "BEGIN" and "END" cards contained the cross section data found in Figure 4-1. The Visual Basic functions output an ASCII file containing the x, y locations with a corresponding elevation.

The cross section data points, which were exported from Excel, were imported into SMS and overlaid on the image. Figure 4-4 shows the resulting georeferenced cross section data in red. The data may seem sparse for a two-dimensional model, but this is not the case. It is a common misconception that two-dimensional models require more data points. The data collected must simply represent the topography of the area of interest. A two-dimensional model requires more information regarding spatial relationships between cross sections, but it does not require more cross section data.

The calibration data for a two-dimensional model should be collected at various locations along each cross section. A one-dimensional engine computes only one value along each cross section, so only a single data point is required for calibration. As this case study may be used to validate both one- and two-dimensional engines, a few observation points at each cross section were digitized from Figure 4-3. Unlike the cross section data, the observation points appear in random clusters. Only SMS was utilized in this digitizing process. In the general location of each cross section, a few points were chosen (one near each bank and one in the middle of the channel). The digitized observation points are shown in red in Figure 4-5.

Table 4-1: Example Cross Section Input For Converting Spreadsheet

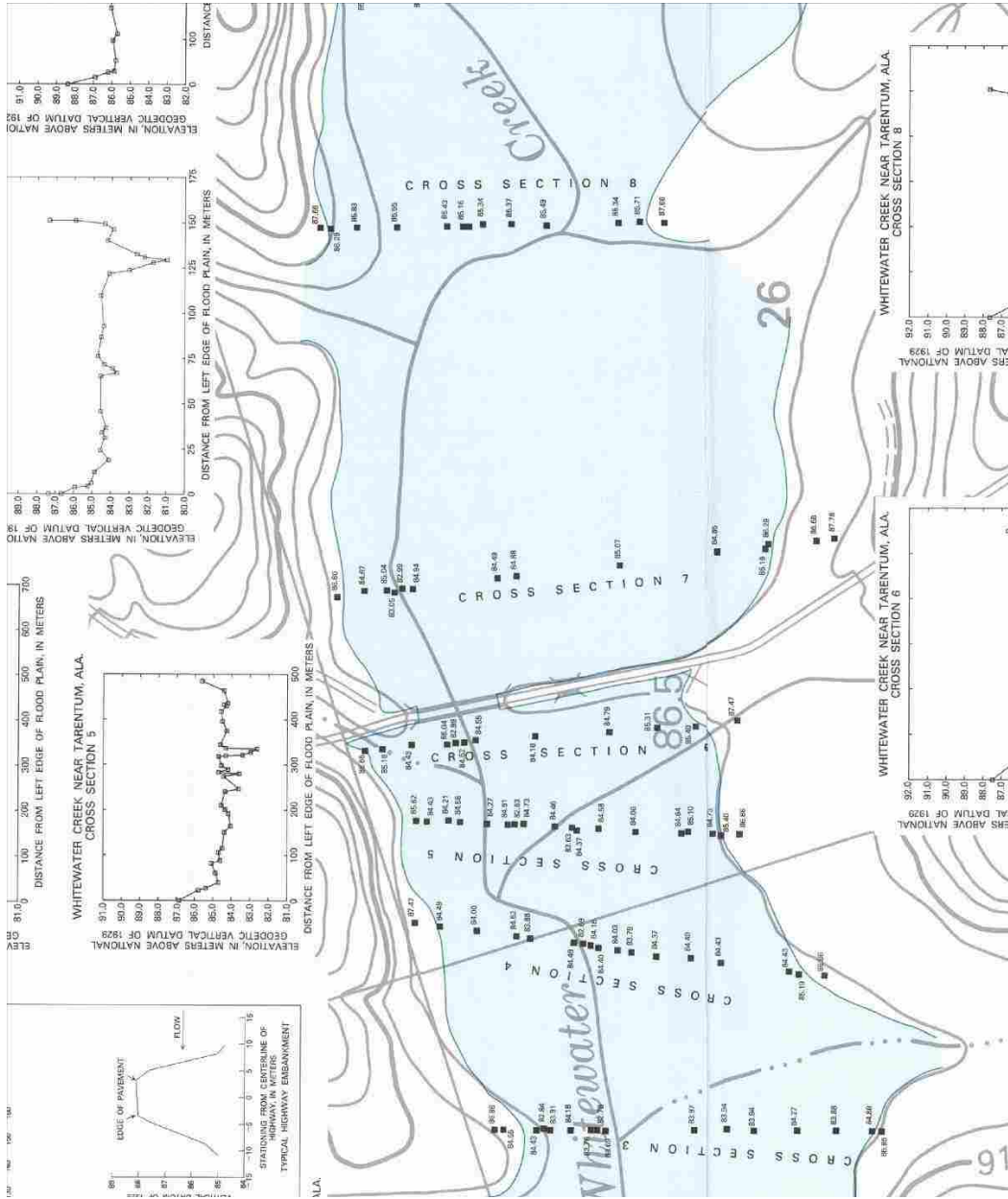| BEGIN	-913.7	944.4	-974	1442.9
0	84.49			
1	84.18			
20	83.45			
48	83.24			
55	82.84			
56	83.21			
71	82.99			
74	82.81			
85	82.6			
89	83.05			
97	83.15			
145	82.84			
150	81.38			
156	83.36			
166	82.87			
168	82.11			
169	82.87			
212	83.33			
213	82.87			
218	82.9			
219	83.45			
260	83.27			
268	83.05			
270	82.08			
272	82.05			
274	83.05			
415	83.57			
476	83.76			
491	84.27			
END				

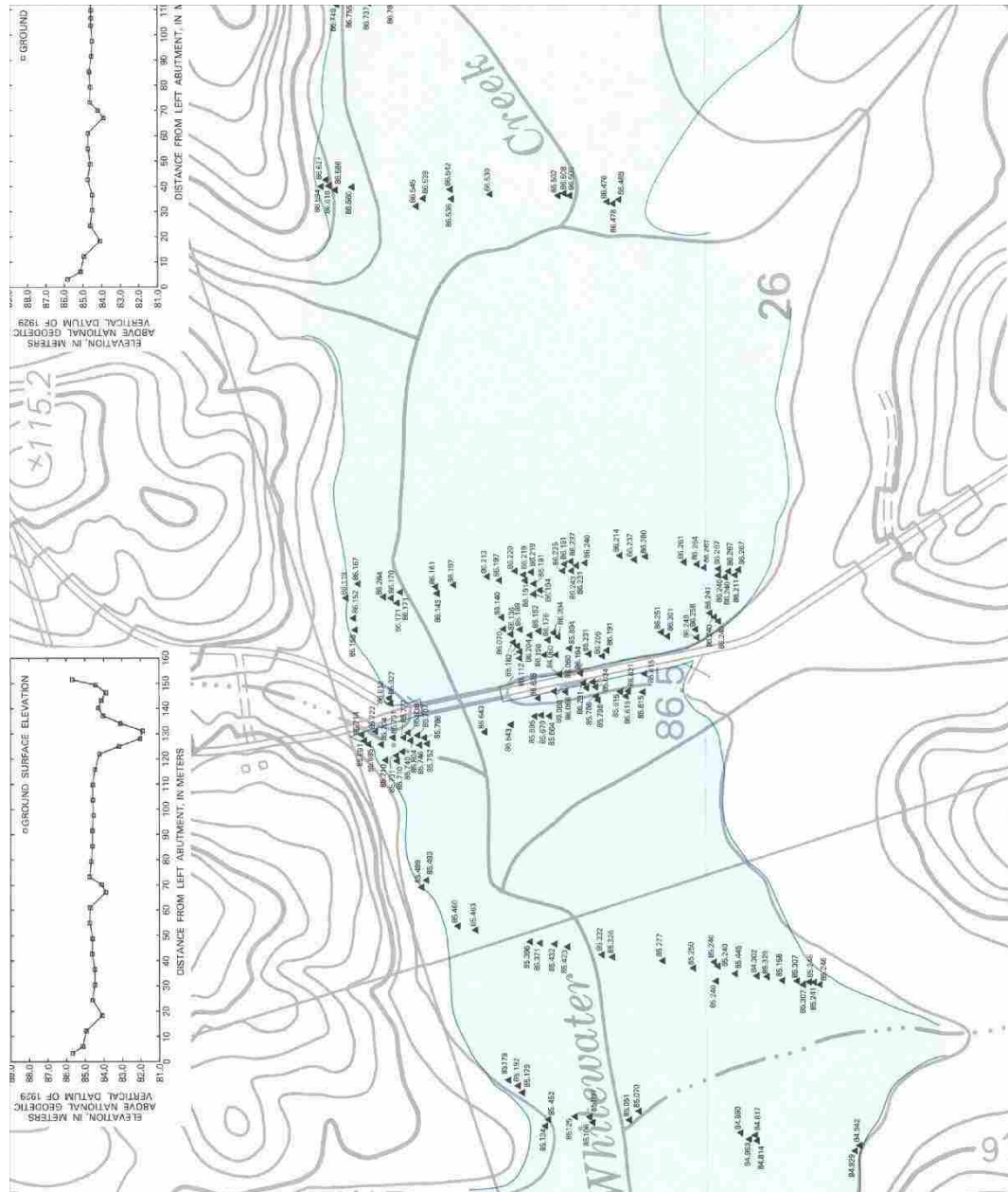**Figure 4-2: Sample Whitewater Creek Topographic Data (USGS n.d.)**

43

Figure 4-3: Sample Whitewater Creek Observation Data (USGS n.d.)

44

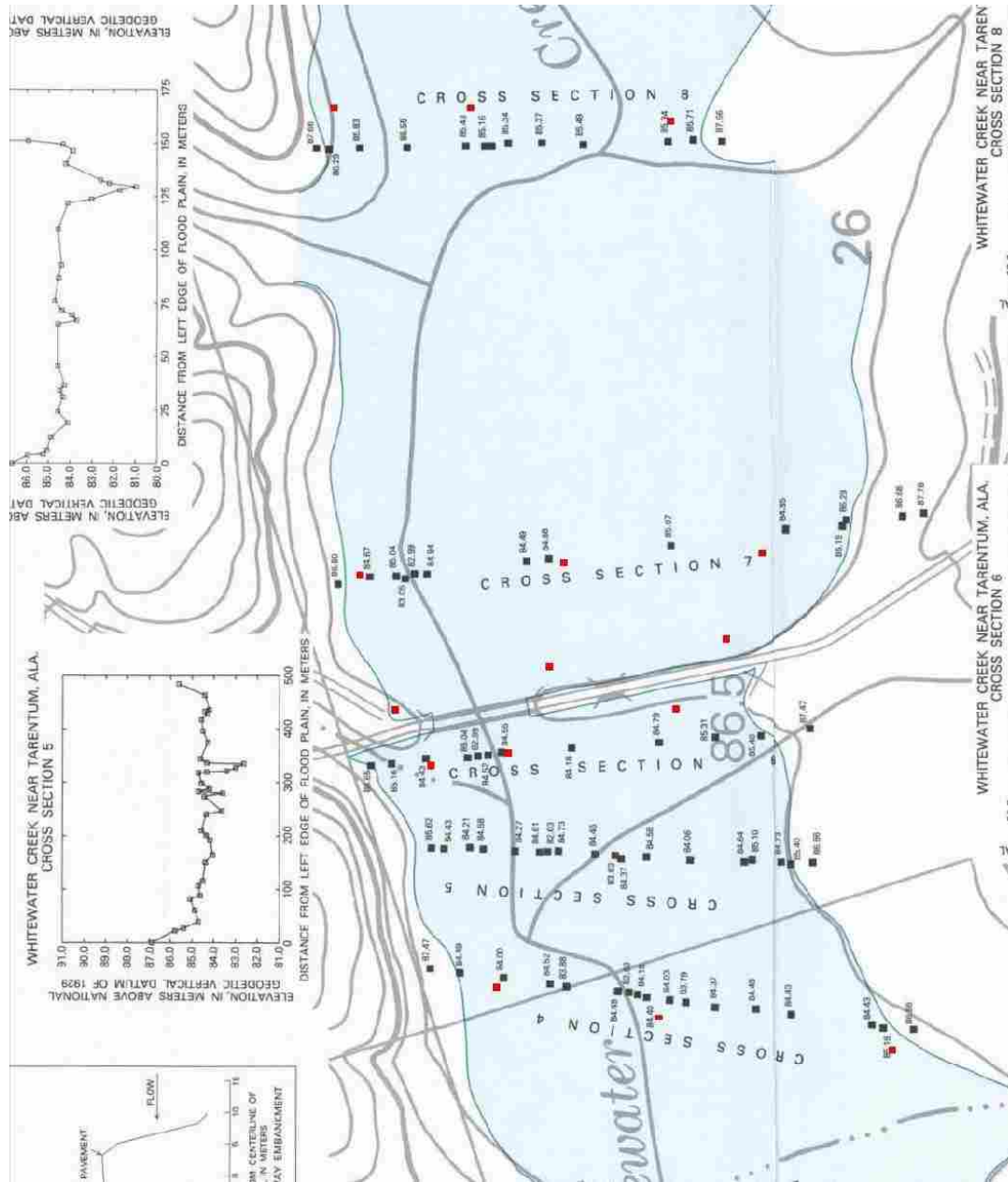**Figure 4-4: Whitewater Creek Topographic Data with Cross Section Points**

45

Figure 4-5: Whitewater Creek Topographic Data with Observation Points

4.4 Boundary Conditions

Boundary conditions specify the conditions which hold at particular locations of the model domain. The Whitewater Creek case uses flow and water surface elevation conditions on the boundaries. At three different times during the flood, discharge measurements were taken. These measurements were made near the bridge (USGS n.d.). In order to specify the boundary conditions at the edges of the model, it must be assumed that the flow at the bridge matches the flow at the upstream boundary. The observation points in Figure 4-3 correspond to the second discharge reading of 121 cubic meters per second. The downstream boundary condition, a water surface elevation, at this flow rate is 84.37 meters. This value comes from the observation points near the downstream boundary. In the images, the upstream boundary is on the east, or right, side.

The time delay between when flow was measured and when the peaks were recorded was unspecified. The verification, in this case, would involve simply involve observing the peak values reported by the numeric engine and comparing those to the peaks measured.

4.5 Submitting the Case Study

The verification repository has a specific set of file requirements when submitting a case study. These requirements outline the files which are necessary to successfully verify a numeric engine. The necessary files include:

- Bathymetry/topography – This file contains the information necessary to recreate the geometric aspects of the site.

- Boundary conditions – This file specifies the input conditions at the boundary of the model. These values should be measured in the field.

- Calibration/verification data – These data specify target values which were observed in the field.

- Description document – This document contains a description of the files included in the case study package. This provides other users with information about the contained data. This is sometimes referred to as metadata.

Depending on the type of model, other files may be necessary, but this set defines the bare minimum needed to reconstruct any model type. The Whitewater Creek files were prepared according to the specifications. The file names listed below are not a standard convention. The names simply provide an idea of what the file contains. The description file, named description.txt, contains a summary of file contents. The files included in the package were:

- bc.txt – A brief explanation of the boundary conditions to be applied at the upstream and downstream data.

- cross_sections.xyz – The digitized x, y, z cross section data in a space separated file.

- description.txt – A description of the other files contained in the package.

- observation.xyz – The digitized x, y, z observation data in a space separated file.

- observe.jpg/jpw – The observation report page with a corresponding world file. The world file provides the information necessary to georeference the image.

- topo.jpg/jpw – The report page containing the cross section data with it's world file.

- Sheet_1.sid/Sheet_2.sid/Sheet_3.sid – The MrSID image files provided by USGS. These contain all the information found in the other files, but the format is not useable for constructing a model.

With the data digitized and gathered, the last step was to add the information and files for the case to the repository. The case was added to the "Riverine" repository using the following steps:

1. The following parameters and a brief description of the case were entered into the form, as shown in Figure 4-6.

   - Case name – Whitewater Creek

   - Type – Real World

   - Characteristics – Bridge Crossing, Flood Conditions, Flow Constriction, and Vegetated Floodplain

2. Upload the files, mentioned above, as a zip archive. After filling out the form shown in Figure 4-6, another form prompts for the files. This form consists of one field, so it is not shown.

Once the upload completed the case was marked as verified and the files are now available for download and use. Figure 4-7 shows the final result available on the web site.

The cases on the repository may be queried based on case attributes. Figure 4-8 shows the search page for the "Riverine" repository. A user may select the model type and/or site characteristics. The site characteristics portion includes all the options used by any case in the repository. Leaving all criteria blank will return all cases in the selected repository. Selecting multiple criteria will return all cases containing any combination of the selected attributes. Checking the "Has all selected" toggle under characteristics will return only the cases containing all of the selected characteristics.

Figure 4-6: Filling Out the Case Information
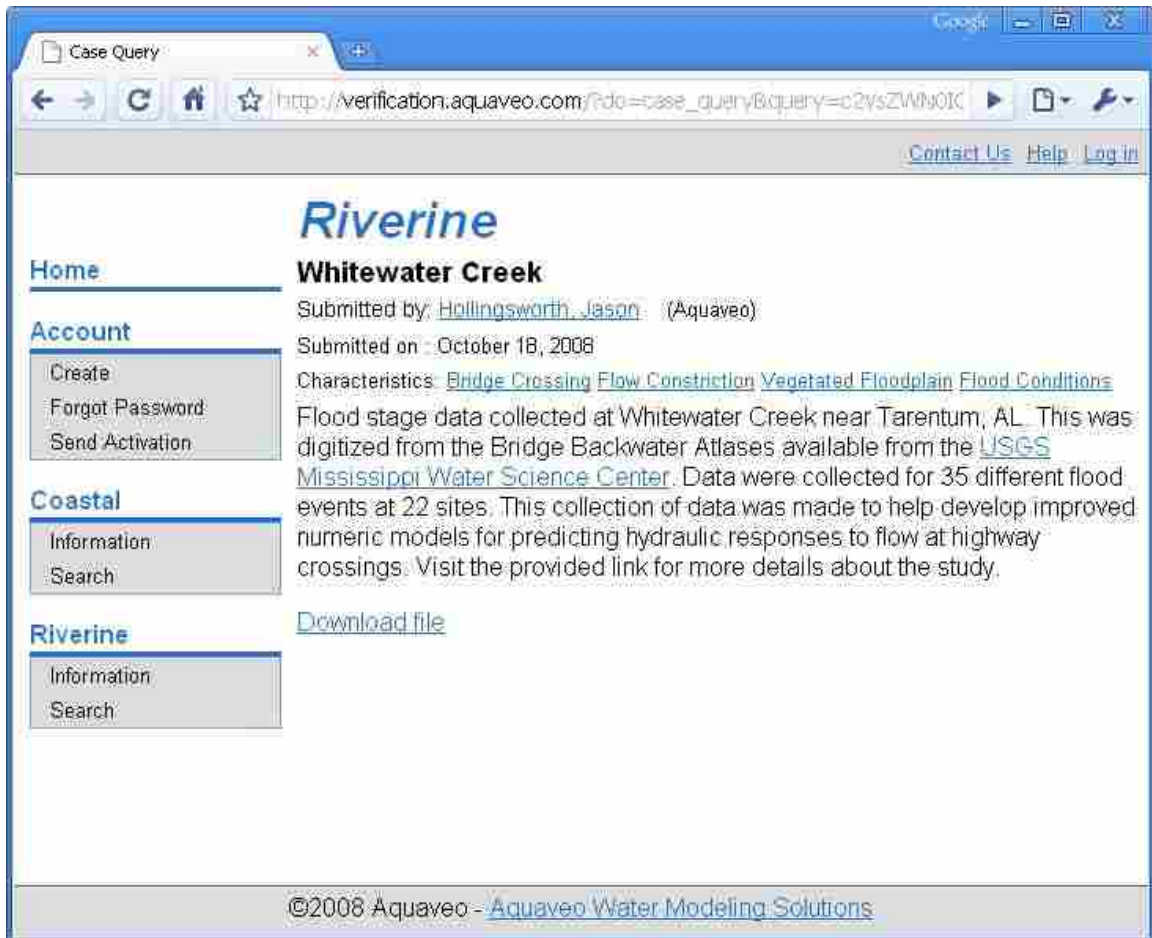
Figure 4-7: Whitewater Creek Case Page on Verification Repository
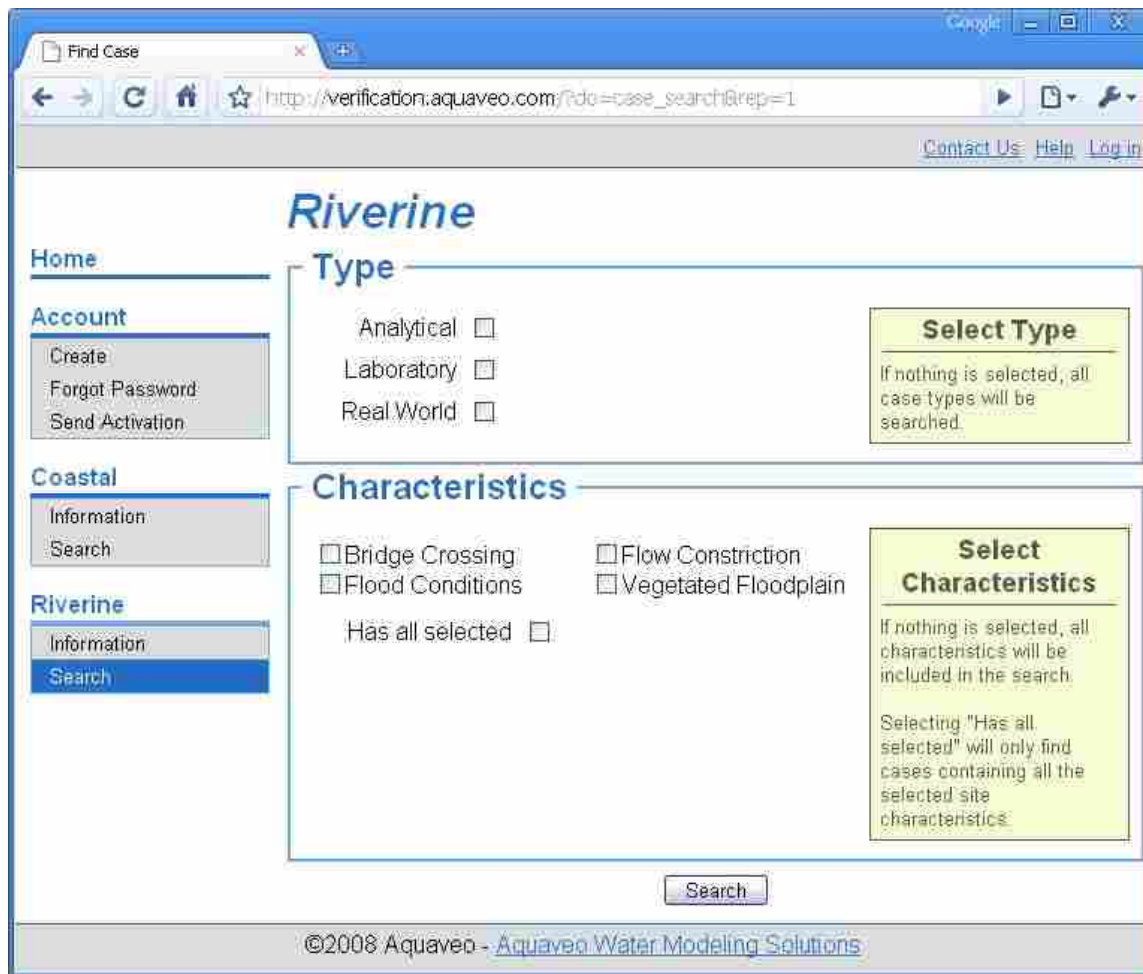
Figure 4-8: Search Page for the Verification Repository

4.6   Overview

The USGS Mississippi Water Science Center provides a series of Bridge Backwater atlases which contain data collected in the 1970's at various bridge crossing sites throughout the states of Mississippi, Alabama, and Louisiana (USGS n.d.). The information currently available to the public consists of series of compressed images scanned from reports. While these images contain the necessary data to build a numeric model, the format needs to be changed in order to be useable by the general public. The steps to converting one of these atlases are as follows:

1. Download the maps sheets from the USGS Mississippi Water Science Center.

2. Convert the MrSID images to jpg images. The current version of SMS does not support rotating MrSID images, which is necessary for georeferencing.

3. Using the scale bar provided on the images, georeference the image.

4. Ensure the images overlay properly. It may be necessary to set benchmarks on one image and georeference the other image using the benchmarks.

5. Enter the cross section data into the Microsoft Excel spreadsheet.

6. Digitize anchor points and another cross section point from the image and input them into the spreadsheet.

7. Convert cross section data into world locations using the spreadsheet.

8. Import the cross section data to ensure it overlays properly.

9. Create observation points within SMS using the observation data provided on the images.

10. Prepare data, following the guidelines on the verification repository web site.

In the case that a model for the Whitewater Creek case study was built, those files could be included in the zip archive and replace the current file set. The operation of replacing existing files is limited to users with verifier rights. This helps ensure the packages available for download contain only data which has been reviewed for content and security.

The Whitewater Creek case study has been prepared as an example of the necessary components for a verification study. The data were collected for the specific purpose of creating improved numeric engines and thus makes it an ideal candidate for an

example. When preparing a case study for the repository, this case should be used as a

guideline of required files and format.

5 Conclusion

Engineers wishing to construct numeric models to aid in analysis/design face many choices when selecting an engine. With the number of engines constantly rising, providing a tool to assist in the selection process becomes increasingly important. An effective selection tool should take into account the strengths and weakness of the engines. Evaluating strengths and weaknesses involves comparing results from various engines to real world values. This report has discussed a means of cataloging and storing case studies which can help in the engine evaluation process.

5.1 Improved Selection Tool

Tools and guidelines exist to aid engineers in the process of selecting an engine for a study. These tools, however, do not provide the necessary information to ensure the selection of an appropriate engine. An effective selection tool should:

- Store information regarding engine performance for various site characteristics
- Not require input from engineers about engine performance during selection process
- Return a list of appropriate engines based on site characteristics
- Show engine performance with site characteristics

- Have a specific set of test suites used to determine engine performance

- Provide useful feedback about engine strengths and weaknesses

The first step in creating this new selection tool requires case studies to be gathered, categorized, and stored in a central location. These studies provide a pool of tests to select from when building test suites for the numeric engines. Building this repository for storing case studies has been the focus of my research. Without these real world studies, the selection tool would not contain enough information to intelligently suggest appropriate numeric engines for a study. Only by verifying engine results against real observations can performance be quantified. When the repository has been populated with enough case studies, work on the next step of the selection tool can commence.

5.2   Future Work

With the repository in place the next step involves outlining a series of test suites for numeric engines. The exact time frame for starting the next phase of the tool is indeterminate, as it cannot begin until a sufficient number of case studies have been populated. Populating the repository requires contributions from the engineering community. The case studies in the repository will consist of data collected for various studies and projects, unrelated to numeric engine verification. Although data exists which has been collected for the express purpose of engine testing, this will most likely not be the main source of studies. Time and cost constraints make it impractical for many organizations to collect data simply for testing and verification of numeric engines.

The work of constructing test suites is not a trivial task. The suites must test a wide range of engine capabilities and also be designed in such a way as to isolate

56

weaknesses and strengths. The engine results must show the performance levels when faced with specific characteristics. The suite must also contain sufficient documentation so future engines can be verified accurately. If strict guidelines and steps are not followed during the testing process, engine performance results may be inaccurate.

The results from the test suite runs for each engine must be analyzed and stored. A tool must be designed to perform this task and also query the analyzed results. This process will play a major role in the effectiveness of the selection tool. The ability to suggest appropriate engines based on real results, balances on the capability to process and query result data. This part of the selection tool will be the interface that most engineers will use to select a numeric engine for a case study. If the final part of the selection tool does not provide an interface which is easy to understand and use, the prior processes will have been in vain.

5.3 Challenges

Collecting case studies for the repository proved to be a more difficult task than originally anticipated. Many of the individuals contacted did not respond, did not have sufficient data for verification, or were unable to release the data for legal reasons. Those individuals/organizations with the type of data needed for the repository have gathered the data with other purposes in mind. These reasons range from consulting jobs to lawsuits. Thus, the data is sensitive and cannot be released until the proceedings in questions are complete. Contacting the individuals at a future date to follow-up on the proceedings and availability of the data could prove beneficial.

5.4 Final Thoughts

Numeric engines save engineers time and money during the design and analysis stages of a project. Currently, however, the numeric engine selection consists of selecting from a list of engines that claim to perform the desired computations. No guidelines exist to aid the engineer in selecting the engine which would be most appropriate for the study. The engine suggestions must be based on results from real world tests. The first step in developing a tool to perform this task is collecting case studies to use for the testing process. This report has presented a tool designed specifically to categorize and store such case studies. Once a sufficient number of quality case studies have been compiled, the work on this new selection tool can continue.

References

"Amazon Simple Storage Service (Amazon S3)." Amazon Web Services LLC. 2008. Amazon.com. 28 October 2008. <http://aws.amazon.com/s3>

"Backwater at Bridges and Densely Wooded Flood Plains at Selected Sites in Alabama, Louisiana, and Mississippi." U.S. Geological Survey – Mississippi Water Science Center. USGS. 20 September 2008. <http://ms.water.usgs.gov/publications/backwater_HA/index.h>

Buckley, Michael "Memorandum: Policy for Accepting Numerical Models for Use in the NFIP." FEMA. 13 July 1999. FEMA Library. 20 September 2008. <http://www.fema.gov/library/viewRecord.do?id=3226>

"Criteria for Selecting Numeric Hydraulic Modeling Software." Transportation Research Board of the National Academies. 2008. National Academy of Sciences. 18 September 2008. <http://www.trb.org/trbnet/projectdisplay.asp?projectid=723>

Gosselin, M., Sheppard, D.M., and McLemore, S. "Criteria for Selecting Hydraulic Models." Transportation Research Board of the National Academies. 26 September 2007. National Academy of Sciences. 9 September 2008. <http://www.trb.org/news/blurb_detail.asp?id=8151>

"MySQL :: The world's most popular open source database." MySQL AB. 2008. Sun Microsystems, Inc. 28 October 2008. <http://www.mysql.com>

"PHP: Hypertext Preprocessor." The PHP Group. 28 October 2008. The PHP Group. 28 October 2008. <http://www.php.net>

"Title 44: Emergency Management and Assistance." Code of Federal Regulations. 2008. U.S. Government Printing Office. 20 September 2008. <http://ecfr.gpoaccess.gov/cgi/t/text/text-idx?c=ecfr&sid=00c15c701e4f7da8145a415d4dc3df24&rgn=div8&view=text&node=44:1.0.1.2.32.0.25.6&idno=44>

"WMS:WMS." Aquaveo, LLC. 10 November 2008. Aquaveo, LLC. 10 November 2008. <http://www.xmswiki.com/wiki/index.php5?title=WMS:WMS>

Appendix A.   Data Transformation Functions (Visual Basic)

```vb
Sub ExportData()
    Dim x0 As Double
    Dim y0 As Double
    Dim dx As Double
    Dim dy As Double
    Dim length As Double
    Dim cosA As Double
    Dim sinA As Double
    Dim dist As Double
    Dim elev As Double
    Dim fh As Integer
    Dim row As Integer

        ' ask for output file
    fh = OpenWriteFile

        ' make sure we have a file
    If (fh = False) Then
        Exit Sub
    End If

    On Error GoTo ErrorMsg

        ' initialize the values
    x0 = NULL_VAL
    y0 = NULL_VAL
    cosA = NULL_VAL
    sinA = NULL_VAL

    row = 1
    Do While (Cells(row, 1).Value <> "")
        If (Cells(row, 1).Value = Cells(row + 1, 1).Value) Then
                        ' check for same distance in cross section data
                MsgBox "Adjacent points have same distance. Please check data."
                GoTo ErrorMsg
        End If

        row = row + 1
    Loop

        ' loop through the rows and output the data
    row = 1
    Do While (Cells(row, 1).Value <> "")
```

```vba
        If (Cells(row, 1).Value = "BEGIN") Then
                        ' read in a new anchor and compute new angle
            x0 = Cells(row, 2).Value
            y0 = Cells(row, 3).Value
                    ' compute angle cos and sin
            dx = Cells(row, 4).Value - x0
            dy = Cells(row, 5).Value - y0
                    ' get overall length
            length = ((dx * dx) + (dy * dy)) ^ (0.5)
            cosA = (dx / length)
            sinA = (dy / length)
        ElseIf (Cells(row, 1).Value = "END") Then
                        ' set the anchor and angle info back to null
            x0 = NULL_VAL
            y0 = NULL_VAL
            cosA = NULL_VAL
            sinA = NULL_VAL
        Else
            If ((x0 = NULL_VAL) And (y0 = NULL_VAL) And (cosA = NULL_VAL) And (sinA
        = NULL_VAL)) Then
                                ' data was not filled in
                    GoTo ErrorMsg
            End If
                    ' convert and save the data
            dist = Cells(row, 1).Value
            elev = Cells(row, 2).Value
                    ' for now always use a space
            Call WriteData(fh, " ", dist, elev, x0, y0, cosA, sinA)
        End If

        row = row + 1
    Loop

    Close fh

    Exit Sub

ErrorMsg:
    Close fh

    MsgBox "Unable to complete conversion."
    Err.Clear
End Sub

Sub ImportAnchors()
    Dim x0 As Double
    Dim y0 As Double
    Dim x1 As Double
    Dim y1 As Double
    Dim fh As Integer
    Dim row As Integer

        ' ask for output file
    fh = OpenReadFile

        ' make sure we have a file
```

```vba
    If (fh = False) Then
        Exit Sub
    End If

    On Error GoTo ErrorMsg

        ' initialize the values
    x0 = NULL_VAL
    y0 = NULL_VAL
    x1 = NULL_VAL
    y1 = NULL_VAL

        ' loop through the rows and fill in the anchor data
    row = 1
    Do While (Cells(row, 1).Value <> "")
        If (Cells(row, 1).Value = "BEGIN") Then
                            ' read in the anchor and second point data
                Input #fh, x0, y0
                Input #fh, x1, y1
                        ' fill in data
                Cells(row, 2).Value = x0
                Cells(row, 3).Value = y0
                Cells(row, 4).Value = x1
                Cells(row, 5).Value = y1
        End If

        row = row + 1
    Loop

    Close fh

    Exit Sub

ErrorMsg:
    Close fh

    MsgBox "Unable to complete import. Check file format."
    Err.Clear
End Sub

Function OpenReadFile()
    Dim fh As Integer
    Dim fname As String

        ' set this to false
    OpenReadFile = False

        ' prompt for name
    fname = Application.GetOpenFilename("", Title:="Specify Input File")

        ' make sure they specified a file (and open it)
    If (fname <> "False") Then
        OpenReadFile = FreeFile()
        Open fname For Input As OpenReadFile
    End If
End Function
```

```vb
Function OpenWriteFile()
    Dim fh As Integer
    Dim fname As String

        ' set this to false
    OpenWriteFile = False

        ' prompt for name
    fname = Application.GetSaveAsFilename("", Title:="Specify Output File")

        ' make sure they specified a file (and open it)
    If (fname <> "False") Then
        OpenWriteFile = FreeFile()
        Open fname For Output Access Write As OpenWriteFile
    End If
End Function

Sub WriteData(fh As Integer, sep As String, distance As Double, elev As Double, x0 As _
        Double, y0 As Double, cosA As Double, sinA As Double)
    Dim x As Double
    Dim y As Double
    Dim line As String

    x = (cosA * distance) + x0
    y = (sinA * distance) + y0

    line = x & sep & y & sep & elev

    Print #fh, line
End Sub
```