Statistical analysis and refinement of earthen embankment erosion models

by

Anthony Atkinson

B.S., Kansas State University, 2018

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2024

Approved by:

Major Professor
Mitchell Neilsen

Copyright

Abstract

This study evaluates the current state of internal erosion modeling for earthwork embankments and proposes methods of expanding upon those models using modern advancements. It is part of an international initiative on the evaluation of dam safety simulation software for internal erosion performance and best practices. The primary experiments involve simulating uncertainty in the failure events of two dams with two different models, DLBreach and WinDAM C. DLBreach is a physically-based Dam/Levee Breach model developed by Wu[1;2]. WinDAM C is also a physically-based dam breach model capable of analyzing both dam overtopping and internal erosion[3]. The dams selected for the analysis include a 1.3-meter-high dam tested in the lab and a larger 15.56 meters high dam which suffered a failure in the field. The findings from these experiments are extended with further analysis on variance, sensitivity, and optimization. Finally, a regression model is trained using the results of these simulators as an inquiry into how well such a system can be captured using machine learning techniques. The results of these experiments, together with the results of the other members of the initiative, improve our understanding of the influences that users bring to using these tools. Finally, two approaches to extending these models are described, and some basic prototypes demonstrated. This extension seeks to replicate the detailed geometry of the real system which many modern models simplify away. The ongoing goal of doing this is to explore the effects these details might have on the development of the hydrograph and breach.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

With some 41,000 large earthen dams across the world[4] and around 76,000 earthen dams of any size in the US alone[5] their impact on the lives and livelihood of many people is significant. Interest in accurately predicting dam breaches and the magnitude of the resulting breach hydrograph increased during the past several decades. Some of the tools developed include equations based on data from historical events[6;7] and sediment transport models such as those developed by Fread[8;9]. They have been used to accurately predict peak discharge and floodplain mapping, but they provide little information about the timing and physics of breach development or the impact of using different materials in the construction of the dam. More recently, research has focused on the response of embankment dams to both overtopping and internal erosion. Several physical model studies have been conducted. They involve a wide range of dams from relatively small-scale embankments in the laboratory[10–14], to medium-scale embankments in large test facilities[3;13;15;16], and large-scale embankments in the field[17]. Because of this research, new computational models of breach erosion processes that more accurately reflect the observed behavior of cohesive soil embankments have been developed. WinDAM (Windows Dam Analysis Modules) is a desktop application that includes modules to analyze overtopping and internal erosion of embankments, embankment breach development, and headcut erosion of earthen spillways. The individual modules and software as a whole are referred to as models. The models are tested and refined to improve model performance. On site erodibility testing techniques, called a Jet test, have been developed to better quantify the erodibility of on-site soil materials[18–20]. These tests can be used to better quantify soil parameters on site.

The development and refinement of various models continues as noted by several recent reviews and model evaluation efforts. ASCE/EWRI[21] provides a discussion of breach processes and tools available. The CEATI International Working Group on Embankment Dam Erosion and Breach Modeling[22] applied the HR Breach[23] and SIMBA[15] models to test cases including laboratory tests and actual dam failures in the field. HR Breach was developed by HR Wallingford. The CEATI report indicates that both models performed well on 5 of the 7 test cases. Zhong et al.[24] discusses the National Weather Service (NWS) BREACH model[9], the HR Breach model, and the DLBreach model[1]. HR Wallingford has since developed EMBREA[25], a successor to HR Breach. Zhong et al.[26] conducted three large-scale physical model tests at the Nanjing hydraulic Research (NHRI) in China, and developed a new model to evaluate overtopping breach processes of a cohesive dam. They compared their model with WinDAM and NWS Breach. Sensitivity analysis shows that all three models are sensitive to soil erodibility, but their proposed model and WinDAM are more sensitive than the NWS BREACH model.

Breach processes associated with overtopping and internal erosion are only imperfectly understood. To gain a better understanding, an international group started an initiative to research how these tools differ in the hands of different modelers. An interesting phase of the initiative was to have modelers focus on the analysis of just two dams – one from the lab (P1) and one from an actual failure in the field (Big Bay).

1.1 Evaluation Initiative

The P1 dam, standing at a 1.5 meters tall, is well documented in the paper *Internal Erosion and Impact of Erosion Resistance* by Gregory Hanson, Ronald Tejral, Sherry Hunt, and Darrel Temple[16]. This work investigates the failure conditions surrounding internal breaches using medium-scale models. Given the controlled nature of these collapses, there is a wealth of data collected to accurately recreate them. This research was conducted at the United States Department of Agriculture – Agricultural Research Service's Hydraulic Engineering Research Unit which is responsible for much of the foundational work on earthen embankment safety and maintenance.

The Big Bay dam in Mississippi, standing at roughly 15.6 meters tall, suffered a piping failure in 2004 which led to the collapse of the dam and damage to the downstream area. The characteristics of this breach are estimated by Steven Yochum, Larry Goertz, and Phillip Jones in *Case Study of the Big Bay Dam Failure: Accuracy and Comparison of Breach Predictions*[27]. This description uses data gathered at the time of the breach and measurements taken afterwards. these measurements may be less accurate compared to those of the P1 dam in its controlled environment. However, natural variance also contributes greatly.

The two models that this paper reports on are DLBreach and WinDAM C. DLBreach, developed by Dr. Weiming Wu of Clarkson University[1;2], is a command-line application designed to analyze earthen dam and levee breaches. A key feature of the latest version of DLBreach is the ability to simulate flow from breaching flow from either side of the embankment; a characteristic of coastal embankments. WinDAM C, developed jointly by the USDA and Kansas State University[3], is a GUI desktop application which includes several hydrologic routing functions, including the analysis of overtopping and internal erosion breaches. Aside from front end differences, these models also differ in how they represent their domain data and carry out their simulations.

To automate the processing of multiple simulation jobs, the Dakota suite developed by Sandia National Laboratories[28] is linked to these simulators via a process outlined by Mitchell

Neilsen and Chendi Cao in *Coupled Dam Safety Analysis using BREACH and WinDAM*[29]. New parsers are developed to process the input and output for DLBreach and WinDAM C.

This article presents an uncertainty analysis of DLBreach and WinDAM C for a large-scale real-world dam (Big Bay) and a medium-scale model dam (P1). This analysis includes comparing the output distributions from these simulators to one another and to real-world observations. Additionally, the individual results are ranked based on how well they model the observed conditions to investigate the per-run relationship between inputs and outputs. This research is then extended with a variance analysis to determine the significance of the influence each parameter has on the models. Finally, several optimization methods are used as a more rigorous approach to evaluating the performance of individual runs.

1.2 Model Development

Following the statistical analyses of existing models, the ongoing work into further developments with new techniques is presented. Regardless of whether a model has been continuously updated over decades or recently developed, many of them use similar methods. As classified by the ASCE/EWRI Task Committee[21] these are generally either parametric or physically-based models. In the early days, those models were limited by the kinds of computational power available. As the progress of technology has advanced, these limitations have lifted. Today however, many models still rely on simplifying assumptions that focus on more macro-scale affects.

An example of this is geometry. Embankments are given simple profiles that remain consistent across their entire length. Breaches are treated as straight channels with either slopped or shear sides, either covered or uncovered tops, and optionally may include a headcut on the downstream side. While sufficient for most applications, the systems we are trying to simulate are chaotic; small changes may lead to different results. In overtopping, the differences in height along the crest of a dam may determine where breaches form. In internal breaches, an uneven or rough channel may affect the flow.

Another example is how roof collapse is determined. These models remove the entirety of the roof when basic conditions are met. In some cases, it can be as simple as checking if the breach width exceeds twice the height of the roof material. In DLBreach, it checks whether the cohesion of the soil and support of the water can withstand the weight of the roof, but the roof is still collapsed all at once[2]. Once the roof collapses, it is typical to assume the entire mass has been washed away so it is simply removed. DLBreach does check that the mass will have been washed away before updating the geometry, but does not consider the effects of the material present in the flow. With modern technology and techniques, we may no longer need these kinds of simplifications. However, it is unclear if the increased complexity will lead to a significant boost in performance.

As a step towards investigating this, two algorithms are proposed and prototyped using computational fluid dynamics. CFD programs simulate fluid flow using numerical analysis where interactions between the fluid and its environment are controlled using boundary conditions[30]. Several CFD programs and libraries exist, but the choice of which one to use is limited by resources, availability, and flexibility. For this reason, Open-Source Field Operation and Manipulation, a free and open-source library of models[31], is used in this study. Some prior work has been done by a previous graduate student, Brian Sweeney, evaluating the usability of OpenFOAM for modeling the flow over different spillways[32]. However, this work did not model erosion, thus where these new algorithms come in.

The two algorithms have their benefits and drawbacks. The first method simulates erosion by manipulating the vertices of the faces that make up the simulation boundary. This approach smoothly deforms the terrain where erosion occurs, allowing the breach channel to have a more complex geometry. But unbounded manipulation of the mesh leads to many invalid configurations, and developing a general solution to handling these invalid solutions has a steep cost. Therefore, the second proposal defines some constraints to mesh manipulation in order to avoid entering those invalid states. Instead of manipulating the individual vertices, this algorithm essentially manipulates the faces of the surface by extruding them as new cells added to the exterior. This approach is rougher in both the quality of the geometries and the kind of motion that can be simulated which may create artifacts in the effects on the fluid flow.

Chapter 2

Evaluation Methods and Materials

2.1 Dakota

Dakota can be used for a number of simulation experiments including uncertainty analysis, parameter optimization, and sensitivity analysis. With the method outlined by Neilsen and Cao, the analysis driver for a Dakota study can be used to bi-directionally link Dakota to external programs such as DLBreach and WinDAM C. The driver is a script outlining the steps to a study. The parameters for the study then are specified in a configuration file. Each model requires their own input parameters which can be passed from Dakota to the model via a plaintext file. Dakota produces these files for each simulation run by modifying a base template. Some pre-processing may be needed to convert between units, and this can be accomplished with purpose made scripts added to the analysis driver. After Dakota runs the model, it must then parse the resulting output file. Once again, external post-processing scripts can be run via the analysis driver to accomplish this. Aside from extracting the desired values from the files, additional calculations such as unit conversions and deriving secondary metrics can be carried out during this step. The results of each run are then aggregated into a single output file.

2.2 Parameters

To describe the dams and their collapse events, three categories of parameters are given. One category is the dimensional data such as the length, width, height, and slope ratios of the dams as well as the initial dimensions of the breach. Another is the soil properties such as particle diameter, cohesion, erodibility, and critical shear stress. Finally, there is hydrologic data such as reservoir volume profiles and inlet/outlet flow over time. A list of the parameters addressed in this study is given in Table 2.1, though additional parameters were provided to modelers. The parameters are arranged into the three categories and ordered within each by how many of the model/dam scenarios use that parameter.

Table 2.1: *Ranges and averages for dam parameter values. Dashes indicate that parameter is not used for that scenario.*

<table>
<tr><th rowspan="2"></th><th colspan="2">Big Bay</th><th colspan="2">HERU P1</th></tr>
<tr><th>DLBreach</th><th>WinDAM C</th><th>DLBreach</th><th>WinDAM C</th></tr>
<tr><td>Dam Height (m)</td><td colspan="2">15.51-15.61 (15.56)</td><td colspan="2">1.2-1.4 (1.3)</td></tr>
<tr><td>Crest Width (m)</td><td colspan="2">11.6-12.8 (12.2)</td><td colspan="2">1.78-2.18 (1.98)</td></tr>
<tr><td>Breach Diameter (m)</td><td colspan="2">0.01-0.05 (0.013)</td><td colspan="2">-</td></tr>
<tr><td>Breach Height (m)</td><td>-</td><td>0.0-0.6 (0.3)</td><td colspan="2">0.9-1.1 (1.0)</td></tr>
<tr><td>Erodibility (cm^3/Ns)</td><td colspan="2">1.5-66 (33)</td><td colspan="2">23-270 (120)</td></tr>
<tr><td>Critical Shear Stress (Pa)</td><td colspan="2">1.0-5.0 (1.5)</td><td colspan="2">0.0-0.16 (0.144)</td></tr>
<tr><td>Manning's N</td><td colspan="2">0.016-0.035 (0.025)</td><td colspan="2">0.016-0.033 (0.025)</td></tr>
<tr><td>Cohesion (kPa)</td><td>5-15 (10)</td><td>-</td><td>4-5.25 (5)</td><td>-</td></tr>
<tr><td>Porosity</td><td>0.235-0.349 (0.297)</td><td>-</td><td>0.33-0.45 (0.34)</td><td>-</td></tr>
<tr><td>Soil Diameter (mm)</td><td>0.1-0.6 (0.3)</td><td>-</td><td>0.1-0.15 (0.13)</td><td>-</td></tr>
<tr><td>Friction Angle Tangent</td><td>0.577-0.675 (0.6)</td><td>-</td><td>0.577-0.675 (0.625)</td><td>-</td></tr>
<tr><td>Total Unit Weight (kN/m^3)</td><td>-</td><td>18-20 (19)</td><td>-</td><td>17.26-20.5 (18.73)</td></tr>
<tr><td>Shear Strength (kN/m^2)</td><td colspan="2">-</td><td>-</td><td>10.96-15.03 (13.02)</td></tr>
<tr><td>Clay Content (%)</td><td colspan="2">-</td><td>0.6-0.8 (0.7)</td><td>-</td></tr>
<tr><td>Reservoir Volume (m^3)</td><td colspan="2">13870000.0-17284449.6 (1557724.8)</td><td colspan="2">-</td></tr>
<tr><td>Surface Elevation (m)</td><td colspan="2">-</td><td colspan="2">0.821-1.003 (0.912)</td></tr>
</table>

Due in part to natural variance and potential imprecision in measuring, the values of these parameters are given as ranges. Alongside these ranges is an estimate of the average value which is used as the mode of a triangular distribution. These ranges and defaults are estimates based on published work[13;27]. To better visualize the difference in dimensions between the two dams, Figure 2.1 shows a comparison of their side profiles.

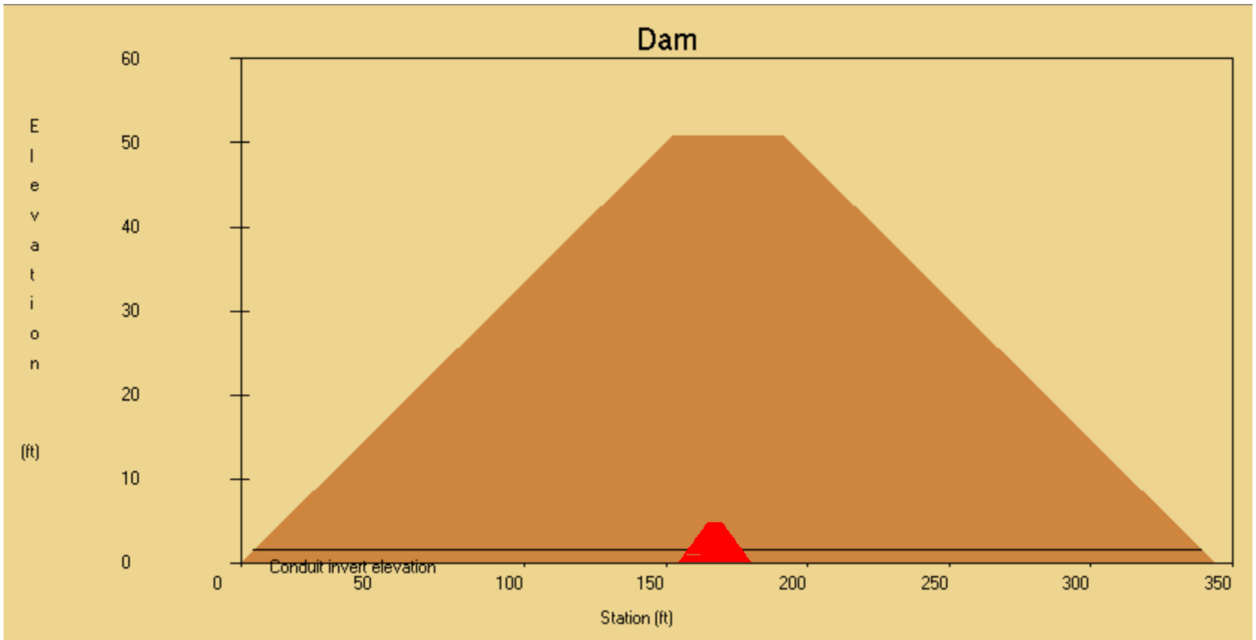

Figure 2.1: *Comparison of the side profiles of the Big Bay (brown) and P1 (red) dams.*

2.3 Uncertainty Analysis

Using Dakota's uncertainty analysis option, the experiment can be defined through a set of input parameters and output metrics. Input parameters are sampled as probability distributions determined through values specific to the type of distribution being used. For this experiment the parameters are assumed to follow a triangular distribution with lower and upper limits as well as a mode. The output metrics of interest and their observed measurements are listed in Table 2.2. For the full study, additional metrics pertaining to the collapse of the breach roof are also tracked, but won't be included here. One goal of this experiment is to determine the distributions of these output metrics as output by the models.

In addition to the inputs and outputs, another important aspect is the number of simulations to run. Enough samples must be taken such that the apparent distributions of the metrics are relatively stable. Stability, in this case, means that the numeric characteristics of these distributions are affected little by further simulations. For this study, the difference between 100 runs and 500 runs showed little improvement in the quality of the distributions therefor batches of 100 runs are used for convenience.

Table 2.2: *Observed measurements of collapse metrics.*

	Big Bay	HERU P1
Peak Flow Rate (m^3/s)	3313	2.979
Time to Peak Flow (s)	3300	1560
Max Breach Width (m)	96.15	6.46
Max Breach Depth (m)	24.7	1.25
Time to Collapse (s)	600	840
Breach Width Before Collapse (m)	0.46	1.28
Breach Width After Collapse (m)	10	1.28
Breach Depth Before Collapse (m)	0.46	0.61
Breach Depth After Collapse (m)	10	1.2

2.4 Sensitivity and Variance Analysis

The sensitivity of the model outputs to changes in the inputs is measured using the *sensemakr* library[33] in R. The minimal reporting function supplied by the library produces many values, but the primary focus will be on the coefficients of determination (R^2) and robustness values. To measure the significance of these results they will be put through a standard variance of analysis (ANOVA) also calculated with R. Of the data generated, only the p-values will be investigated. The p-value being the probability that the observed results are not caused by the changes in the independent variables. A p-value less than 0.05 (that is, the probability being less than 5%) is considered significant. For both measures, the relationship between inputs and outputs are specified as a linear combination with no covariates.

2.5 Parameter Optimization

The relationships examined in the uncertainty analysis are supplemented by a parameter optimization study. Four methods of optimization are used here; each with their own benefits and drawbacks. Three methods present practical approaches for their particular application while one represents the kind of naïve approach that researchers might investigate.

The guidelines of the initiative provide a method for ranking results based on performance functions with the aim of investigating the relationship between inputs and outputs for any given individual run. These functions use the sum of the squared natural logarithms of the proportion of simulation values to estimated values. There are three functions each using a different subset of the output metrics. The main focus will be on the first function which uses the peak flow rate, time to peak flow, and max breach width. Features of this kind of formulation that benefit analysis include making exact matches equal to zero, making all outputs non-negative, and adjusting the ranges of possible outputs below and above the target value to be equally weighted. Thus, performance values closer to zero are better. In addition to the performance function, the average percent error of all output metrics is used for the sake of comparison.

To determine the limits of model performance in a more rigorous manner, another method is employed. Dakota provides a wealth of parameter optimization approaches including several gradient-based and derivative-free methods. For this study the asynchronous pattern search method is used based on the nature of the data. This is a kind of local search, but starting with a rough estimate can improve this search by reducing the search time and aid in preventing the search from getting stuck in a local optimum. The configuration for this study has a similar format to the uncertainty experiment, but requires an objective function to optimize on. To achieve this, the provided performance function and the average percent error are added as two options for optimization.

The last two methods utilize machine learning by training regression models to emulate the results of the original models without having to carry out a full simulation. The data for this could be drawn from the uncertainty analysis, but instead new simulations are run with an even spacing between parameter values. This allows for full breadth coverage of the parameter space at varying resolutions depending on the number of divisions. As the resolution becomes finer, a better approximation of the underlying function is given. Due to the number of parameters, even three values per input can lead to tens of thousands of simulations. Therefore, three is the upper limit of divisions for this study.

Using the scikit-learn library for Python, an XGBoost model is trained with crossvalidation. A ratio of 4:1 training vs testing material is used. As will be seen, a model of sufficient quality can be attained in relatively little training time.

The first method demonstrates the naïve approach which instead trains a reverse model where metrics are the input and parameter values are the output. This method aims to create a simple function to directly get the desired values. However, the issue with this approach is the assumptions that are required for it to work as will be discussed.

The alternative method addresses these issues with a more typical approach. In this case, the model is trained normally (inputs and outputs not reversed). That model is then used as a surrogate for the original model in a standard parameter optimization search. Here, since the models are trained in Python, SciPy's implementation of dual-annealing is used for the search algorithm. Dual-annealing provides is another derivative free method that aims to find a global optimum.

Chapter 3

Evaluation Results and Discussion

3.1 Uncertainty

For the uncertainty analysis, we are interested in those densities which most closely match the observed metrics. There will also be comparisons between models and between the scales of the dams using the observational measurements as a frame of reference. Figures 3.1 and 3.2 provide plots of the probability densities of the output metrics. To quantify the relationships between densities, Table 3.1 provides the range, mean, median, mode, and standard deviation for these densities. Additionally, the average values are paired with their percent error compared to the observed measurement.

Looking at how each model performed for each dam, we can see that quite a few metrics have high percent errors. However, there are some densities that align within at least 15% of their targets. DLBreach has a good approximation of the peak flow rate and max breach width for the P1 dam, at least relative to its other approximations. WinDAM C has good alignment for the peak flow rate and breach depth for the P1 dam as well as the time to peak and max breach width for the Big Bay dam. From the history of their development, we expect these models to predict peak flow rate well so to see that this is the case, even in a limited capacity, is unsurprising. Instead, we should focus on the high errors of the other

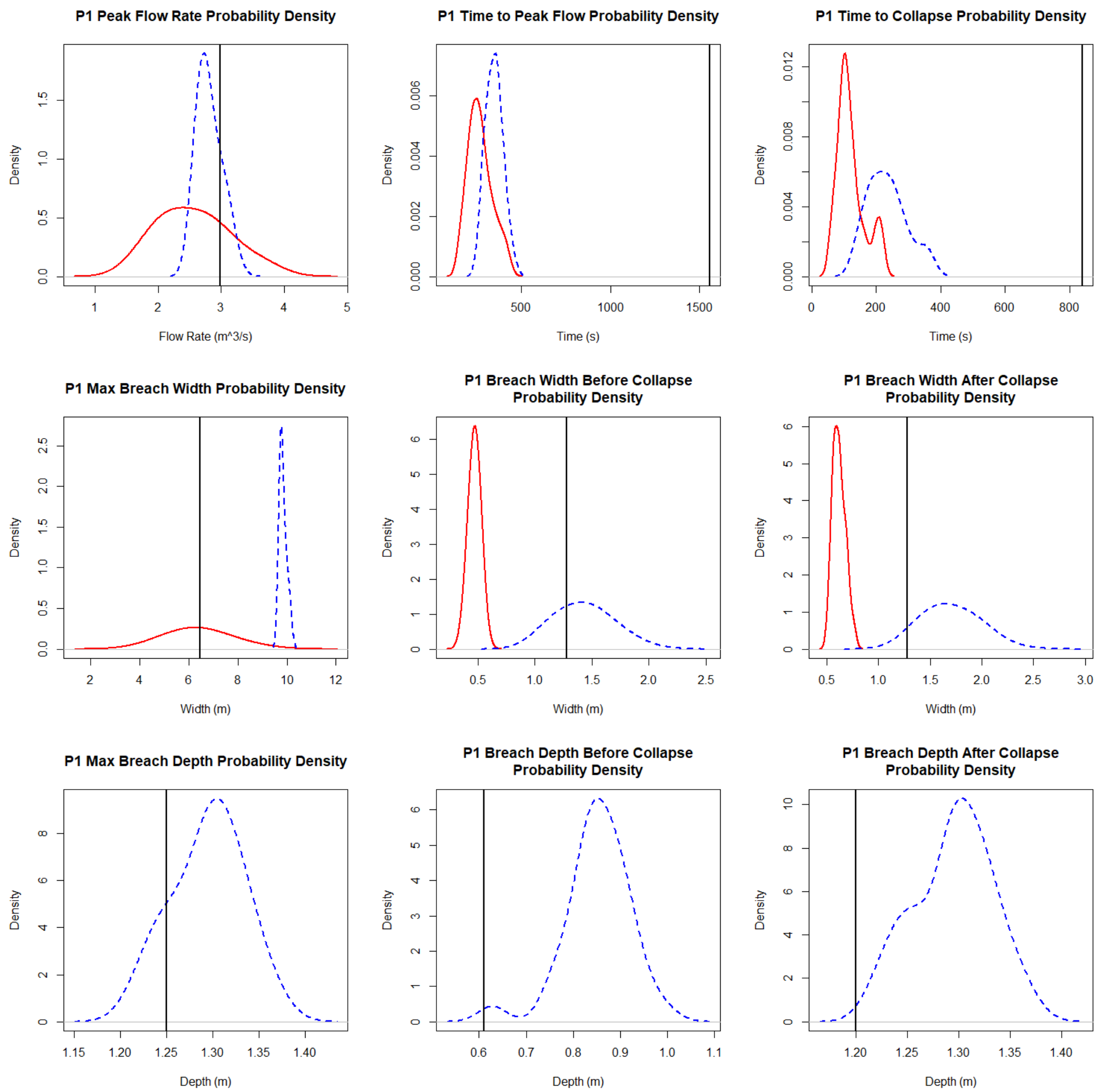

Figure 3.1: *Comparison between the DLBreach (red solid line) and WinDAM C (blue dotted line) probability densities for HERU P1.*

metrics.

When we create simplifications of complex systems, decisions must be made on what parts of the system are most import. Programs such as WinDAM C are used by professionals to design and test safe embankments. When the matter of liability is involved, it is usually safer to assume that something will fail sooner and "worse" so as not to be surprised by the suddenness or severity of a failure. As an example, we can see in Figure 3.3 that

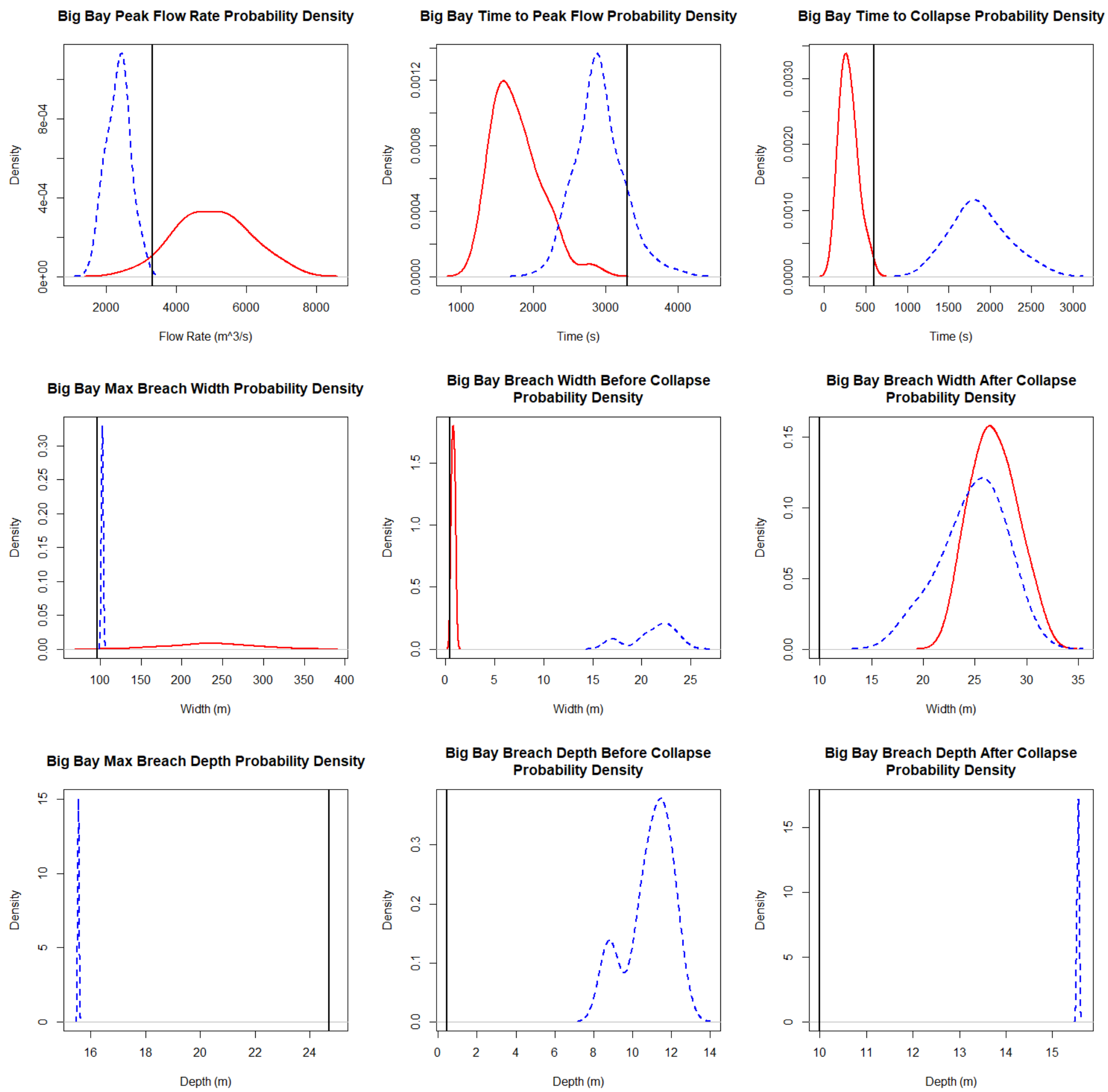

Figure 3.2: *Comparison between the DLBreach (red solid line) and WinDAM C (blue dotted line) probability densities for Big Bay.*

WinDAM C's predictions are designed to encompass the worst-case scenario for a wide range of conditions[34]. Some results from the field will be quite close to those predictions while others will be significantly "tamer" by comparison.

These kinds of decisions are not limited to the development of models. Both modelers and data gatherers out in the field must make their own decisions. Modelers, for instance, are given a single input to represent the height of a dam, but earthen dams are not always

Table 3.1: *DLBreach and WinDAM C probability density characteristics. Mean, median and mode also include their percent error to the observed measurements.*

HERU P1	DLBreach					WinDAM C				
	Range	Mean	Median	Mode	SD	Range	Mean	Median	Mode	SD
Peak Flow Rate (m^3/s)	1.58-3.93	2.56 (14%)	2.53 (15%)	2.19 (26%)	0.568	2.55-3.25	2.81 (6%)	2.77 (7%)	2.73 (8%)	0.182
Peak Time (s)	155-430	267 (83%)	260 (83%)	245 (84%)	67.2	252-468	347 (78%)	360 (77%)	360 (77%)	47.8
Max Breach Width (m)	4.33-9.05	6.38 (1%)	6.29 (3%)	6.15 (5%)	1.13	9.75-10.2	9.85 (52%)	9.76 (51%)	9.76 (51%)	0.132
Max Breach Depth (m)						1.21-1.38	1.29 (3%)	1.3 (4%)	1.3 (4%)	0.038
Collapse Time (s)	59.8-220	122 (85%)	110 (87%)	103 (88%)	42.7	144-360	235 (72%)	216 (74%)	216 (74%)	60.9
Breach Width B.C. (m)	0.38-0.55	0.473 (63%)	0.478 (63%)	0.498 (61%)	0.0366	0.98-2.09	1.43 (12%)	1.4185 (11%)	1.49 (16%)	0.246
Breach Width A.C. (m)	0.50-0.78	0.619 (52%)	0.613 (52%)	0.591 (54%)	0.0613	1.27-2.37	1.7 (33%)	1.66 (30%)	1.53 (20%)	0.24
Breach Depth B.C. (m)						0.622-1.00	0.5 (18%)	0.85 (39%)	0.84 (38%)	0.068
Breach Depth A.C. (m)						1.21-1.375	1.29 (8%)	1.3 (8%)	1.3 (8%)	0.038
Big Bay	**DLBreach**					**WinDAM C**				
	Range	Mean	Median	Mode	SD	Range	Mean	Median	Mode	SD
Peak Flow Rate (m^3/s)	2730-7230	5040 (52%)	5110 (54%)	5280 (59%)	1040	1590-3030	2350 (29%)	2380 (28%)	2460 (26%)	325
Peak Time (s)	1260-2860	1780 (46%)	1690 (49%)	1560 (53%)	337	2160-3960	2920 (12%)	2880 (13%)	2880 (13%)	18
Max Breach Width (m)	128-331	235 (144%)	234 (143%)	233 (142%)	45.3	101-104	103 (7%)	103 (7%)	103 (7%)	0.689
Max Breach Depth (m)						15.5-15.6	15.55 (37%)	15.56 (37%)	15.56 (37%)	0.022
Collapse Time (s)	125-565	295 (51%)	275 (54%)	242 (60%)	105	1440-2520	1870 (212%)	1800 (200%)	1800 (200%)	308
Breach Width B.C. (m)	0.46-1.22	0.813 (77%)	0.81 (76%)	0.771 (68%)	0.174	16.6-24.7	21.1 (4487%)	21.7 (4617%)	22.3 (4748%)	2.27
Breach Width A.C. (m)	22.3-31.8	26.9 (169%)	26.7 (167%)	26.3 (163%)	2.18	18.5-30.0	24.8 (148%)	25.4 (154%)	25.9 (159%)	2.87
Breach Depth B.C. (m)						8.68-12.55	10.9 (2270%)	11.18 (2330%)	11.5 (2400%)	1.139
Breach Depth A.C. (m)						15.5-15.6	15.55 (56%)	15.56 (56%)	15.56 (56%)	0.022

perfectly even across their crest. Another example is the materials used for the dams; both models assume a homogeneous soil, but some dams are made with different layers or with a special core. One way of addressing these differences in our models is the processes demonstrated here. This does not perfectly replicate the effects of a slump in a dam or a core of a different material, but we hope that the distributions of the results approximate reality in some way. For data gatherers, one of the decisions they make is when to start measuring time. The models likewise start at some point in the middle of the breach-forming process. The discrepancy between these two starting points could explain some of the high errors for the time-related metrics. Regardless of how they perform individually, there is also the question of how the models and dams compare.

The comparison between models generally favors WinDAM. On average, it more closely matches the observed measurements. This is sometimes even the case when the measurement in question doesn't fall within the range of outputs from the model. The exceptions are P1's max breach width and Big Bay's time to collapse and breach width before collapse where DLBreach has a lower error. WinDAM also tends to have higher average values, particularly for P1 results. However, this does not equate to being any more or less constrained. The

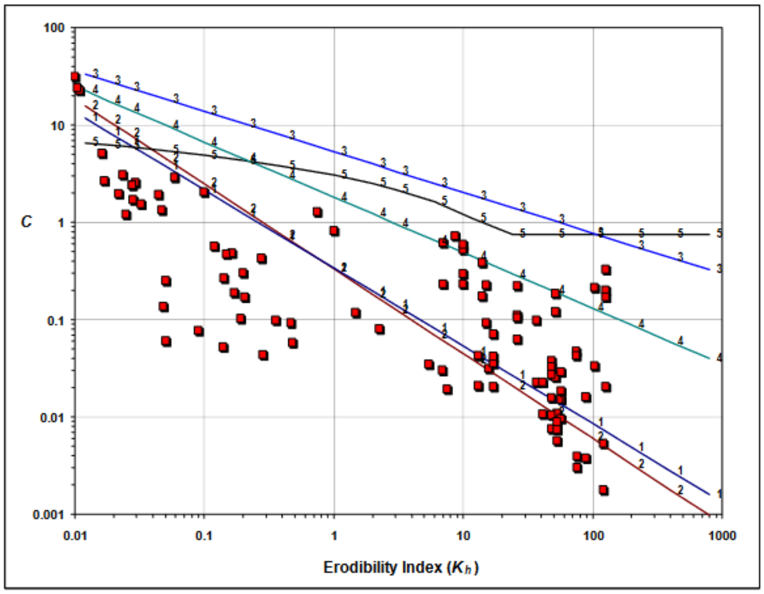

Figure 3.3: *Comparison of historical results and WinDAM predictions*[34].

better measure of constraint is the range and standard deviations. WinDAM tends to have narrower ranges and standard deviations for the peak flow rate, time to peak flow, and max breach width while DLBreach is narrower for the other non-depth metrics. However, there are also similarities. Both models underestimate the peak flow rate and time to collapse for P1 as well as the time to peak for both dams. They also overestimate all of the breach widths for Big Bay.

The comparison between scales favors the P1 dam. The exceptions being the time to peak and time to collapse for DLBreach as well as the time to peak and max breach width for WinDAM. For both models, the results would suggest that the error improves for the peak flow rate and the before/after collapse breach widths as the scale decreases. It also appears to improve for the time to peak flow as the scale increases. However, it's difficult to support these patterns from two data points. Remembering Figure 3.3, it is possible that either dam may represent a particularly good or bad example of its type. Instead of extrapolating into a pattern based on scale, these results may demonstrate the kind of accuracy ranges that these programs can produce.

It should be noted here that only WinDAM provides a reliable way of determining the breach depth from the output files. These metrics range from some of the most accurate in the case of the P1 max breach depth to some of the least accurate such as the Big Bay breach depth before collapse. In nearly every simulation run, the full height of the dam is always eroded, so the max breach depth is not necessarily of interest. However, the collapse metrics are a different story.

Generally, the collapse statistics are less accurate than the other metrics. One possible explanation for this could be the limited nature of how roof collapse is modeled. As previously discussed, there are many areas where roof collapse has been simplified: the triggering condition, the way it occurs all at once, and what happens to the roof material upon collapse. The way collapse is triggered would obviously affect the time metric, but it could also affect the dimensions of the breach at the time before collapse. Then there is also how material is handled after collapse. If it manages to settle on the bottom even briefly then that would affect the post collapse dimensions. It would also likely affect how the flow progresses.

Figures 3.4 and 3.5 show the development of the flow rates and breach width for key simulation runs. These runs include the max flow rate, minimum flow rate, and several runs with optimal metrics. The optimal metrics runs were determined using the optimization methods evaluated later. Comparing between scales, we see that for P1 the optimal runs for DLBreach are spread out between the min and max runs while for Big Bay they favor the max run. For WinDAM, the optimal results are more in the middle for both scales. For the P1 dam, it is interesting to note that the flow rate graphs look very different. Part of this difference is caused by the inflow data provided to the models. The inflow for P1 is significant enough to greatly alter the outflow charts. Based on the difference in time, it is likely that DLBreach is not considering the inflow data, thus this would be the major cause. Another factor could be how roof collapse is handled. Remember that when the roof collapses, the geometry is not updated till it is determined that the roof material has been swept away.

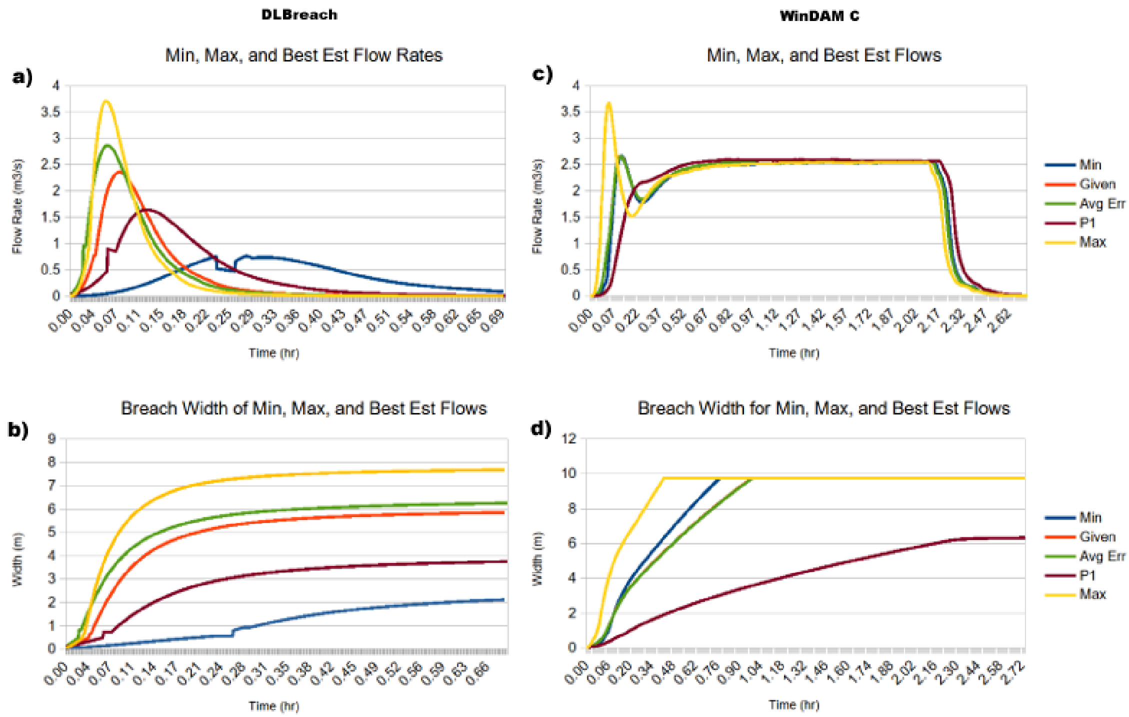

Figure 3.4: *The development of the flow rate and breach width for HERU P1 in both DLBreach (a,b) and WinDAM (c,d).*

3.2 Sensitivity

The minimal reporting function in sensemakr generates a coefficient estimate, standard error, and t-value, as well as a partial R^2 and two robustness values. However, only the partial R^2 and robustness values will be evaluated. Tables 3.2 and 3.3 provide some of the most significant results for the four dam/model scenarios. The full results are available in Appendix A. These values give an idea of the strength of the relationship between the parameters and metrics. It is clear that peak flow rate, time to peak flow, and max breach width are highly sensitive to soil erodibility. Additionally, the peak flow rate is sensitive to reservoir volume/-surface level. We can also see that the three metrics are sensitive to Manning's roughness coefficient, but only for DLBreach. For max depth, the only sensitive parameter is the dam height which makes intuitive sense. It is likely that because the depth reached the max height of the dam in most runs, there is not enough information to determine what other

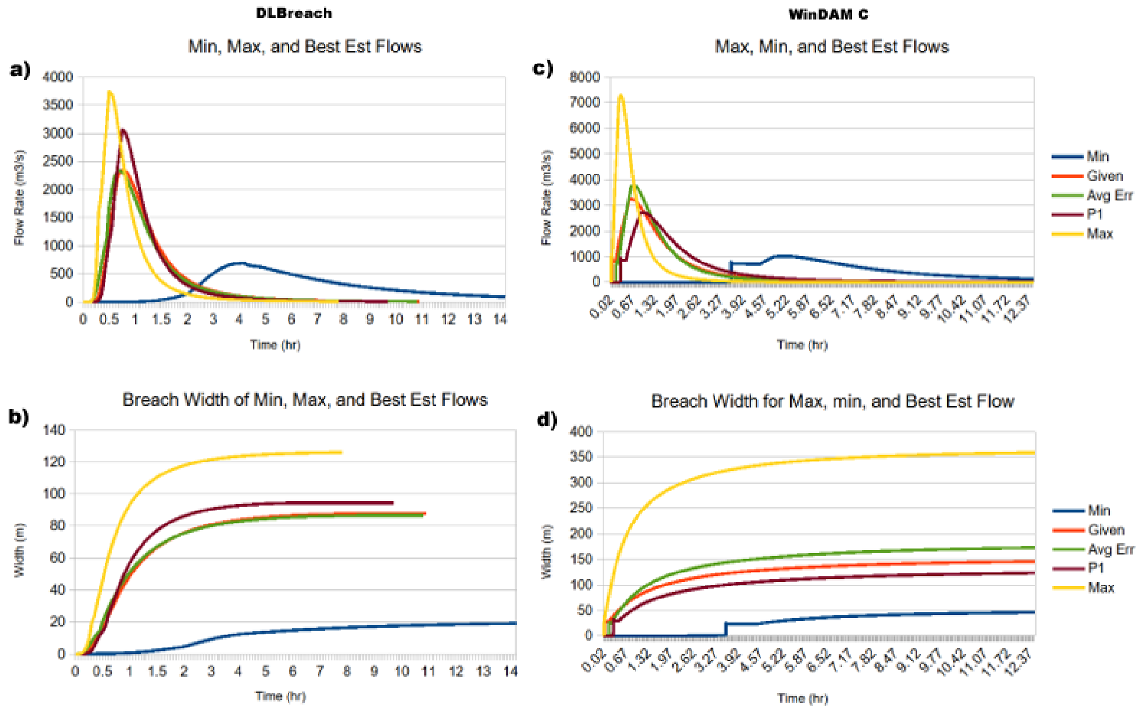

Figure 3.5: *The development of the flow rate and breach width for Big Bay in both DLBreach (a,b) and WinDAM (c,d).*

parameters would affect this. In the collapse statistics we see that time to collapse is sensitive to erodibility, breach width before collapse is sensitive to cohesion, and breach depth after collapse is once again sensitive to dam height.

These results support what is already known to be the most influential parameters, but it is likely an incomplete picture. Again, there are only two dams with fairly similar soil types. As such, this part of the analysis is just an evaluation of what metrics are sensitive to which parameters for a fairly limited set of circumstances. What this data is primarily useful for is determining what parameters should be focused on in the experimentation. For instance, Parameters such as clay content show next to no influence and thus computational resources would be best spent elsewhere. However, there is still an issue with these sensitivity results The relationship of the inputs to outputs that is provided to the analysis function is linear without covariates. Parameters such as erodibility and cohesion though are not completely independent. A better formulation would account for these and other relationships between

Table 3.2: *Non-collapse sensitivity results for the four dam/model pairs. The order of values in each cell is P1 DLBreach, P1 WinDAM, Big Bay DLBreach, then Big Bay WinDAM C.*

	Peak Flow Rate			Time to Peak		
	$R^2_{Y\ D\|X}$	$RV_{q=1}$	$RV_{q=1,\alpha=0.05}$	$R^2_{Y\ D\|X}$	$RV_{q=1}$	$RV_{q=1,\alpha=0.05}$
Dam Height	16.5, 30.5, 1.8, 1	35.6, 47.8, 12.5, 9.7	18.6, 31.1, 0, 0	28.2, 4.1, 12.6, 4.4	46, 18.6, 31.4, 19.3	32, 0, 10.1, 0
Crest Width	0, 1.5, 0, 1.1	2, 11.5, 0.4, 10.1	0, 0, 0, 0	0.2, 4.5, 2.7, 0.2	4, 19.5, 15.4, 4	0, 0, 0, 0
Breach Height	1.1, 42.3, -, 8.4	10.1, 56.5, -, 26.1	0, 42.9, -, 3.8	14.2, 49.2, -, 3.8	33.2, 61.3, -, 18.1	15.5, 49.5, -, 0
Reservoir Contents	89.9, 78.4, 87.1, 95.5	90.8, 81.6, 88.4, 95.7	89.4, 77.2, 86.2, 95.2	42.3, 67.8, 10.1, 37.3	56.5, 74, 28.4, 52.9	45.6, 66.9, 6.1, 39.4
Erodibility	97.5, 95.5, 96.2, 98.2	97.5, 95.7, 96.4, 98.3	97.3, 95.1, 96, 98.1	90.9, 90.4, 88.2, 69.7	91.6, 91.1, 89.3, 75.3	90.5, 89.6, 87.4, 69.3
Manning's N	95.9, 3.3, 95.3, 1.2	96, 17, 95.5, 10.5	95.7, 0, 95, 0	38.3, 0.7, 76, 0	53.6, 7.9, 79.8, 0.8	41.9, 0, 75.1, 0
Cohesion	1.7, -, 35, -	12.3, -, 51.3, -	0, -, 36.7, -	5.2, -, 10.9, -	20.8, -, 29.3, -	0, -, 7.3, -
	Max Width			Max Depth		
	$R^2_{Y\ D\|X}$	$RV_{q=1}$	$RV_{q=1,\alpha=0.05}$	$R^2_{Y\ D\|X}$	$RV_{q=1}$	$RV_{q=1,\alpha=0.05}$
Dam Height	0.3, 70.8, 0, 57	5.2, 76.1, 1, 66.6	0, 69.8, 0, 57.6	-, 100, -, 99.9	-, 100, -, 99.9	-, 100, -, 99.9
Crest Width	0, 27, 0.8, 66.4	0.2, 45.1, 8.4, 73	0, 27.4, 0, 66.2	-, 1.2, -, 0	-, 10.6, -, 0.9	-, 0, -, 0
Breach Height	0.1, 0, -, 85.3	3.1, 1.5, -, 87	0, 0, -, 84.5	-, 2.7, -, 0.3	-, 15.4, -, 5.7	-, 0, -, 0
Reservoir Contents	29.4, 0.1, 10.9, 20.8	47, 2.3, 29.4, 39.8	33.3, 0, 7.4, 21.9	-, 0.2, -, 0.8	-, 4.6, -, 8.6	-, 0, -, 0
Erodibility	97.8, 4.1, 96.3, 89	97.8, 18.6, 96.4, 90	97.6, 0, 96, 88.3	-, 2.2, -, 0.1	-, 14, -, 2.7	-, 0, -, 0
Manning's N	97.1, 0.5, 96.5, 1.2	97.2, 6.8, 96.6, 10.4	97, 0, 96.2, 0	-, 0.5, -, 1.8	-, 7, -, 12.8	-, 0, -, 0
Cohesion	24, -, 41.3, -	42.6, -, 55.8, -	27.5, -, 42.8, -	-, -, -, -	-, -, -, -	-, -, -, -

Table 3.3: *Collapse sensitivity results for the four dam/model pairs. The order of values in each cell is P1 DLBreach, P1 WinDAM, Big Bay DLBreach, then Big Bay WinDAM C.*

	Time to Collapse			Width Before		
	$R^2_{Y\ D\|X}$	$RV_{q=1}$	$RV_{q=1,\alpha=0.05}$	$R^2_{Y\ D\|X}$	$RV_{q=1}$	$RV_{q=1,\alpha=0.05}$
Dam Height	24.5, 12.7, 39.3, 6.1	43, 31.5, 54.3, 22.5	28.1, 9, 40.8, 0	27.9, 7.9, 1.6, 6.8	45.8, 25.3, 12, 23.6	31.7, 0.6, 0, 0.4
Crest Width	0.1, 3.1, 2.8, 2	3.2, 16.4, 15.5, 13.3	0, 0, 0, 0	1.4, 3.4, 4.1, 0.1	11.3, 17, 18.7, 3.5	0, 0, 0, 0
Breach Height	12.3, 27.5, -, 1.4	31, 45.4, -, 11.3	12.7, 27.9, -, 0	24.3, 2.4, -, 0	42.9, 14.6, -, 0.3	27.9, 0, -, 0
Reservoir Contents	33.2, 22, 0.8, 0.5	49.9, 40.8, 8.8, 7.1	37, 21.6, 0, 0	29.2, 0.1, 0.3, 1.9	46.9, 3, 5.4, 13	33.1, 0, 0, 0
Erodibility	88.2, 60.6, 73.6, 64	89.3, 69, 78.1, 71.4	87.6, 60.2, 72.8, 64.1	40.6, 1, 0.7, 16.5	55.3, 9.8, 8, 35.7	44, 0, 0, 16.4
Manning's N	3, 0.8, 2.1, 0	16, 8.8, 13.6, 1.2	0, 0, 0, 0	0.6, 1.1, 0.6, 0.1	7.7, 9.9, 7.4, 2.6	0, 0, 0, 0
Cohesion	25, -, 11.6, -	43.5, -, 30.2, -	28.7, -, 8.5, -	96.8, -, 99.4, -	96.9, -, 99.4, -	96.6, -, 99.4, -
	Width After			Depth After		
	$R^2_{Y\ D\|X}$	$RV_{q=1}$	$RV_{q=1,\alpha=0.05}$	$R^2_{Y\ D\|X}$	$RV_{q=1}$	$RV_{q=1,\alpha=0.05}$
Dam Height	1.1, 7.9, 10.3, 7.2	10, 25.3, 28.6, 24.3	0, 0.6, 6.4, 1.3	-, 100, -, 99.9	-, 100, -, 99.9	-, 100, -, 99.9
Crest Width	1.3, 3.4, 0.2, 0.1	10.8, 17, 4, 2.7	0, 0, 0, 0	-, 1.2, -, 0.3	-, 10.6, -, 5.4	-, 0, -, 0
Breach Height	45.8, 2.8, -, 0	58.9, 15.5, -, 0	48.7, 0, -, 0	-, 2.7, -, 0.2	-, 15.4, -, 4.8	-, 0, -, 0
Reservoir Contents	0, 0, 1.7, 1.6	1.1, 0.7, 12.4, 11.8	0, 0, 0, 0	-, 0.2, -, 0.7	-, 4.6, -, 8.1	-, 0, -, 0
Erodibility	12.4, 3.7, 1.1, 42.1	31.2, 17.8, 10, 56.4	12.9, 0, 0, 44	-, 2.2, -, 0.2	-, 14, -, 4.3	-, 0, -, 0
Manning's N	1.1, 1, 0.2, 0.1	10, 9.5, 3.8, 2.7	0, 0, 0, 0	-, 0.5, -, 2.7	-, 7, -, 15.3	-, 0, -, 0
Cohesion	29.1, -, 99.2, -	46.8, -, 99.2, -	33, -, 99.1, -	-, -, -, -	-, -, -, -	-, -, -, -

parameters.

3.3 Variance

Table 3.4 provides the statistical significance of the parameters for the output metrics. Significance is defined as a p-value less than 0.05, meaning that there is a less than 5% chance that the same results would occur otherwise. The exact values are provided in Appendix A. Looking at the non-collapse metrics, we see that the effects of initial breach height, water levels, and erodibility on peak flow rate and time to peak are consistently significant. The effects of dam height, crest width, and erodibility on max breach width is also consistently significant. Finally, Manning's coefficient significantly affects all these metrics for DLBreach, and dam height significantly affects max breach depth for WinDAM. The general relationships conform to the findings of earlier phases, and some of the more specific relationships can be readily explained. It is likely that DLBreach makes greater use of Manning's coefficient in the internal erosion process, and we already know that the breach generally fully erodes the embankment so the relationship between height and depth makes sense.

Table 3.4: *Parameter significance for P1 DLBreach, P1 WinDAM C, Big Bay DLBreach, and Big Bay WinDAM C. "O", "x", and "-" are used to mean the parameter is either significant ($p < 0.05$), not significant, or not used for that scenario respectively.*

	Peak Flow	Peak Time	Breach Width	Breach Depth	Collapse Time	Width BC	Width AC	Depth BC	Depth AC
Dam Height	O,x,O,x	O,O,x,x	x,O,O,O	-,O,-,O	O,O,O,O	O,x,x,O	O,x,O,O	-,x,-,O	-,O,-,O
Crest Width	x,x,O,O	x,x,O,x	x,O,O,O	-,O,-,x	x,x,x,x	x,x,x,x	x,x,x,x	-,x,-,x	-,O,-,x
Breach Diameter	-,-,x,O	-,-,O,O	-,-,x,O	-,-,-,x	-,-,O,O	-,-,x,x	-,-,x,x	-,-,-,x	-,-,-,x
Breach Height	O,O,-,O	O,O,-,O	O,x,-,O	-,x,-,x	O,O,-,O	O,x,-,x	O,x,-,x	-,O,-,x	-,x,-,x
Reservoir Contents	O,O,O,O	O,O,O,O	O,x,x,O	-,x,-,x	O,O,x,O	O,x,x,O	x,x,x,O	-,O,-,O	-,x,-,x
Erodibility	O,O,O,O	O,O,O,O	O,x,O,O	-,x,-,x	O,O,O,O	O,x,x,O	O,x,x,O	-,x,-,O	-,x,-,x
Shear Stress	x,x,O,x	x,O,x,x	x,x,O,x	-,x,-,x	x,x,x,x	O,x,x,x	x,x,O,x	-,x,-,x	-,x,-,x
Manning's N	O,x,O,x	O,x,O,x	O,x,O,x	-,x,-,O	x,x,x,x	x,x,x,x	x,x,x,x	-,x,-,x	-,x,-,O
Cohesion	O,-,x,-	O,-,x,-	O,-,O,-	-,-,-,-	x,-,x,-	O,-,O,-	O,-,O,-	-,-,-,-	-,-,-,-
Particle Diameter	O,-,O,-	x,-,x,-	x,-,O,-	-,-,-,-	x,-,x,-	O,-,x,-	x,-,O,-	-,-,-,-	-,-,-,-
Porosity	x,-,x,-	x,-,x,-	x,-,x,-	-,-,-,-	x,-,O,-	O,-,O,-	O,-,x,-	-,-,-,-	-,-,-,-
Friction Angle	x,-,O,-	x,-,x,-	x,-,x,-	-,-,-,-	x,-,x,-	x,-,x,-	O,-,O,-	-,-,-,-	-,-,-,-
Unit Weight	-,x,-,O	-,O,-,x	-,x,-,x	-,x,-,x	-,O,-,x	-,O,-,x	-,x,-,x	-,x,-,x	-,x,-,x
Shear Strength	-,O,-,-	-,O,-,-	-,x,-,-	-,O,-,-	-,x,-,-	-,x,-,-	-,x,-,-	-,x,-,-	-,O,-,-
Clay Content	x,-,-,-	O,-,-,-	x,-,-,-	-,-,-,-	O,-,-,-	O,-,-,-	x,-,-,-	-,-,-,-	-,-,-,-

Looking at the collapse metrics we see less significant relationships, but there are a few worth pointing out. For one, DLBreach shows significance for the effects of cohesion and friction angle on the collapse widths. Both of these factors play into the roof collapse calculations, but so does the total unit weight. It may be because the weight does not vary enough that its results do not show significance. The variance in dam height is also shown to be significant for breach depth after the collapse. This suggests that by the time of collapse the breach has likely already reached the bottom. Finally, the variance in dam height, breach height, water levels, and erodibility are all shown to be significant for the time to collapse. This likely has to do with roof collapse inequalities between the support and weight. For instance, a larger dam may have a heavier roof.

Similar to the sensitivity analysis, the ANOVAs are complicated by several factors. For one, while the distributions of the metrics closely resemble normal distributions, they contain anomalous features which may prevent them from being true normal distributions. Since an ANOVA of this type typically requires a normal distribution, it is uncertain the degree to which these "bumps" affect that. While more simulations may improve matters thanks to the central limit theory, some of these bumps may be caused by violations of relationships between the input parameters. Factors such as erodibility and cohesion are not fully independent, and unrealistic combinations of values for these parameters are likely to skew the results. Again, these dependent relationships are not reflected in the linear formula used for analysis. To improve this, a proper function formulated from the interactions of these parameters would have to be used instead. This, of course, requires knowledge of the domain. Finally, ANOVAs done in this way assume that the independent variables are categorical, which the input parameters are not. Together, the covariate and continuous natures must be addressed for more accurate results.

3.4 Parameter Optimization

The four optimization methods are evaluated based on how they are used. The first method is provided with the uncertainty analysis, and so it's focus is on how the relationship between inputs and outputs of an individual run can supplement the relationship found as a whole. The Dakota method by comparison is focused on the absolute limits of the relationship between the results and the observed measurements. The last two methods utilize machine learning, and they focus on the efficiency of getting the results.

3.5 Ranked Uncertainty Runs

The probability density results give an idea of the overall behavior of the models. However, these results are detached from the capabilities of any particular run. That is, it is not clear how close any single run is to matching the peaks of these distributions, and more importantly, how close any single run is to matching the observed measurements. For this purpose, investigating the performance of individual runs is necessary to understand this relationship.

Table 3.5: *Top five runs for each scenario/model pair based on performance function score.*

	1st	2nd	3rd	4th	5th
P1 DLBreach	1.45	1.46	1.46	1.49	1.49
P1 WinDAM C	1.28	1.35	1.35	1.36	1.36
Big Bay DLBreach	0.38	0.54	0.60	0.61	0.65
Big Bay WinDAM C	0.19	0.28	0.28	0.29	0.29

The first performance function provided as part of the initiative is calculated for every run of each dam/model pair. Table 3.5 provides the scores for the top five runs. There is a clear difference between scales and models. These scores show that individual runs for the larger dam are significantly closer to the observed measurements. It can also be seen that WinDAM C generally has a smaller difference. The difference between this observation and that of the uncertainty run is the difference between the majority of runs collectively being closer versus the best individual runs being closer. To break down these results, Table 3.6 provides the percent error of the output metrics for the top scoring runs. As can be seen, these errors are quite high thus explaining some of the higher scores. These values support some of the earlier observations, but we also see that time to peak in particular is especially different from the observed measurements. While the depth and collapse metrics are not used in the calculation of the performance function, they are included. It is not particularly surprising to see that their error is quite high; not just because they were not optimized for, but because they have given similar results in the prior sections. We can also see that some of the Depth metrics have a low error, once again attributed to the fact that there is little variance possible there.

Table 3.6: *Percent error of output metrics compared to their measured values for top ranked runs.*

	Peak Flow	Peak Time	Breach Width	Breach Depth	Collapse Time	Width BC	Width AC	Depth BC	Depth AC
P1 DLBreach	77.6	263	37.6		282	200	123		
P1 WinDAM	15	233	34.1	4.2	133	15.8	25	25.9	8.1
Big Bay DLBreach	21.6	15.6	25.1		20.1	29.7	65.1		
Big Bay WinDAM	12.3	14.6	6.3	59.2	58.3	97.3	55.1	94.9	35.5

While the model outputs are the main interest in such an investigation, it is also useful to investigate the inputs. By examining where these values lie in their ranges and how they compare to the provided inputs, we can complete our understanding of how these models function in comparison to the system they are modeling. Table 3.7 provides the percent error of the inputs compared to their provided values. It should be noted here how far off certain parameters, such as the reservoir volume for the Big Bay scenarios, are from their provided values. This might suggest that the measurements were off, or that the models do not accurately reflect the impact of these parameters in their equations.

While looking at the single best scores for each scenario gives an idea of the upper limit of performance, extending the view reveals additional patterns. Here we focus on the distributions of input parameters and output metrics formed from those runs which scored in the top 25%. Comparing these distributions to those of the full set of runs we see that, on average between the four scenarios, the range of values for erodibility is about 30% that of the provided range. Other parameters such as crest height and width as well as water levels use around 60% of their ranges. These are generally much narrower than the other parameters which use 90-100% of their ranges. This could further support some of the variance and sensitivity results in suggesting that those parameters with narrower ranges have a greater influence. Of course, there are other factors that could explain this clustering effect such as the sizes of the provided ranges and the number of runs done for each scenario.

Table 3.7: *Percent error of input parameters compared to their estimates for top ranked results.*

	P1 DLBreach	P1 WinDAM	Big Bay DLBreach	Big Bay WinDAM
Dam Height	4.6	0	0.3	0.4
Crest Width	2	0.5	0.8	0.8
Erodibility	46.9	10.8	45.2	35.8
Critical Shear Stress	41.3	4.2	152	50
Manning's N	2	8.4	24.4	1.6
Reservoir Contents	1.6	4.9	933.6	818
Breach Height	2.6	66.3		73
Breach Diameter			119.2	11.5
Cohesion	12		19	
Porosity	12.4		2.4	
Particle Diameter	5.4		8	
Friction Tangent	2.4		2.7	
Unit Weight		0.4		3.7
Shear Strength		104.3		
Clay Content	90.3			

3.6 Asynchronous Pattern Search

As outlined earlier, an asynchronous pattern search is used for a more rigorous optimization method. Table 3.8 provides the average percent errors and performance function scores for the results of each parameter set found. The parameter sets themselves are provided in Appendix B. Additionally, a set of percent error calculations using only the performance score metrics is included to allow a better comparison between the two optimized sets.

It is important to note that the estimates provided with the initial data result in neither the best performance score nor average percent error. This matches the results found earlier with the density graphs, and gives another metric for quantifying those differences.

What these results demonstrate is that the choice in which output metrics to optimize upon clearly matters. What is harder to determine is the effect that the formulation of the objective functions has on the parameters that are found. A better comparison could be drawn if the average percent error of the performance function metrics was used as the alternative objective function instead of the average error of all output metrics.

Table 3.8: *Percent error of input parameters compared to their estimates for top ranked results.*

		Given Values	Best Avg. Error	Best Score
HERU P1 DLB	Average error	50.77	46.51	55.13
	Perf. metrics error	37.03	29.96	51.08
	Perf. Score	1.72	1.96	1.46
HERU P1 WDC	Average error	30.24	18.72	29.66
	Perf. metrics error	44.92	41.21	42.48
	Perf. Score	1.35	0.65	0.67
Big Bay DLB	Average error	89.05	58.81	57.03
	Perf. metrics error	80.82	41.53	18.82
	Perf. Score	1.164	0.69	0.35
Big Bay WDC	Average error	89.05	58.81	57.03
	Perf. metrics error	80.82	41.53	18.82
	Perf. Score	1.16	0.69	0.35

Aside from comparing the outputs, we can also compare the different sets of parameters that were found. The parameters which are most similar between the two suggested sets includes dam height, erodibility, water level, Manning's coefficient, particle diameter, and friction angle. From this, and the earlier variance evaluations, we can infer there are likely two reasons for this pattern. For one, those parameters with a strong influence will greatly change the results for any small change in the parameter value. Therefore, to obtain optimal results, regardless of the differences in the two objective functions, these values are limited to a small range of possibilities. On the other hand, if a parameter has little to no influence, then there is no reason to change their value. What we are left with are likely parameters which have a medium effect on the results and thus can be tweaked to optimize for one objective over another.

This also provides another method of comparing models and the effects of embankment size. Once again, we see some of the same patterns present in the uncertainty and rankings analyses. Generally, it would seem that WinDAM C produces lower average percent errors and better performance scores. This is expected at this point, but something else that also appears in these results, and perhaps is clearer here, is the difference in scales. At the small scale, different parameters are changed between functions. For WinDAM, the percent error and performance scores are generally worse at the smaller scale. Some of this difference may be due to differences in the error of measurement between the two models.

3.7 Regression Models

The two methods using machine learning vary in practicality. As outlined in the methodology, both methods use regression models trained on regularly-spaced samples of the parameter space. Figure 3 shows the graphs for the fit of regressions models trained on the performance scores for each of the four scenarios. Provided in the caption are the root mean squared errors (RMSE). As can be seen there is a good relationship between predicted value and actual value suggesting promising results for an optimization search.

Table 3.9 provides the percent error of output metrics for the parameters suggested by the reverse, surrogate, and Dakota methods. Only Big Bay is shown here as these last two methods are meant as a simple demonstration. While the results for the reverse function shown here appear relatively similar to the other two methods, it should be noted that this was a kind of best-case scenario. Other attempts of using this method are fraught with complications that will be discussed later. Provided in Appendix B are the parameter sets found by each method as well as an invalid set found by the inverse function method. Instead, the similarity between the results of the surrogate model with dual-annealing vs the original models with asynch pattern search are of greater interest. It should be noted that pattern search is a local method while dual-annealing is a global method which could contribute to some of the poorer performance of the Dakota results.

The dual-annealing technique demonstrates the efficacy of using surrogate models in this field. However, as employed here, it comes at the cost of additional work to train the model. More practical methods exist, including those implemented in Dakota, if parameter

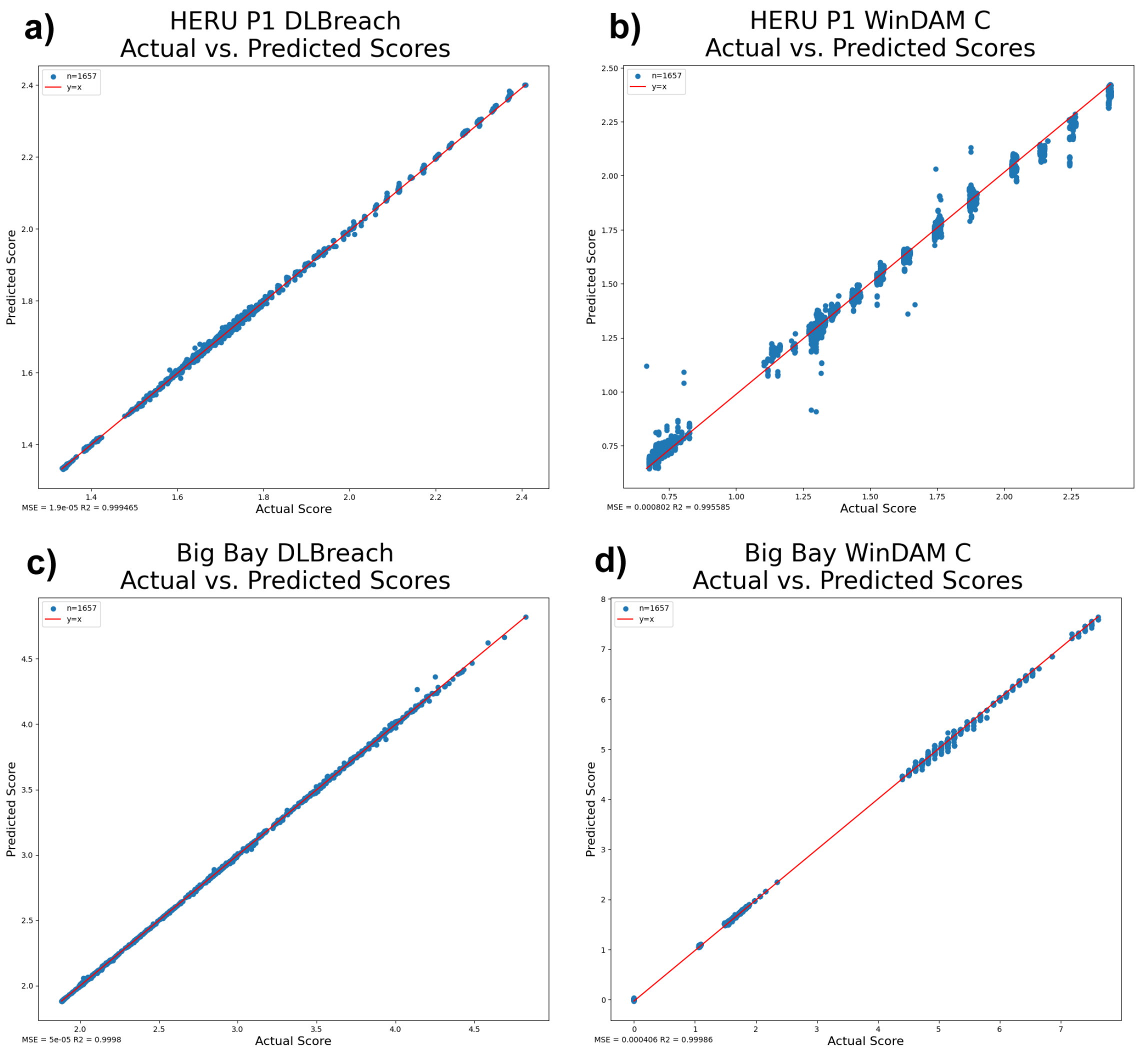

Figure 3.6: *Regression model Perf1 scores vs predicted Perf1 scores for (a) HERU P1 DLBreach (MSE = 0.000019, R2 = 0.99946), (b) HERU P1 WinDAM C (MSE = 0.0008, R2 = 0.99558), (c) Big Bay DLBreach (MSE = 0.00005, R2 = 0.9998), and (d) Big Bay WinDAM C. (MSE = 0.0004, R2 = 0.99986).*

optimization is the only desire. Otherwise, a better application of this technique may be in training the models on historical and laboratory data to attempt to capture the intricacies that may be lost in our manually-developed models.

The inverse function approach on the other hand demonstrates the issues with poor adaptation of machine learning techniques. The goal with this approach is to get a simple model that can be fed the desired output metrics and receive input metrics as a result. In

Table 3.9: *Big Bay metric percent errors for the parameter sets found by three different optimization methods.*

		Reverse	Dual-Annealing	Dakota
DLBreach	Peak Flow	5.9	23.5	19.7
	Time to Peak	6.1	3.6	36.2
	Breach Width	49.5	40.3	97.8
WinDAM C	Peak Flow	17	4.6	7.7
	Time to Peak	23.6	34.5	1.8
	Breach Width	6.6	7.6	7.2

order for such a method to work as intended, there needs to be a one-to-one relationship between inputs and outputs for the original function such that the it is reversible. Such a relationship does not exist for the models trained here. Additionally, we desire the inputs to be within the ranges that are provided, but this method does not have that constraint. Simply clamping to the nearest limit may not necessarily work either depending on the complexity of the underlying function. Finally, this method shows the kind of pitfall that is perhaps most dangerous: it works for one scenario. This can encourage developers to pursue ill-fit options.

Chapter 4

CFD Methods and Materials

OpenFOAM provides models, here referred to as solvers, for a wide range of systems. There are, of course, 2-dimensional and 3-dimensional fluid solvers. Then there are heat, electromagnetism, chemical reaction, and even economics solvers. Being the most obvious use case, the 2D and 3D fluid solvers have a wealth of options including: compressible vs incompressible, laminar vs turbulent, buoyancy, single or multiple phases, and so on. A phase is a set of properties associated with a fluid. There can be multiple phases for the same fluid, differentiated by unique parameter values.

The attributes of OpenFOAM simulations are defined in several plaintext files. These include the type of simulation, the geometry, and the initial conditions. The *controlDict* file contains settings for how simulations are executed. The solver to be used, the start and end times, the length for time steps, the frequency at which the current state should be written off to disk, and several other options are specified here. The test simulations in this paper use the incompressibleVoF solver for incompressible, isothermal immiscible fluids. The time step length is set to 0.05 seconds, and simulations are run for 3 seconds.

OpenFOAM has a unique format for defining geometry. In a field such as 3D rendering, an environment is usually modeled as the objects that occupy it. However, in fluid simulations, the space between objects is modeled instead. This will be referred to as the domain.

The interior of the domain is divided into cells. These cells can be any polyhedron, but a number of limitations makes hexahedrons the preferred shape. This format is specified in the *blockMeshDict* file.

The *blockMeshDict* has three main sections. The first section is an ordered list of the 3D points (vertices) that lie on the surface of the domain. The indices of these vertices are then used to define basic shapes which can be divided into smaller cells by specifying the number of subdivisions along each axis. Finally, the boundaries of the domain are defined using those same vertex indices. An simple cube with 1,000 cells and with a single named boundary is provided below as an example.

```
vertices
(
        (0.0 0.0 0.0) // 0
        (0.1 0.0 0.0) // 1
        (0.1 0.1 0.0) // 2
        (0.0 0.1 0.0) // 3
        (0.0 0.0 0.1) // 4
        (0.1 0.0 0.1) // 5
        (0.1 0.1 0.1) // 6
        (0.0 0.1 0.1) // 7
);

blocks
(
        hex (0 1 2 3 4 5 6 7) (10 10 10) simpleGrading (1 1 1)
);

boundary
```

```
(
    outlet
    {
        type patch;
        faces
        (
        (1 2 6 5)
        );
    }
);
```

Notice the order that the vertices are listed for the block and the boundary face. This order imparts directionality to the corresponding faces. A face can either be pointing inside or outside its parent block. Faces on a boundary should point outside the domain while internal faces should point from a subordinate block to a superior block. This set of constraints organizes the cells into a tree which can be traversed by the simulation program.

While only one boundary patch is explicitly defined in the example, all external faces are assigned to a boundary patch. Any face not listed as part of a boundary patch will be assigned to a default patch that is automatically created. Separating parts of the boundary in this way allows the user to control how the model should treats these areas.

The attributes for the fluids used in these simulations are spread across several files. The first of these is the *phaseProperties* file which contains a list of phase names as well as constant values pertaining to all phases. For each phase listed here, a *physicalProperties*.[*phase*] file is also required. These files contain the viscosity model and constants specific to a particular phase. The tests in this paper have a phase for "water" and "air" which both use a constant model requiring viscosity (ν) and density (ρ) constants.

The initial state of the simulation is defined in the "0" folder which also includes several files. These files define fields, multi-dimensional data structures that associates numerical values to each cell and external face. Examples of fields include the dynamic pressure (p_rgh), velocity (U), and the phase fraction (α).

In a single-phase models, all cells are assumed to completely contain a single fluid. In a multi-phase model, like the ones used for this paper, every cell contains a ratio of fluids. This proportion is the α field. The overall fluid properties of a cell are a combination of the physical properties of all fluids proportional to how much of each is in the cell. An example of the *alpha.water* file is provided below:

```
dimensions      [ 0 0 0 0 0 0 0 ];
internalField   uniform 1;
boundaryField
{
    outlet
    {
        type            inletOutlet;
        inletValue      uniform 0;
        value           uniform 0;
    }
    default
    {
        type            zeroGradient;
    }
}
```

In this example, the cube defined in the previous example is filled with water by setting the *internalField* value to 1. The boundary conditions define how the water phase should be treated for the walls and outlet. The dimensions parameter specifies the units of a given field by listing the exponents for each basic SI unit. As an example, the dimension value for velocity (m/s) is [0 1 -1 0 0 0 0].

Simulations are executed by solving a set of equations for the current state of the fields using the previous state. For this reason, the "resolution" of the domain (size of the cells) is important as that determines the density of points in these fields. There are, however, limitations to this approach. Data such as the exact, physical boundary between phases is lost.

If such properties are needed, then the flexible nature of the software can allow for a number of workarounds. For instance, if the topology of the boundary is important, dynamic refinement of the cells can be applied. This will shrink the size of cells in heterogeneous regions and will increase the size of cells in homogeneous regions with the goal of focusing computation and accuracy to those places which are likely of interest. There are a number of solvers which affect the mesh. Most often this is some cyclic manipulation of the mesh. They also often are one-directional with the motion of the mesh imparting changes to the fluid, but the fluid having no effect on the mesh. OpenFOAM is capable of local, acyclic, and bi-directional mesh updates. However, there is little work done on such simulations with nothing provided as part of the base project. This leads to another option for circumventing the limitations of the software; the open-source code freely available for anyone to edit for their own projects.

Chapter 5

OpenFOAM Models

Current models implemented in programs such as WinDAM C and DLBreach represent breaches primarily as a continuous channel maintaining a simple profile from one side of the dam to the other, possibly with an additional head cut profile. These flat surfaces allow for simpler erosion calculations on a macro scale which are often suitable for their intended purposes. However, there is the question of how the more complex geometries of reality affect flows and the development of breaches. It would also be nice to take something like a scan of a dam and import it directly into a computer model. This geometry would then be a more accurate replication of how overtopping and spillway erosion might occur. These are the motivations for a method of modeling the local changes to geometry.

There are a number of ways in which the desired motion solver might be implemented. Two will be proposed here alongside their benefits and drawbacks. It should once again be noted that these kinds of models are far outside the usual use cases for OpenFOAM. There is little support for such a model, and there is little to no work on this topic.

5.1 Vertex Manipulation

Regardless of the technique, the surface of the mesh will need to be updated as a response to a local property of the fluid. One option is to manipulate the location of the cell vertices. We can start by outlining the basic mechanic of this algorithm. At the end of the fluid simulation phase, the erosion motion is performed. For each surface vertex, some calculation is done to determine the amount to be eroded. This quantity is then translated into a displacement of the vertex along its normal vector or, in other words, "away" from the model. The properties of the neighboring cells would then be updated to reflect the effects that such a motion would have. From this base description we can then work out the details.

For one, how would this erosion calculation be performed? The erosive force of each cell which shares this vertex needs to be calculated. When developing this model, the average pressure of the cells was used as a stand-in for the erosive forces. However, this should be replaced by one of the standard equations of erosion. Programs such as WinDAM C calculate the shear stress acting on the surfaces which can be acquired from the acceleration component parallel to the surface and the mass of the fluid or fluids (although air is usually negligible). This is used with the erodibility and critical shear stress properties of the surface material to determine the amount eroded. This calculation usually includes a step that translates the eroded volume to a distance by which the surface "retreats" to expand the volume of the breach. Since the way in which volume changes as a surface of the shapes moves is different to how it changes when a vertex moves, this step will need to be replaced. The possible configurations that any vertex may be in with its neighbors is particularly vast so a general solution to how movement of a vertex along its normal vector changes the volume of the cells may also be rather complex. The normal of a vertex in this case is calculated as the average of the normal vectors of the faces which share that vertex.

Since OpenFOAM is already well equipped to handle updating the fluids based on motion in the mesh, this portion of the algorithm can be handled by native implementations. However, determining how this is done is not necessarily straightforward. Different implementations of motion solvers handle this step slightly differently depending on what tools they used for the mesh motion. An example of this is the difference between the polyTopoChange and polyTopoChanger classes. One method may use the first directly, and another may use the later as a wrapper around it and thus handle propagation of updates differently. An example of how the vertex method can handle propagation is by looking at the dynamic refiner from earlier. Since this method also resizes cells, it would make sense that they share similar mechanisms. However, this solver does not do some of the things the dynamic refiner does, and, as will be seen, it does some things that the refiner doesn't.

5.2 Vertex Manipulation Prototype

A prototype following the described algorithm was developed as a mesh topology solver. When the solver object is created, a list of patch IDs is used to specify which boundaries can be manipulated so that things like inlets and outlets don't get warped. A mesh modifier must be created for each patch that will be manipulated. These modifiers are existing modules provided as part of the codebase. After the fluid simulations are finished, the update function is called on this solver.

The update function iterates through the vertices of the modifiable boundary patches and does the calculations to determine which will move and how. The resulting list of new vertex locations is then applied to the mesh using the modifiers API. The API handles updating cell fields after motion.

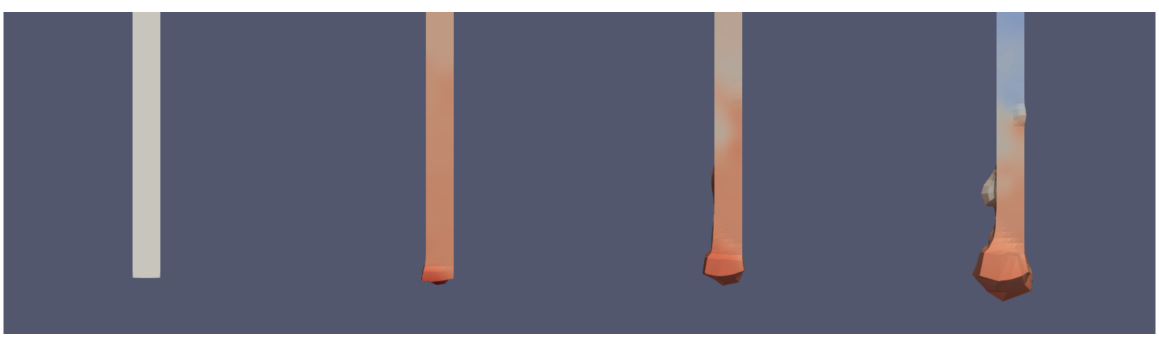

Figure 5.1: *Example of the vertex manipulation solver reacting to changes in pressure.*

Figure 5.1 depicts this model being used in a basic test. For this test, the stand-in for the erosive force is the pressure normal to the surface. The example domain is a meter tall tube where water flows in from one side at the top and exists from the opposite side at the bottom. Pressure is primarily exerted against the bottom of the tube where the water splashes down. It can clearly be seen that the areas of higher pressure begin to bulge out.

5.3 Vertex Manipulation Issues

So far, the basic algorithm has only considered the case where a vertex is minutely disturbed from its original position with no obstacles. However, even in quite ordinary scenarios changes will likely accumulate in a localized region leading to drastic differences between neighbors. When cells are stretched in this manner, they can become exceedingly large or pointed. This can pose a problem under the limits for a valid mesh under OpenFOAM. Often these cells will need to be split to maintain stability. OpenFOAM has some built-in mechanisms for adding and removing cells. However, these are primarily done through groups of cells that form an entire layer throughout the mesh. This limitation makes sense given some of the constraints on what constitutes a valid mesh. For one, faces can only be shared by at most two cells so splitting a single cell requires special handling of the neighboring cells.

Another issue that must be addressed is potential intersections as cells grow into one another. There are a number of common scenarios where we'd expect this to occur: parallel flows combining, internal breaches breaking through to the surface, etc. If one were to break down all the ways in which one cell could potentially intersect with any others the number of configurations quickly grow. This greatly increases the work necessary to implement such a model and the main limiting factor in implementing the motion solver in this way.

5.4 Face Extrusion

The primary problem with the first method, despite it seeming at first like an obvious and straightforward approach, is that it lacks constraints to maintain the qualities of the mesh which make those simple operations simple in the first place. As such, while running the model may have been tractable with modern hardware, developing it likely would not be. When planning a new model then, it is important to have some key characteristics which are maintained regardless of the operations performed. The qualities which this new approach will maintain are very simple: every cell will be a cube limited to a regular grid in 3D space, any operations on the mesh will be to add a cell that is aligned to that grid as if a face is being extruded.

These constraints address some of the key problems that stem the issue described earlier. For one, not deforming the shapes of the cells prevents any cell from becoming so stretched as to break the limits of OpenFOAM. When manipulating the vertices, it was necessary to determine how much a vertex needed to move to gain a specific volume for the erosion calculations. With the only operations being the addition or removal of known volumes, this step is greatly simplified. Finally, since the addition of cells is limited to a regular grid, the intersection problem is also greatly simplified. When a new cell is added that would connect existing cells, the faces that would be shared are already aligned one-to-one.

The basic algorithm is similar to the one outlined earlier with a few different considerations. Where before the erosive forces were calculated for each surface vertex, they will now be calculated for each yet-to-exist cell on the surface of the mesh. This requires calculating the forces of each face that is used by that cell. As mentioned previously, we only need the volume to be eroded, which can be compared against the volume of cell to be added. Since the mesh does not technically move, the propagation of motion to the fluid is handled slightly differently. More importantly, any cells that are added need to be instantiated with the properties used in calculations.

What fluid should it be filled with? In the case of a single-phase simulation there is no options and thus no question. However, in the multiphase case there are multiple options with their own implications. If the cell is filled with air, this “bubble” can a quite noticeable effect on the surrounding flow. Whatever solid was there previously has presumably been pushed out by the surrounding flow, and thus filling the vacant space with air is not necessarily realistic. Filling the cell with water instead is not unprecedented. When a mass is eroded in current simulations, it is usually the case that the mass is assumed to be completely washed away and replaced with water. But currently the simulation has no concept of a reservoir and thus no pool of resources to pull this liquid from. Even if it did, that water would be teleported to this space instantly as the fluid that should have filled that space has already been allocated elsewhere. These together essentially mean that it would be making water out of nothing which is also not very realistic. Another option then would be to use the water in surrounding cells. This is still not perfect, but the water is not created out of nothing, and the space is at least partially filled rather than left as a bubble. Ideally, the water that would have filled that space should have done so during the initial fluid calculations instead of traveling to other cells and thus affecting their calculations. Regardless, as can be imagined, this reallocation of water requires updates to other properties such as pressure.

5.5 Face Extrusion Prototype

The next prototype is very similar to the first. It uses a similar mesh topology changer object as the base. This base is then modified to determine which cells need to be added as outlined in the algorithm. Here, a purpose-built mesh modifier must be employed. This modifier is similar to the cellLayerAdditionRemoval modifiers provided with the codebase, but it instead adds specific cells rather than an entire layer of cells. This custom modifier must handle updating the fluid based on how the mesh is moved.

Figure 5.2 demonstrates the same test model using the new face extrusion model. As can be seen here, adding new cells requires careful consideration of the order that vertices are ordered in the new cell. If the vertices aren't in the correct order, then the cube will be malformed.

Once this was made to work, the pressure surrogate was replaced with a more typical

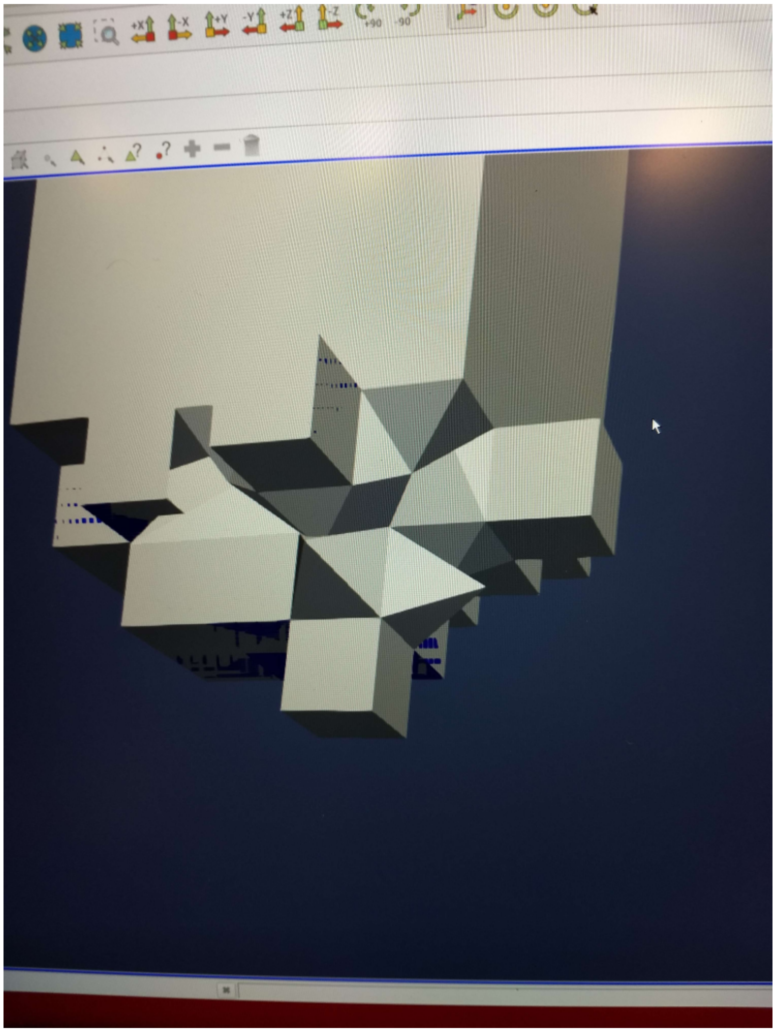

Figure 5.2: *Example of the face extrusion solver reacting to changes in pressure.*

erosion equation. Generally, the erodibility is multiplied against the difference between the shear stress against the surface and the critical shear stress of the surface material. The shear stress in turn is calculated as the force applied over the face of the breach. The force involves the volume of fluid, the density of the fluid, and gravity proportional to the slope of the surface. Instead of the full face of the breach, we will use the face of a cell. Additionally, the acceleration taken from the proportion of gravity is replaced with the acceleration of the fluid in that cell.

5.6 Face Extrusion Issues

The astute will notice something glossed over in the description of the algorithm; the comparison of the eroded volume to the volume of the cell. When allowing continuous volumes to be eroded - as is the case with existing simulations - then any solution the erosion calculations can be accepted. When limited to discrete volumes, the results of the erosion equation may not be an exact unit. Most of the time it is in fact the case that for any given minute time step, not enough erosion will happen on a face for a whole cell to be eroded. Here, we can use another built in feature of OpenFOAM. Those properties which are associate with cells and faces are known as fields, and it is possible for users to define their own fields. Thus, it is possible to accumulate the erosion that should occur for each face. If the sum of accumulated erosion on all faces of a yet-to-be cell is greater than or equal to the volume of a cell, then that cell should be added, and the remaining accumulated erosion shared upon the external faces of that new cell. This is obviously not a perfect solution, and is part of a number of problems with this approach.

Simplifying the model to maintain stability has its cost. In both current models and the vertex method outlined earlier, erosion is reflected immediately in the geometry of the simulation and thus affects the flow. In this cell-based method, if erosion is slow, then it will take several time steps for it to be reflected in the geometry. If erosion is faster than one cell per step, then it will take several steps to catch up unless the check for erosion is run multiple iterations per time step. Something else the vertex method does better than the cell-based method is that changes along a surface transition smoothly as opposed to 90 degree turns. Those sharp edges and the pockets created by new cells affect the flow in unusual ways.

5.7 Full Model Roadmap

So far, the only development that has been done is in producing prototypes of the basic algorithm. The models currently available have taken many people many years to develop, and although this work is meant only to be a minor extension, it too will take more time. The next steps of development are to implement the custom fields for erosion accumulation and handling. From there, a more refined erosion equation can be implemented. So far, only a handful of the parameters that modern models can use are available for these prototypes. These steps should allow for testing with basic spillway and over-toping modeling. At this stage, it will be desirable to implement parallelism. OpenFOAM has some support for parallelism with their fluid solvers, but the mesh solvers will need to be adjusted to utilize this functionality. With parallelization, domains with finer resolutions can be run in suitable time. One of the major limitations of this model is that it cannot smoothly replicate the sloped sides of embankments, and with those finer resolutions the effects of the cubic grid may be reduced. When these kinds of simulations are successful, internal erosion can be addressed by adding handling of roof collapse. A benefit of this method over current models is that roof collapse can be calculated locally. It can often be the case that parts of the downstream roof collapse before the rest of the roof does. These partial collapses affect flow and add substrate into suspension.

5.8 Further Model Extensions

Finally, there are a couple ways this model may be extended further. First and foremost is the ability to simulate material carried in suspension. One approach would be to use different phases to represent different loads within the flow. This method is rough and may not accurately capture the behavior of particles interacting in the fluid. However, OpenFOAM is actually capable of simulating probabilistic particle clouds. These clouds are influenced by the surrounding fluids, and collisions with boundaries can be tracked in heatmaps. This functionality may be useful for evaluating how the material in suspension contributes to erosive forces. As previously discussed, one of the major assumptions in current models is that collapsed materials are fully washed away. The effects of this could be simulated with dense clouds injected into the fluid at the time of collapse. How much of that material actually gets carried away and how much settles out of solution could be investigated with the next development.

Both model extension proposals have the useful feature of being reversable. That is, they can be made to add material instead of eroding it. This may be useful for the sudden addition of large quantities of suspended material or for when fluids slow down. As they slow down, their carrying capacity is reduced, causing material to settle. One place this can occur is downstream from a breach as it spreads out after exiting.

Finally, another important possible extension is the use of partial cells. Cells that do not fully occupy the cubic grid can be used to represent slopes and partially eroded cells. The addition of this functionality greatly increases the possible configurations of interactions between adjacent cells. However, it would likely still be more viable than the vertex manipulation method while overcoming several limitations of the basic extrusion method.

Chapter 6

Conclusions

The international initiative has straightforward goals, and these results provide straightforward answers. From the uncertainty results and simulation run rankings we can see that WinDAM C is, on average, more performant. While some metrics are better matched for the P1 dam, others are better matched for Big Bay. The variance and sensitivity analysis seems to support known behavior in that parameters such as erodibility and the water levels have a greater influence. Finally, while it may be intentional for the purposes of safety, accuracy of these models can be rather low. In addition to the margin of error used for safety, a major contributing factor is their simplicity. What these results show is that there is more work to be done.

Going forward, there are some potential refinements that could be made for the analysis process. The first being a more consistent experimental design. Primarily, it would benefit the comparisons between models and scales, as well as other elements of the analysis, if the parameters used in these simulations were limited to a select subset which has shown significance in the variance analysis and which can also be used by all dam/model pairs. From there, a far greater number of simulation iterations may be beneficial in confirming the distributions found here. While something like 500 simulation runs was tested, it showed little difference from the 100 runs. If there are any benefits to be gained it is likely that the number of runs needs to be in the thousands. From there, the central limit theory will hopefully ensure we get well formed normal distributions. For the optimization sections, the choice of optimization functions is also inconsistent. Perhaps a better comparison might be made between the performance score and the average percent error of only the performance function metrics.

Additional work could also be done with regression models. The use of surrogate models is a known technique with well implemented options available. The fit of the regression models on just these summary metrics shows promise, but these alone have limited use. Perhaps a more useful application would be training models to accurately recreate the overtime characteristics such as how the flow rate and breach develop. Aside from training them on the output from models, it may also be worth training them on historical and laboratory data. Hopefully, these kinds of models might improve our own understanding of the underlying systems.

Finally, the models we have work, but there are ways we can expand upon them. When doing so, we must put them through the same rigorous evaluations that have already been carried out so as to ensure that something of value has been gained. With techniques such as CFD modeling, we may be able to improve accuracy. We may be able to use these simulators in ways that we previously couldn't such as importing models of real dams. Still, they need work to reach a useable state.

Bibliography

[1] Weiming Wu. Simplified physically based model of earthen embankment breaching. *Journal of Hydraulic Engineering*, 139(8):837–851, Aug 2013. doi: 10.1061/(asce)hy.1943-7900.0000741.

[2] Weiming Wu. Introduction to dlbreach a simplified physically-based dam/levee breach model. Technical report, Department of Civil and Environmental Engineering, Clarkson University, 2016.

[3] Sherry L. Hunt, Darrel M. Temple, Mitchell L. Neilsen, Abdelfatah A. Ali, and Ronald D. Tejral. Windam c: Analysis tool for predicting breach erosion processes of embankment dams due to overtopping or internal erosion. *Applied Engineering in Agriculture*, 37(3):523–534, 2021. doi: 10.13031/aea.14334.

[4] Icold cigb, Apr 2023. URL https://www.icold-cigb.org/GB/world_register/general_synthesis.asp.

[5] Usace, 2020. URL https://nid.sec.usace.army.mil/#/.

[6] Tony L. Wahl, Jeff A. Farrar, Jim Melena, Danielle Mares, and Brent W. Mefford. Prediction of embankment dam breach parameters a literature review and needs assessment. In *DSO-98-004 Dam Safety*, Denver, CO, 1998. U.S. Dept. of the Interior, USBR, Dam Safety Office.

[7] Christopher I. Thornton, Michael W. Pierce, and Steven R. Abt. Enhanced predictions for peak outflow from breached embankment dams. *Journal of Hydrologic Engineering*, 16(1):81–88, Jan 2011. doi: 10.1061/(asce)he.1943-5584.0000288.

[8] Danny L. Fread. *Dambrk: The NWS dam-break flood forecasting model*. Hydrologic Research Laboratory, National Weather Service, NOAA, Silver Spring, MD, 1984.

[9] Danny L. Fread. *Breach, an erosion model for earthen dam failures*. Hydrologic Research Laboratory, National Weather Service, NOAA, Silver Spring, MD, 1988.

[10] Mohamed Hassan, Mark Morris, Gregory J. Hanson, and Karim Lakhal. Breach formation: Laboratory and numerical modeling of breach formation. In *State Dam Safety Officials Association Proceedings*, Lexington, KY, 2004. ASDSO.

[11] Tony L. Wahl, Dale Lentz, Bruce Feinberg, and Bobby Rinehart. Physical hydraulic modeling of canal breaches. Technical report, U.S. Dept. of Interior, USBR, Dam Safety Office, Water Resources Research Lab, Denver, CO, 2011.

[12] Mahmoud Al-Riffai. *Experimental study of breach mechanics in overtopped noncohesive earthen embankments*. PhD thesis, University of Ottawa, 2014.

[13] Muhammad Ashraf, Ahmed Hussein Soliman, Entesar El-Ghorab, and Alaa El Zawahry. Assessment of embankment dams breaching using large scale physical modeling and statistical methods. *Water Science*, 32(2):362–379, Oct 2018. doi: 10.1016/j.wsj.2018.05.002.

[14] T. Alvarez, S. Valente, S. Amaral, T. Viseu, and R. M. Ferreira. Sensitivity analysis to the pilot channel geometry in dam breach by overtopping. *38th IAHR World Congress - Water: Connecting the World*, Sep 2019. doi: 10.3850/38wc092019-5555.

[15] D. M. Temple, G. J. Hanson, M. L. Neilsen, and K. R. Cook. Simplified breach analysis for homogeneous embankments: Part 1, background and model components. In *U.S. Soc. on Dams. Technologies to Enhance Dam Safety and the Environment*, page 151–161, Westminster, Co, 2005.

[16] G. J. Hanson, R. D. Tejral, S. L. Hunt, and D. M. Temple. Internal erosion and impact of erosion resistance. In *U.S. Soc. on Dams. Technologies to Enhance Dam Safety and the Environment*, page 773–784, Westminster, CO, 2010.

[17] K. A. Vaskinn, A. Lovoll, K. Hoeg, M. Morris, and G. J. Hanson. Physical modeling of breach formation: Large scale field tests. In *Assoc. State Dam Safety Officials Annual Meeting*, Lexington, KY, 2004.

[18] G. J. Hanson and K. R. Cook. Apparatus, test procedures, and analytical methods to measure soil erodibility in situ. *Applied Engineering in Agriculture*, 20(4), 2004. doi: 10.13031/2013.16492.

[19] G. J. Hanson and K. R. Cook. Determination of material rate parameters for headcut migration of compacted earthen materials. In *Assoc. State Dam Safety Officials Annual Meeting*, Lexington, KY, 2004.

[20] G. J. Hanson and S. L. Hunt. Lessons learned using laboratory jet method to measure soil erodibility of compacted soils. *Applied Engineering in Agriculture*, 23(3):305–312, 2007. doi: 10.13031/2013.22686.

[21] ASCE/EWRI Task Committee on Dam/Levee Breaching. Earthen embankment breaching. *Journal of Hydraulic Engineering*, 137(12):1549–1564, Dec 2011. doi: 10.1061/(asce)hy.1943-7900.0000498.

[22] CEATI. Evaluation of numerical models for simulating embankment dam erosion breach processes. In *DSO-2017-02*. U.S. Depart. of the Interior, USBR, Dam Safety Development Program, 2017.

[23] Hr breach model, 2020. URL http://www.floodsite.net/html/HR_Breach_Model.htm.

[24] Qiming Zhong, Weiming Wu, Shengshui Chen, and Meng Wang. Comparison of simplified physically based dam breach models. *Natural Hazards*, 84(2):1385–1418, Aug 2016. doi: 10.1007/s11069-016-2492-9.

[25] Matthew West, Mark Morris, and Mohamed Hassan. A guide to breach prediction. *Dams and Reservoirs*, 28(4):150–152, Dec 2018. doi: 10.1680/jdare.18.00031.

[26] Q. M. Zhong, S. S. Chen, Z. Deng, and S. A. Mei. Prediction of overtopping-induced breach process of cohesive dams. *Journal of Geotechnical and Geoenvironmental Engineering*, 145(5), May 2019. doi: 10.1061/(asce)gt.1943-5606.0002035.

[27] Steven E. Yochum, Larry A. Goertz, and Phillip H. Jones. Case study of the big bay dam failure: Accuracy and comparison of breach predictions. *Journal of Hydraulic Engineering*, 134(9):1285–1293, Sep 2008. doi: 10.1061/(asce)0733-9429(2008)134:9(1285).

[28] Brian Adams, William Bohnhoff, Keith Dalbey, Mohamed Ebeida, John Eddy, Michael Eldred, Russell Hooper, Patricia Hough, Kenneth Hu, John Jakeman, and et al. *Dakota a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis version 6.15 users manual.* Sandia National Laboratories, Nov 2021.

[29] Mitchell Neilsen and Chendi Cao. Coupled dam safety analysis using breach and windam. In *15th International Conference on Scientific Computing*, page 82–85, 2017.

[30] Patrick A. McMurtry, Todd C. Gansauge, Alan R. Kerstein, and Steven K. Krueger. Linear eddy simulations of mixing in a homogeneous turbulent flow. *Physics of Fluids A: Fluid Dynamics*, 5(4):1023–1034, Apr 1993. doi: 10.1063/1.858667.

[31] Openfoam features, 2024. URL http://www.openfoam.org/features/.

[32] Brian P. Sweeney. Converged stepped spillway models in openfoam. Master's thesis, Kansas State University, 2014.

[33] C. Cinelli, J. Ferwrda, C. Hazlet, and A. Rudkin. Sensemakr: Sensitivity analysis tools for regression models, Oct 2021. URL https://cran.r-project.org/web/packages/sensemakr/sensemakr.pdf.

[34] Mitchell L. Neilsen, Darrel M. Temple, and Johannes L. Wibowo. Analysis of spillway erosion rate parameters, Jan 2006.

Appendix A

Extended Tables

Table A.1: *Full non-collapse sensitivity results for the four dam/model pairs. The order of values in each cell is P1 DLBreach, P1 WinDAM, Big Bay DLBreach, then Big Bay WinDAM C.*

	Peak Flow Rate			Time to Peak		
	$R^2_{Y\ D\|X}$	$RV_{q=1}$	$RV_{q=1,\alpha=0.05}$	$R^2_{Y\ D\|X}$	$RV_{q=1}$	$RV_{q=1,\alpha=0.05}$
Dam Height	16.5, 30.5, 1.8, 1	35.6, 47.8, 12.5, 9.7	18.6, 31.1, 0, 0	28.2, 4.1, 12.6, 4.4	46, 18.6, 31.4, 19.3	32, 0, 10.1, 0
Crest Width	0, 1.5, 0, 1.1	2, 11.5, 0.4, 10.1	0, 0, 0, 0	0.2, 4.5, 2.7, 0.2	4, 19.5, 15.4, 4	0, 0, 0, 0
Breach Diameter	-, -, 0.2, 0.3	-, -, 4.6, 5	-, -, 0, 0	-, -, 20.3, 44.3	-, -, 39.3, 57.8	-, -, 20.6, 46
Breach Height	1.1, 42.3, -, 8.4	10.1, 56.5, -, 26.1	0, 42.9, -, 3.8	14.2, 49.2, -, 3.8	33.2, 61.3, -, 18.1	15.5, 49.5, -, 0
Reservoir Contents	89.9, 78.4, 87.1, 95.5	90.8, 81.6, 88.4, 95.7	89.4, 77.2, 86.2, 95.2	42.3, 67.8, 10.1, 37.3	56.5, 74, 28.4, 52.9	45.6, 66.9, 6.1, 39.4
Erodibility	97.5, 95.5, 96.2, 98.2	97.5, 95.7, 96.4, 98.3	97.3, 95.1, 96, 98.1	90.9, 90.4, 88.2, 69.7	91.6, 91.1, 89.3, 75.3	90.5, 89.6, 87.4, 69.3
Shear Stress	0.1, 1.6, 3.2, 3.7	3.6, 11.9, 16.6, 17.8	0, 0, 0, 0	0.2, 1, 0.2, 1.2	4.2, 9.5, 4.3, 10.3	0, 0, 0, 0
Manning's N	95.9, 3.3, 95.3, 1.2	96, 17, 95.5, 10.5	95.7, 0, 95, 0	38.3, 0.7, 76, 0	53.6, 7.9, 79.8, 0.8	41.9, 0, 75.1, 0
Cohesion	1.7, -, 35, -	12.3, -, 51.3, -	0, -, 36.7, -	5.2, -, 10.9, -	20.8, -, 29.3, -	0, -, 7.3, -
Particle Diameter	0, -, 0, -	2.1, -, 1, -	0, -, 0, -	0, -, 3.5, -	1.2, -, 17.3, -	0, -, 0, -
Porosity	1.4, -, 2.5, -	11.1, -, 14.9, -	0, -, 0, -	0.1, -, 1.2, -	2.2, -, 10.5, -	0, -, 0, -
Friction Angle	0.2, -, 0, -	4.5, -, 1.8, -	0, -, 0, -	0.1, -, 2.7, -	3, -, 15.4, -	0, -, 0, -
Unit Weight	-, 15.7, -, 0.5	-, 34.9, -, 6.7	-, 13.5, -, 0	-, 0, -, 0.4	-, 0.7, -, 6.2	-, 0, -, 0
Shear Strength	-, 0, -, -	-, 0.9, -, -	-, 0, -, -	-, 2.2, -, -	-, 14, -, -	-, 0, -, -
Clay Content	7.2, -, -, -	24.2, -, -, -	3.9, -, -, -	0, -, -, -	1.8, -, -, -	0, -, -, -
	Max Width			Max Depth		
	$R^2_{Y\ D\|X}$	$RV_{q=1}$	$RV_{q=1,\alpha=0.05}$	$R^2_{Y\ D\|X}$	$RV_{q=1}$	$RV_{q=1,\alpha=0.05}$
Dam Height	0.3, 70.8, 0, 57	5.2, 76.1, 1, 66.6	0, 69.8, 0, 57.6	-, 100, -, 99.9	-, 100, -, 99.9	-, 100, -, 99.9
Crest Width	0, 27, 0.8, 66.4	0.2, 45.1, 8.4, 73	0, 27.4, 0, 66.2	-, 1.2, -, 0	-, 10.6, -, 0.9	-, 0, -, 0
Breach Diameter	-, -, 0.1, 2.4	-, -, 3.5, 14.6	-, -, 0, 0	-, -, -, 2.4	-, -, -, 14.5	-, -, -, 0
Breach Height	0.1, 0, -, 85.3	3.1, 1.5, -, 87	0, 0, -, 84.5	-, 2.7, -, 0.3	-, 15.4, -, 5.7	-, 0, -, 0
Reservoir Contents	29.4, 0.1, 10.9, 20.8	47, 2.3, 29.4, 39.8	33.3, 0, 7.4, 21.9	-, 0.2, -, 0.8	-, 4.6, -, 8.6	-, 0, -, 0
Erodibility	97.8, 4.1, 96.3, 89	97.8, 18.6, 96.4, 90	97.6, 0, 96, 88.3	-, 2.2, -, 0.1	-, 14, -, 2.7	-, 0, -, 0
Shear Stress	14.3, 2.4, 19.6, 1.9	33.4, 14.6, 38.7, 12.8	15.6, 0, 19.8, 0	-, 0.5, -, 1.5	-, 7.1, -, 11.4	-, 0, -, 0
Manning's N	97.1, 0.5, 96.5, 1.2	97.2, 6.8, 96.6, 10.4	97, 0, 96.2, 0	-, 0.5, -, 1.8	-, 7, -, 12.8	-, 0, -, 0
Cohesion	24, -, 41.3, -	42.6, -, 55.8, -	27.5, -, 42.8, -	-, -, -, -	-, -, -, -	-, -, -, -
Particle Diameter	0, -, 0, -	1.6, -, 2, -	0, -, 0, -	-, -, -, -	-, -, -, -	-, -, -, -
Porosity	5, -, 2.1, -	20.6, -, 13.6, -	0, -, 0, -	-, -, -, -	-, -, -, -	-, -, -, -
Friction Angle	0.7, -, 1.2, -	8, -, 10.3, -	0, -, 0, -	-, -, -, -	-, -, -, -	-, -, -, -
Unit Weight	-, 0.1, -, 0	-, 3, -, 0.4	-, 0, -, 0	-, 5.3, -, 0	-, 21.1, -, 1.8	-, 0, -, 0
Shear Strength	-, 0, -, -	-, 0.1, -, -	-, 0, -, -	-, 0, -, -	-, 1.6, -, -	-, 0, -, -
Clay Content	2.8, -, -, -	15.7, -, -, -	0, -, -, -	-, -, -, -	-, -, -, -	-, -, -, -

Table A.2: *Full collapse sensitivity results for the four dam/model pairs. The order of values in each cell is P1 DLBreach, P1 WinDAM, Big Bay DLBreach, then Big Bay WinDAM C.*

	Width Before			Width After		
	$R^2_{Y\ D\|X}$	$RV_{q=1}$	$RV_{q=1,\alpha=0.05}$	$R^2_{Y\ D\|X}$	$RV_{q=1}$	$RV_{q=1,\alpha=0.05}$
Dam Height	27.9, 7.9, 1.6, 6.8	45.8, 25.3, 12, 23.6	31.7, 0.6, 0, 0.4	1.1, 7.9, 10.3, 7.2	10, 25.3, 28.6, 24.3	0, 0.6, 6.4, 1.3
Crest Width	1.4, 3.4, 4.1, 0.1	11.3, 17, 18.7, 3.5	0, 0, 0, 0	1.3, 3.4, 0.2, 0.1	10.8, 17, 4, 2.7	0, 0, 0, 0
Breach Diameter	-, -, 0.4, 0.3	-, -, 5.9, 5.2	-, -, 0, 0	-, -, 0.1, 0.5	-, -, 2.3, 7	-, -, 0, 0
Breach Height	24.3, 2.4, -, 0	42.9, 14.6, -, 0.3	27.9, 0, -, 0	45.8, 2.8, -, 0	58.9, 15.5, -, 0	48.7, 0, -, 0
Reservoir Contents	29.2, 0.1, 0.3, 1.9	46.9, 3, 5.4, 13	33.1, 0, 0, 0	0, 0, 1.7, 1.6	1.1, 0.7, 12.4, 11.8	0, 0, 0, 0
Erodibility	40.6, 1, 0.7, 16.5	55.3, 9.8, 8, 35.7	44, 0, 0, 16.4	12.4, 3.7, 1.1, 42.1	31.2, 17.8, 10, 56.4	12.9, 0, 0, 44
Shear Stress	5.9, 0.6, 0.4, 0.2	22.2, 7.7, 6, 4.6	1.3, 0, 0, 0	0, 0.6, 1.1, 0.1	0.6, 7.7, 10, 3.3	0, 0, 0, 0
Manning's N	0.6, 1.1, 0.6, 0.1	7.7, 9.9, 7.4, 2.6	0, 0, 0, 0	1.1, 1, 0.2, 0.1	10, 9.5, 3.8, 2.7	0, 0, 0, 0
Cohesion	96.8, -, 99.4, -	96.9, -, 99.4, -	96.6, -, 99.4, -	29.1, -, 99.2, -	46.8, -, 99.2, -	33, -, 99.1, -
Particle Diameter	2.4, -, 3.3, -	14.4, -, 16.9, -	0, -, 0, -	0.1, -, 2.2, -	3.1, -, 13.9, -	0, -, 0, -
Porosity	75.9, -, 45.1, -	79.8, -, 58.4, -	75.7, -, 46.4, -	19.1, -, 47, -	38.2, -, 59.8, -	21.9, -, 48.2, -
Friction Angle	0.5, -, 0.1, -	6.9, -, 2.7, -	0, -, 0, -	22.1, -, 95.5, -	40.9, -, 95.7, -	25.4, -, 95.1, -
Unit Weight	-, 6, -, 0.8	-, 22.2, -, 8.8	-, 0, -, 0	-, 6.1, -, 1.1	-, 22.4, -, 9.8	-, 0, -, 0
Shear Strength	-, 0, -, -	-, 0.5, -, -	-, 0, -, -	-, 0, -, -	-, 0.2, -, -	-, 0, -, -
Clay Content	0, -, -, -	0.5, -, -, -	0, -, -, -	2.1, -, -, -	13.6, -, -, -	0, -, -, -
	Depth Before			Depth After		
	$R^2_{Y\ D\|X}$	$RV_{q=1}$	$RV_{q=1,\alpha=0.05}$	$R^2_{Y\ D\|X}$	$RV_{q=1}$	$RV_{q=1,\alpha=0.05}$
Dam Height	-, 0.9, -, 6.8	-, 9.1, -, 23.6	-, 0, -, 0.4	-, 100, -, 99.9	-, 100, -, 99.9	-, 100, -, 99.9
Crest Width	-, 2.1, -, 0.1	-, 13.6, -, 3.5	-, 0, -, 0	-, 1.2, -, 0.3	-, 10.6, -, 5.4	-, 0, -, 0
Breach Diameter	-, -, -, 0.3	-, -, -, 5.6	-, -, -, 0	-, -, -, 1.6	-, -, -, 11.9	-, -, -, 0
Breach Height	-, 10.5, -, 1.5	-, 28.9, -, 11.7	-, 5.4, -, 0	-, 2.7, -, 0.2	-, 15.4, -, 4.8	-, 0, -, 0
Reservoir Contents	-, 9, -, 1.9	-, 26.8, -, 13	-, 2.6, -, 0	-, 0.2, -, 0.7	-, 4.6, -, 8.1	-, 0, -, 0
Erodibility	-, 2, -, 16.5	-, 13.2, -, 35.7	-, 0, -, 16.4	-, 2.2, -, 0.2	-, 14, -, 4.3	-, 0, -, 0
Shear Stress	-, 0, -, 0.2	-, 2.1, -, 4.6	-, 0, -, 0	-, 0.5, -, 0.6	-, 7.1, -, 7.7	-, 0, -, 0
Manning's N	-, 0.3, -, 0.1	-, 5.4, -, 2.6	-, 0, -, 0	-, 0.5, -, 2.7	-, 7, -, 15.3	-, 0, -, 0
Cohesion	-, -, -, -	-, -, -, -	-, -, -, -	-, -, -, -	-, -, -, -	-, -, -, -
Particle Diameter	-, -, -, -	-, -, -, -	-, -, -, -	-, -, -, -	-, -, -, -	-, -, -, -
Porosity	-, -, -, -	-, -, -, -	-, -, -, -	-, -, -, -	-, -, -, -	-, -, -, -
Friction Angle	-, -, -, -	-, -, -, -	-, -, -, -	-, -, -, -	-, -, -, -	-, -, -, -
Unit Weight	-, 0.2, -, 0.8	-, 4.4, -, 8.7	-, 0, -, 0	-, 5.3, -, 0.2	-, 21.1, -, 4.4	-, 0, -, 0
Shear Strength	-, 0.3, -, -	-, 5.2, -, -	-, 0, -, -	-, 0, -, -	-, 1.6, -, -	-, 0, -, -
Clay Content	-, -, -, -	-, -, -, -	-, -, -, -	-, -, -, -	-, -, -, -	-, -, -, -
	Time to Collapse					
	$R^2_{Y\ D\|X}$	$RV_{q=1}$	$RV_{q=1,\alpha=0.05}$			
Dam Height	24.5, 12.7, 39.3, 6.1	43, 31.5, 54.3, 22.5	28.1, 9, 40.8, 0			
Crest Width	0.1, 3.1, 2.8, 2	3.2, 16.4, 15.5, 13.3	0, 0, 0, 0			
Breach Diameter	-, -, 50.4, 30	-, -, 62.1, 47.5	-, -, 51.3, 32.1			
Breach Height	12.3, 27.5, -, 1.4	31, 45.4, -, 11.3	12.7, 27.9, -, 0			
Reservoir Contents	33.2, 22, 0.8, 0.5	49.9, 40.8, 8.8, 7.1	37, 21.6, 0, 0			
Erodibility	88.2, 60.6, 73.6, 64	89.3, 69, 78.1, 71.4	87.6, 60.2, 72.8, 64.1			
Shear Stress	0.3, 0.3, 1.2, 2.8	4.9, 5.1, 10.3, 15.7	0, 0, 0, 0			
Manning's N	3, 0.8, 2.1, 0	16, 8.8, 13.6, 1.2	0, 0, 0, 0			
Cohesion	25, -, 11.6, -	43.5, -, 30.2, -	28.7, -, 8.5, -			
Particle Diameter	0.1, -, 5.1, -	2.6, -, 20.7, -	0, -, 0, -			
Porosity	0.3, -, 3, -	5, -, 16.1, -	0, -, 0, -			
Friction Angle	0, -, 0.2, -	0.2, -, 4.3, -	0, -, 0, -			
Unit Weight	-, 8.1, -, 0.7	-, 25.6, -, 8.1	-, 1, -, 0			
Shear Strength	-, 0, -, -	-, 1, -, -	-, 0, -, -			
Clay Content	0.1, -, -, -	2.5, -, -, -	0, -, -, -			

Table A.3: *p-values for the HERU P1 dam in DLBreach.*

	Peak Flow	Peak Time	Breach Width	Collapse Time	Width BC	Width AC
Dam Height	1.56E-05	3.39E-04	7.70E-02	9.31E-03	2.50E-07	3.86E-02
Crest Width	9.47E-02	5.95E-01	5.36E-01	8.22E-01	3.36E-01	3.19E-01
Erodibility	1.95E-56	6.65E-38	1.09E-58	1.74E-34	4.86E-11	2.24E-02
Critical Shear Stress	8.14E-01	5.77E-01	5.84E-02	3.90E-01	3.85E-02	5.30E-01
Manning's N	6.03E-51	5.35E-09	1.94E-56	1.45E-01	5.00E-01	3.79E-01
Surface Elevation	1.69E-36	4.38E-10	1.19E-05	5.69E-08	3.97E-07	8.83E-01
Breach Height	4.91E-06	1.08E-04	2.40E-06	3.52E-03	1.10E-05	2.98E-11
Cohesion	1.14E-13	3.92E-02	8.32E-08	6.98E-02	8.50E-54	4.21E-04
Porosity	5.62E-01	4.76E-01	4.65E-01	9.04E-01	5.34E-23	6.29E-04
Particle Diameter	1.12E-03	4.26E-01	7.30E-02	5.43E-01	2.77E-02	9.03E-01
Friction Angle	9.69E-02	3.21E-01	2.11E-01	4.45E-01	7.99E-01	2.19E-04
Clay Content	6.69E-02	3.31E-02	1.20E-01	5.34E-03	1.88E-06	4.50E-01

Table A.4: *p-values for the HERU P1 dam in WinDAM C.*

	Peak Flow	Peak Time	Breach Width	Breach Depth	Collapse Time	Width BC	Width AC	Depth BC	Depth AC
Dam Height	2.07E-01	2.30E-03	4.30E-15	1.79E-94	2.58E-03	5.87E-02	9.30E-02	4.92E-01	1.79E-94
Crest Width	1.32E-01	1.19E-01	4.14E-04	9.18E-25	2.26E-01	1.91E-01	1.81E-01	2.71E-01	9.18E-25
Erodibility	5.01E-34	1.76E-27	1.22E-01	2.03E-01	2.64E-12	4.54E-01	1.54E-01	3.04E-01	2.03E-01
Critical Shear Stress	7.17E-01	1.30E-02	2.73E-01	7.52E-01	1.67E-01	7.22E-01	7.29E-01	6.68E-01	7.52E-01
Manning's N	1.94E-01	5.62E-01	6.22E-01	6.09E-01	5.20E-01	4.62E-01	4.84E-01	6.96E-01	6.09E-01
Surface Elevation	2.34E-18	1.59E-14	7.73E-01	6.41E-01	5.30E-04	9.67E-01	9.06E-01	2.11E-02	6.41E-01
Breach Height	3.81E-07	5.83E-09	9.71E-01	2.10E-01	3.86E-05	2.22E-01	1.95E-01	2.01E-02	2.10E-01
Unit Weight	8.13E-01	3.15E-02	7.57E-01	1.02E-01	2.44E-03	4.90E-02	5.46E-02	9.39E-01	1.02E-01
Shear Strength	1.35E-12	1.73E-04	7.15E-01	1.40E-14	5.06E-01	8.71E-01	6.60E-01	6.61E-01	1.40E-14

Table A.5: *p-values for the Big Bay dam in DLBreach.*

	Peak Flow Rate	Peak Time	Breach Width	Collapse Time	Width BC	Width AC
Dam Height	2.43E-06	9.53E-01	1.26E-05	2.17E-04	5.16E-01	6.77E-03
Crest Width	1.58E-11	2.15E-03	7.41E-03	8.17E-02	1.38E-01	9.98E-01
Erodibility	3.62E-39	4.93E-26	3.39E-38	4.68E-18	1.31E-01	8.11E-01
Critical Shear Stress	1.67E-02	8.94E-01	6.73E-04	8.17E-02	6.17E-01	6.48E-05
Manning's N	2.99E-38	1.14E-18	1.46E-41	2.83E-01	5.70E-01	7.73E-01
Reservoir Volume	6.87E-30	9.73E-04	1.55E-01	8.19E-01	7.60E-01	3.04E-01
Breach Diameter	2.30E-01	1.25E-03	2.14E-01	5.54E-10	6.42E-01	8.57E-01
Cohesion	9.78E-01	9.82E-01	1.18E-02	2.59E-01	2.96E-64	5.04E-60
Porosity	7.74E-01	6.68E-01	2.84E-01	2.39E-02	1.07E-08	5.20E-02
Particle Diameter	1.51E-05	2.05E-01	3.25E-08	4.81E-01	3.70E-01	9.01E-04
Friction Angle	3.58E-03	7.03E-01	6.63E-01	9.65E-01	8.92E-01	1.04E-38

Table A.6: *p-values for the Big Bay dam in WinDAM C.*

	Peak Flow	Peak Time	Breach Width	Breach Depth	Collapse Time	Width BC	Width AC	Depth BC	Depth AC
Dam Height	3.27E-01	1.92E-01	6.31E-12	1.26E-93	4.47E-02	4.28E-02	3.73E-02	4.31E-02	6.15E-94
Crest Width	2.98E-03	7.59E-01	4.51E-13	2.25E-01	3.68E-01	8.29E-01	9.01E-01	7.83E-01	3.81E-01
Erodibility	1.22E-52	9.33E-17	1.35E-29	8.66E-01	1.34E-14	1.32E-03	2.09E-08	1.32E-03	7.62E-01
Critical Shear Stress	1.40E-01	4.11E-01	2.99E-01	3.59E-01	1.99E-01	7.19E-01	8.00E-01	7.20E-01	5.46E-01
Manning's N	1.87E-01	7.98E-01	6.27E-01	5.20E-14	4.94E-01	9.64E-01	7.71E-01	8.96E-01	7.50E-14
Reservoir Volume	1.20E-50	3.70E-13	1.86E-14	3.96E-01	3.35E-02	4.02E-02	5.81E-03	4.03E-02	4.17E-01
Breach Height	4.67E-14	3.50E-03	5.98E-29	6.59E-01	3.85E-02	6.92E-01	4.41E-01	5.92E-01	7.25E-01
Breach Diameter	6.10E-19	2.93E-05	1.49E-02	2.39E-01	2.26E-02	7.56E-01	3.10E-01	7.07E-01	3.01E-01
Unit Weight	2.28E-02	2.75E-01	2.81E-01	9.24E-01	2.29E-01	5.82E-01	6.28E-01	5.83E-01	7.45E-01

Appendix B

Optimization Parameters

Table B.1: *Parameter values found by Dakota parameter optimization search.*

	P1 DLB Average Error	Perf1	P1 WDC Average error	Perf1	Big Bay DLB Average Error	Perf1	Big Bay WDC Average error	Perf1
Dam height	1.4	1.4	1.22	1.2	15.56	15.61	15.56	15.61
Crest width	1.78	2.18	1.58	1.78	11.6	11.6	11.6	11.6
Breach Diameter					0.01	0.01	0.01	0.01
Breach height	1	1	0.33	0.4			0.6	0.1
Surface elevation	1.003	1.003	0.78	1.003	15577224	15577224	15577224	15577224
Erodibility	121.4	44.27	23.4	23	19.63	19.63		
Shear Stress	0.016	0	0.093	0	2	2	19.63	19.63
Manning's N	0.025	0.025	0.026	0.033	0.03	0.03	2	2
Cohesion	5250	4835			10015.5	5000	0.03	0.03
Particle diameter	0.0001	0.00013			0	0		
Porosity	0.33	0.45			0.35	0.35	0	0
Friction Angle	0.577	0.577			0.68	0.68	0.35	0.35
Unit Weight			19.03	17.26			0.68	0.68
Shear Strength			28354.24	26000				
Clay Content	0.07	0.07						

Table B.2: *Parameter values found by inverse function, surrogate model search, and Dakota.*

		Reverse	Dual-annealing	Gradient Decent
DLBreach	Dam Height	5.56	15.51579	15.61
	Crest Width	12.232	12.172	11.6
	Breach Diameter	0.013	0.049	0.349
	Reservoir Volume	15259503	16051291	0.0003
	Erodibility	12.11	8.102	19.625
	Critical Shear Stress	1.3624	1.464	2
	Manning N	0.025	0.025	0.675
	Cohesion	10707	5066.592	5000
	Particle Diameter	0.0003	0.001	15577224
	Porosity	0.276	0.297	0.013
	Friction Angle	0.677	0.598	0.025
WinDAM	Dam Height	15.6	15.502	15.5
	Crest Width	11.24	11.785	11.6
	Breach Diameter	0.031	0.011	0.01
	Breach Height	0.35	0.046	0.1
	Reservoir Volume	14106049	13979082	13880000
	Erodibility	34.93	48.914	41.125
	Critical Shear Stress	2.04	4.23	3.125
	Manning N	0.027	0.025	0.025
	Unit Weight	18.49	18.38039	19

Table B.3: *Invalid parameters found for P1 WinDAM by the inverse function method.*

Dam Height	1.200011
Crest Width	1.750408
Breach Height	0.374768
Reservoir Contents	0.860799
Erodibility	23.02752
Shear Stress	0.093165
Manning's N	0.026391
Unit Weight	19.0251
Shear Strength	28354.24