

PARAMETER STUDY FOR WINDAM USING DAKOTA

by

ASHWIN RAMACHANDRA BHAT

B.E., Visvesvaraya Technological University, 2009

A REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences  
College of Engineering

KANSAS STATE UNIVERSITY  
Manhattan, Kansas

2014

Approved by:

Major Professor  
Dr. Mitchell L. Neilsen

# **Copyright**

ASHWIN RAMACHANDRA BHAT

2014

## **Abstract**

Windows™ Dam Analysis Modules (WinDAM) is a set of modular software components that can be used to analyze overtopped earthen embankments and internal erosion of embankment dams. Sandia National Laboratories' DAKOTA (Design Analysis Kit for Optimization and Terascale Applications) toolkit provides algorithms to perform iterative analysis with the help of built-in algorithms for uncertainty quantification with sampling and parameter study methods. This software integrates Sandia National Laboratories' DAKOTA software suite with WinDAM. It provides a user-interface for input and manipulation of parameters and performs various (centered and multi-dimensional) parameter studies on a wide range of parameters. This software takes advantage of the various algorithms present in DAKOTA to perform parameter studies on the various properties of WinDAM and provides users with detailed output on the changes caused by these variations.

# Table of Contents

List of Figures .....	v
Acknowledgements .....	vi
Chapter 1 - Introduction .....	1
Chapter 2 - Background .....	2
WinDAM .....	2
DAKOTA .....	4
Centered Parameter Study .....	5
Multi-Dimensional Parameter Study .....	7
Chapter 3 - Software for Parameter Study .....	8
Chapter 4 - Conclusion .....	21
References .....	22

## List of Figures

Figure 1: Iterative Analysis.....	1
Figure 2: WinDAM Software Architecture.....	3
Figure 3: Internal Erosion of USDA NRCS Structure.....	4
Figure 4: Centered Parameter Study.....	6
Figure 5: Multi-Dimensional Parameter Study.....	7
Figure 6: Random Samples Generated.....	8
Figure 7: Software Data Flow.....	10
Figure 8: Software Control Flow.....	11
Figure 9: WDC Input File.....	12
Figure 10: WDT Template File.....	12
Figure 11: Centered Parameter Study Panel.....	13
Figure 12: Multi-dimensional Parameter Study Panel.....	14
Figure 13: Dakota Input File.....	15
Figure 14: WinDam.bat Script File.....	15
Figure 15: Processed WDC File.....	16
Figure 16: Example OUT File.....	17
Figure 17: Dakota_tabular.dat File.....	17
Figure 18: DAKOTA Input File.....	19
Figure 19: WDC File with Parameters.....	20

## **Acknowledgements**

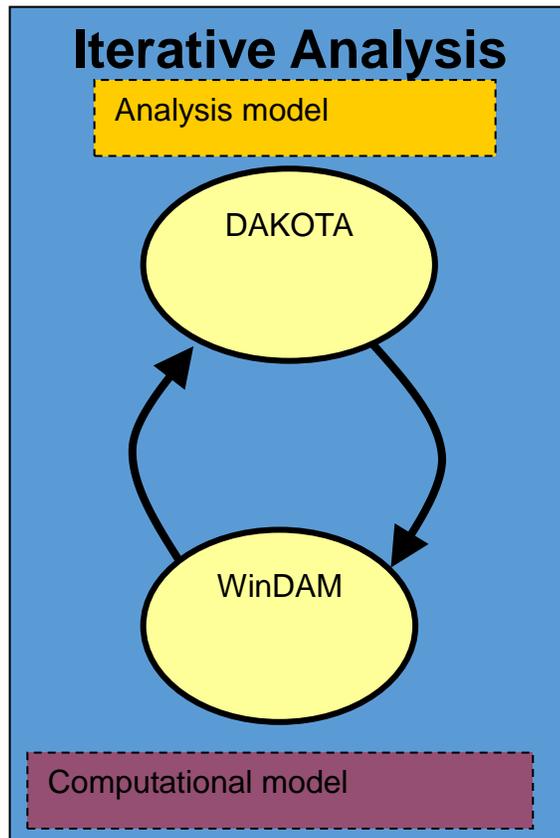
I take this opportunity to express my profound gratitude and deep regards to my major professor Dr. Mitchell L. Neilsen for his exemplary guidance, monitoring and constant encouragement throughout the course of this report. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I also take this opportunity to express a deep sense of gratitude to Dr. Daniel Anderson, Dr. Torben Amtoft for their cordial support, valuable information and guidance.

Lastly, I thank almighty, my parents, brother and friends for their constant encouragement without which this assignment would not be possible.

## Chapter 1 - Introduction

This software integrates Sandia National Laboratories' DAKOTA software suite with WinDAM. It provides a user-interface for input and manipulation of parameters and performs various (centered and multi-dimensional) analyses on a wide range of parameters. This software can be used to automatically generate DAKOTA input files. For parameter studies, the user indirectly specifies these components through strategy, method, variables, interface, and response keywords. Then, DAKOTA is invoked to iterate on the WinDAM simulation models. The integrated development system for water resource site analysis WinDAM+DAKOTA is designed to fully integrate the simulation models in WinDAM. The current version of WinDAM only uses the Latin Hyper-Cube Sampling and Monte-Carlo Sampling libraries in DAKOTA to generate random samples. Thus, the request comes from WinDAM, and random samples are supplied by DAKOTA. The goal of this project is to allow DAKOTA to drive the simulation.



**Figure 1: Iterative Analysis**

In the proposed model based on [1], DAKOTA is given more capability in order to generate data within the constraints and execute it against on the WinDAM for detailed analysis.

## **Chapter 2 - Background**

### **Windows™ Dam Analysis Modules (WinDAM)**

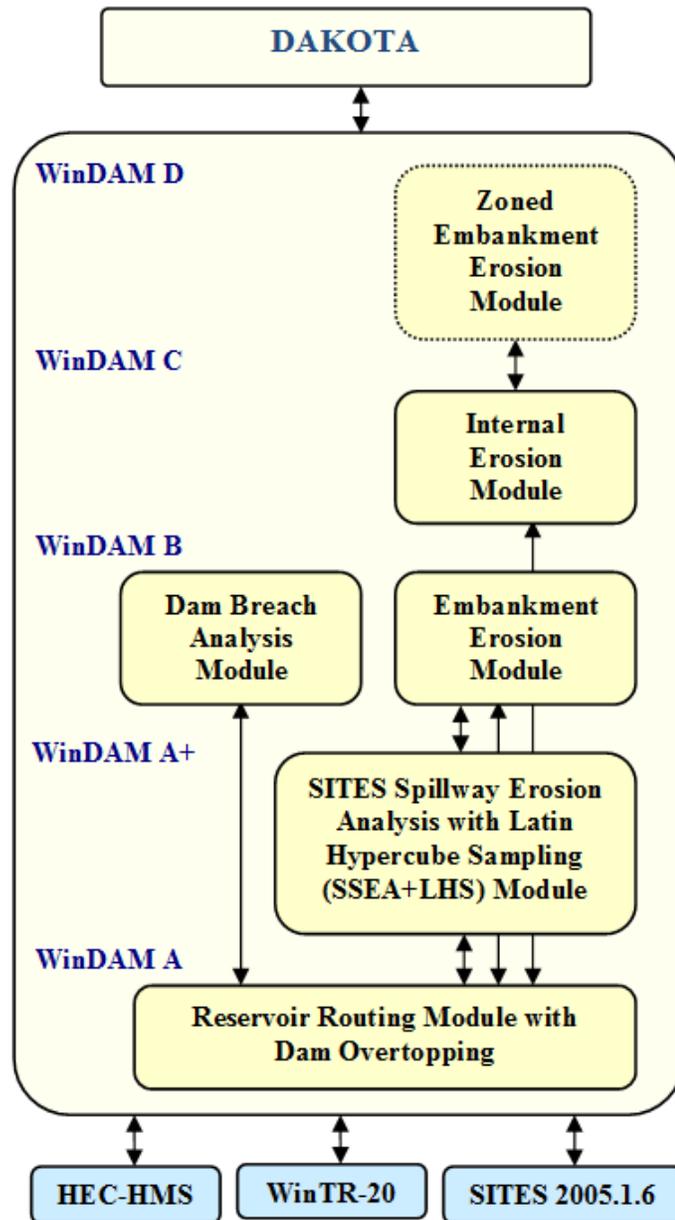
Windows™ Dam Analysis Modules (WinDAM) is a set of modular software components that can be used to analyze overtopped earthen embankments and internal dam erosion. The primary purpose of WinDAM B is threefold:

- Hydraulically route one input hydrograph through, around, and over a single earthen dam.
- Estimate auxiliary spillway erosion in up to three earthen or vegetated auxiliary spillways.
- Estimate erosion of the earthen embankment caused by overtopping of the dam embankment.

Since WinDAM B does not include any specific hydrology component, the user must create the input hydrograph using other software. This allows the user the flexibility to choose the hydrologic software most suitable for analysis of site conditions; e.g., HEC-HMS, etc. WinDAM B assumes the embankment of the dam is a homogenous earthen material.

Computation of the discharge through the area of the breach, if any, is unit discharge based on the effective width. If breach is to be evaluated, the associated erosion is assumed to be initiated in an area corresponding to maximum unit discharge over the top of the dam.

Following breach initiation, the unit discharge is computed assuming negligible energy loss from the reservoir to the hydraulic control and critical flow conditions with hydrostatic pressure at the hydraulic control. The processes that determine the erosion during embankment breach are dependent on the breach geometry and the breach area discharge. Many different software packages can be used with WinDAM to generate the input hydrographs that are used by WinDAM to determine the amount of erosion that results. As shown in Figure 2, these packages include WinTR-20, Sites, HEC-RAS, etc.



**Figure 2. WinDAM Software Architecture**

In WinDAM, users may select an energy-based or a stress-based model. The energy-based model, designated the Temple/Hanson model, is described by Temple et al. [3]. The stress-based model, designated the Hanson/Robinson model, is an adaptation of the model described by Hanson, et al. [4], These advance rate models reflect different degrees of simplification of the complex process and have different input requirements. These models are used to predict the

amount of erosion that results from overtopping or internal erosion. An example of internal erosion is shown in Figure 3.



**Figure 3. Internal Erosion of USDA NRCS Structure**

Although WinDAM currently relies on DAKOTA for random samples used in the simulations, those are the only features of DAKOTA that are used.

### **Design Analysis Kit for Optimization and Terascale Applications (DAKOTA):**

**DAKOTA** is a software toolkit developed by engineers at Sandia National Laboratories to provide a flexible, extensible interface between analysis codes and iterative systems analysis methods [2]. DAKOTA contains optimization algorithms to help with parameter study capabilities. It provides an ideal platform for providing iterative analysis on the inputs selected by a user. DAKOTA currently supports four types of parameter studies:

1. **Vector parameter studies** compute response data sets at selected intervals along an n-dimensional vector in parameter space. This capability encompasses both single-coordinate parameter studies (to study the effect of a single variable on a response set) as well as multiple coordinate vector studies (to investigate the response variations along some arbitrary vector).

2. **List parameter studies** compute response data sets at a list of points in parameter space, defined by the user. These points are explicitly specified by the user and are not confined to lie on any line or surface. Thus, this parameter study provides a general facility that supports the case where the desired set of points to evaluate does not fit the prescribed structure of the vector, centered, or multidimensional parameter studies.

3. A **centered parameter study** computes multiple coordinate-based parameter studies, one per parameter, centered about the initial parameter values. The centered parameter study executes multiple coordinate-based parameter studies, one per parameter, centered about the specified Initial Values. This is useful for investigation of function contours in the vicinity of a specific point.

4. A **multi-dimensional parameter study** computes response data sets for an n-dimensional hypergrid of points. Each variable is partitioned into equally spaced intervals between its upper and lower bounds, and each combination of the values defined by these partitions is evaluated.

For the project, we have used the Centered Parameter Study and Multi-Dimensional Study for the analysis.

### ***2.1. Centered Parameter Study:***

Centered Parameter Study helps in investigating of a function in the vicinity of a specific point. Thus, after computing an optimal design, Centered Parameter Study could be used for post-optimality analysis in verifying that the computed solution is actually at a minimum or constraint boundary and in investigating the shape of this minimum or constraint boundary.

In order to do Centered Parameter Study method, Dakota input file would require to have `step_vector` (list of reals) and `steps_per_variable` (list of integers) specifications, where the `step_vector` specifies the size of the increments per variable (employed sequentially, not all at once) and the `steps_per_variable` specifies the number of increments per variable (employed sequentially, not all at once) for each of the positive and negative step directions.

For example, with Initial Values of (1.0, 1.0), a step vector of (0.1, 0.1), and a steps per variable of (2, 2), the center point is evaluated followed by four function evaluations (two negative deltas and two positive deltas) per variable:

**Parameters for function evaluation 1:**

1.000000000e+00 d1

1.000000000e+00 d2

**Parameters for function evaluation 2:**

8.000000000e-01 d1

1.000000000e+00 d2

**Parameters for function evaluation 3:**

9.000000000e-01 d1

1.000000000e+00 d2

**Parameters for function evaluation 4:**

1.100000000e+00 d1

1.000000000e+00 d2

**Parameters for function evaluation 5:**

1.200000000e+00 d1

1.000000000e+00 d2

**Parameters for function evaluation 6:**

1.000000000e+00 d1

8.000000000e-01 d2

**Parameters for function evaluation 7:**

1.000000000e+00 d1

9.000000000e-01 d2

**Parameters for function evaluation 8:**

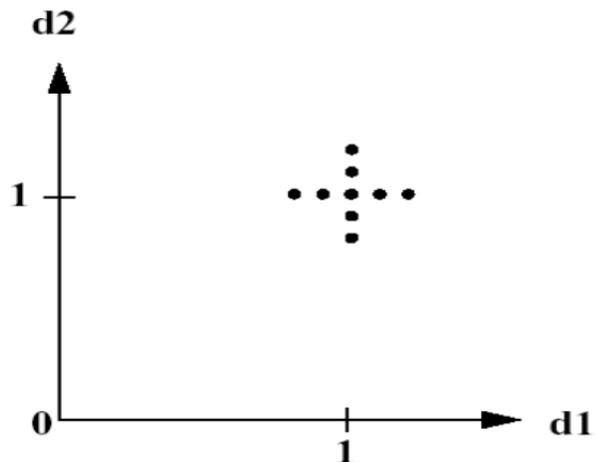
1.000000000e+00 d1

1.100000000e+00 d2

**Parameters for function evaluation 9:**

1.000000000e+00 d1

1.200000000e+00 d2



**Figure 4: Centered Parameter Study**

## 2.2. Multi-dimensional Parameter Study:

The Multi-Dimensional Parameter Study computes response data sets for an n-dimensional hypergrid of points. Each variable is partitioned into equally spaced intervals between its upper and lower bounds, and each combination of values defined by these partitions is evaluated. For example, let's consider a two variable example problem with  $d1 \in [0,2]$  and  $d2 \in [0,3]$  (as defined by the lower and upper bounds from the variables specification) and with partitions of (2, 3), Then the interval  $[0, 2]$  is divided into 2 equal partitions and  $[0, 3]$  is divided into 3 equal partitions which can be represented by the following diagram:

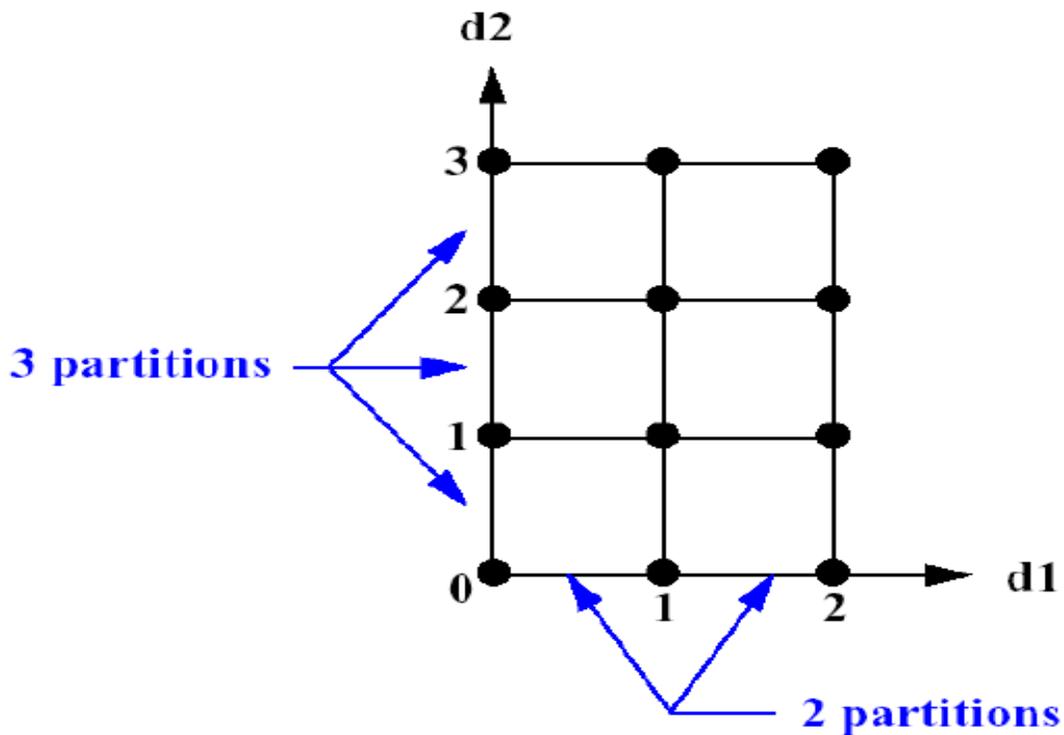


Figure 5: Multi-Dimensional Parameter Study

The above diagram can be described by simply using all points in all dimensions. Such output is useful to generate multi-dimensional plots of the data for visualization during post processing.

## Chapter 3 - Software for Parameter Study

In the previous software, the integrated development system for water resource site analysis WinDAM+DAKOTA is designed to fully integrate the simulation models in WinDAM with the uncertainty quantification, sensitivity analysis, and parameter studies capabilities in DAKOTA. This environment interactively guides user input, invokes the sampling and simulation models in the background, and parses the results. This version of the software also allows a user to conduct parameter studies and specify the inputs to be analyzed based on a list of probability distribution functions. Twenty-five different types of distributions can be specified [5]. Random samples are generated using the Latin Hypercube Sampling (LHS) library routines found in Dakota 5.2. In addition to specifying the types of distributions to be used to generate samples, the user can also specify the number of instances to be generated and the algorithm to be used to generate those samples. In particular, the user can select between Monte Carlo and Latin Hypercube Sampling. With Monte Carlo Sampling, the samples are generated at random. The user can specify a random number seed to generate the same sequence of random samples. With Latin Hypercube Sampling, the samples are more evenly distributed across the search space, resulting in better coverage and fewer samples required [6]. Given below is an instance where a user could request 10 instances (samples) to be generated for a given material's headcut index using a Uniform Distribution from 0.001 to 0.201. Then, one sample would be randomly generated for each interval of length 0.02 from 0.001 to 0.201. The generated samples are shown in Figure 5.

```
@UNCERTAINTY
  @OBSERVATIONS      10
  @VARIABLES         1
    KH(1) :
@SAMPLEDATA
1 1  0.178143860112386
2 1  0.114162037013804
3 1  0.153554977141378
4 1  0.184678807170631      <- min erosion
5 1  0.140484162236030
6 1  7.696084019646307E-003 <- max erosion
7 1  6.790786373412117E-002 <- mean erosion
8 1  2.112528697716014E-002
9 1  5.711419140612775E-002
10 1 9.154328306156087E-002
```

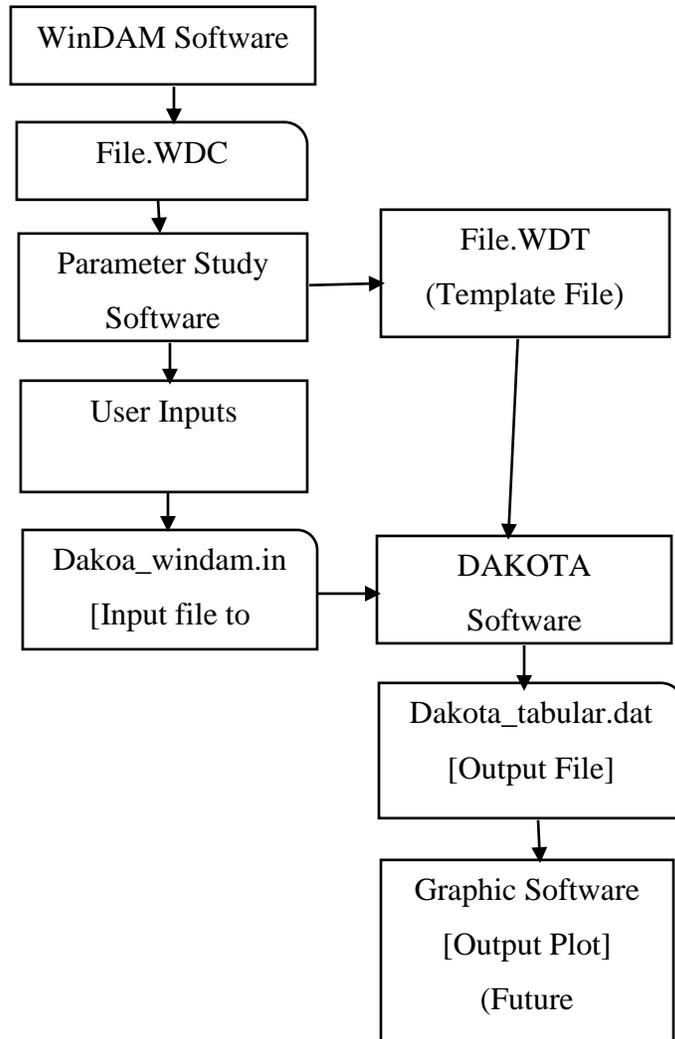
Figure 6. Random Samples Generated

As in the previous software WinDAM used to drive the analysis, with the current software we can allow DAKOTA to be used to drive the analysis in an iterative fashion with the capability of Centered Parameter Study and Multi-Dimensional Parameter Study on the inputs.

With the software, we can analyze the variation of the output by fluctuating the basic parameters and material properties in WinDAM. The variation of the properties can be generated with the help of DAKOTA. The user provides the desired inputs within which variation of a property is to be analyzed and the randomized sets of data will be generated for the purpose through DAKOTA. The software has 2 variations that is the Centered Parameter Study where it takes the initial points, the steps of variation and the number of steps for each variable through the tool and generate a DAKOTA input file and the required WinDAM Template file. If Multi-Dimensional Parameter Study is selected, the user will have input an expected lower bound and upper bound value with the number of partitions for each variable through the tool and the corresponding DAKOTA input file and WinDAM Template file are generated. DAKOTA input file is run on DAKOTA and its outputs are recorded. These outputs are generated by using centered parameter study or multidimensional parameter study are used to generate multiple input files for WinDAM. After the software builds, DAKOTA input file will generate all the possible combinations of input values which are parsed to generate WinDAM input file which are sent to WinDAM for execution. The output files contain all of the execution results from WinDAM where it compute various required output parameters. The software parses output files for relevant data such as peak outflow, maximum unit discharge. These parsed values are collected and the DAKOTA chooses the next set of parameters to continue the process. The parsed outputs are saved in a file for each run and can be sent to the visualization software to display the variation using a scatter plot. After having all the possible variations output, these values can provide us with a fair idea on all of the possible combination of parameters that could be selected. The plotted graph helps in identifying the maximum disturbance achieved. If the user would like to test another possible variation, the software will again follow the same process mentioned and generate output.

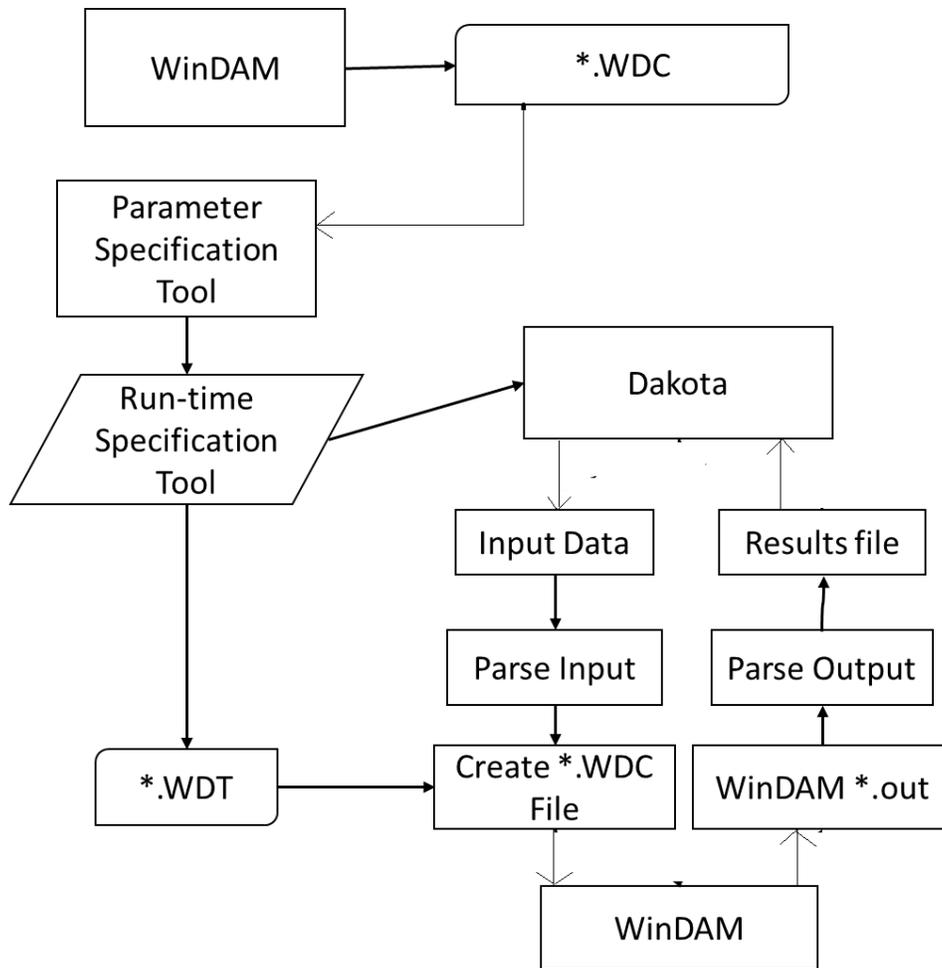
The Dakota template file is used for filling all the values and generating the Dakota control file `Dakota_windam.in`.

In the whole model, Dakota has control over the generation of the WinDAM files and parses the output for required results. Figure 7 describes the data flow of the software.



**Figure 7: Software Data Flow**

Figure 8 shows the control flow of the software. As a separate process, users can execute and generate WinDAM control files \*.WDC files.



**Figure 8: Software Control Flow**

The WinDAM control files (\*.WDC files) are used as input to the software. The software parses the WinDAM control file to generate a WinDAM template file (\*.WDT file) with tags replacing the data fields to be used for the parameter study. Then, random parameters are generated by Dakota and the random samples are inserted into the template file to create an input file that can be passed to the WinDAM simulator to generate the predicted erosion for the given input parameters. Finally, the output generated by the WinDAM simulator is parsed to create a results file that is passed back into Dakota. In this way, Dakota can iterate several times over the parameters to perform parameters studies and uncertainty analysis. For example, suppose that we are interested in the highlighted line shown below. It describes the dam erosion model properties.

1	WINDAM	01/01/2009	Internal Erosion					
2	OPTION	SIMPLE	EARTH			INTERNAL		
3	*	K Visser, FTW						
4	*	March 6, 2013						
5	*	Normal Internal Erosion Run						
6	IEMODEL	2	100	.05	100	0		
7	HYD	Triangular	0.05	0	0C			
8			0	100	200	300	400	
9			500	600	700	800	900	
10			1000	1100	1200	1300	1400	
11			1500	1600	1700	1800	1900	
12			2000	2100	2200	2300	2400	
13			2500	2600	2700	2800	2900	
14			3000	3100	3200	3300	3400	
15			3500	3600	3700	3800	3900	
16			4000	4100	4200	4300	4400	
17			4500	4600	4700	4800	4900	
18			5000	5100	5200	5300	5400	
19			5500	5600	5700	5800	5900	
20			6000	5950	5900	5850	5800	

**Figure 9: WDC Input File**

After the WDC file is provided as input with the type of parameter study to be done, a template file with the extension .WDT is automatically generated by the new software:

1	WINDAM	01/01/2009	Internal Erosion					
2	OPTION	SIMPLE	EARTH			INTERNAL		
3	*	K Visser, FTW						
4	*	March 6, 2013						
5	*	Normal Internal Erosion Run						
6	IEMODEL	2	{TW}	{KD}	{US}	{CS}		
7	HYD	Triangular	0.05	0	0C			
8			0	100	200	300	400	
9			500	600	700	800	900	
10			1000	1100	1200	1300	1400	
11			1500	1600	1700	1800	1900	
12			2000	2100	2200	2300	2400	
13			2500	2600	2700	2800	2900	
14			3000	3100	3200	3300	3400	
15			3500	3600	3700	3800	3900	
16			4000	4100	4200	4300	4400	
17			4500	4600	4700	4800	4900	
18			5000	5100	5200	5300	5400	
19			5500	5600	5700	5800	5900	
20			6000	5950	5900	5850	5800	
21			5750	5700	5650	5600	5550	
22			5500	5450	5400	5350	5300	

**Figure 10: WDT Template File**

In the above figure, the parameters on which we are doing our parameter study are replaced by “{TW}, {KD}, {US}, and {CS}”. These tags identify the parameters’ location in the file for further processing. The software collects the current values and fills up any one of two panels; i.e., Centered Parameter Study Panel or Multi-dimensional Parameter Study Panel.

Based on the type of parameter study the user will be performing, the user will have to provide different inputs for DAKOTA.

**Centered Parameter Study**

HeadCut Model:

Material Properties:

	Current Value	New Value	Step Vector
Advanced Coef. :	<input type="text"/>	<input type="text"/>	<input type="text"/>
Total Unit Weight:	<input type="text"/>	<input type="text"/>	<input type="text"/>
Erodibility:	<input type="text"/>	<input type="text"/>	<input type="text"/>
Undrained Shear Stress:	<input type="text"/>	<input type="text"/>	<input type="text"/>
Critical Shear Stress:	<input type="text"/>	<input type="text"/>	<input type="text"/>

Number of Steps :

**Figure 11: Centered Parameter Study Panel**

**Multi-Dimensional Parameter Study:**

HeadCut Model:

Material Properties:

	Current Value	Lower Bound	Upper Bound	Partiti...
Advanced Coef. :	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Total Unit Weight :	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Erodibility :	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Undrained Shear Stress :	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Critical Shear Stress :	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

**Figure 12: Multi-dimensional Parameter Study Panel**

After entering parameters for the study, the user clicks on run to generate the DAKOTA input file and run it. If the user selected the centered parameter study then the following file is generated.

```

1 # DAKOTA INPUT FILE - dakota_windam_template.in
2 strategy,
3     single_method
4     .....
5     graphics,tabular_graphics_data
6 method,
7     centered_parameter_study
8 step_vector 10 0.01 5 0.001
9 steps_per_variable 4 4 4 4
10
11 model,
12     single
13
14 variables,
15 continuous_design = 4
16 initial_point 100 0.05 100 0
17 descriptors 'TW' 'KD' 'US' 'CS'
18
19 interface,
20     system
21     .....
22     analysis_driver = 'WinDam.bat'
23     parameters_file = 'params.in'
24     results_file    = 'results.out'
25     file_save
26
27 responses,
28     num_objective_functions = 2
29     no_gradients
30     no_hessians

```

**Figure 13: Dakota Input File**

DAKOTA allows the programmer to specify the interface they like to utilize in performing analysis. The software has its own defined interface system which are defined between line numbers 19 – 24 in Figure 13. The WinDam.bat script contains a set of commands which would be invoked to perform the analysis.

```

1 perl parseWinDamIn params.in newParams.in
2 perl dprepro newParams.in example.WDT example.WDC
3 WinDamSim.exe example.WDC
4 perl parseWinDamOut example.OUT results.out
5

```

**Figure 14: WinDam.bat Script File**

These commands help us to perform the following operations:

1. Read the parameters generated by DAKOTA and parse them into a WinDAM recognizable format.
2. The preprocessor dprepro does the preprocessing where a new example.WDC file is generated from the template file and the parameters generated by Dakota.
3. WinDamSim.exe is the WinDAM executable for simulating the erosion generated when running the example.WDC file.
4. Read the example.OUT file which is the output generated after execution which is parsed to fetch the required results.

The file parseWinDamIn.pl is a perl script which reads the param file and makes them usable for WinDAM. The parameters generated are converted to a 10-column integer format which can be used in WinDAM.

The preprocessor dprepro replaces the tags delimited with “{..}”, with the new parameters generated, in the template file to create a new example.WDC file as shown in Figure 15.

1	WINDAM	01/01/2009	Internal Erosion						
2	OPTION	SIMPLE	EARTH		INTERNAL				
3	*	K Visser, FTW							
4	*	March 6, 2013							
5	*	Normal Internal Erosion Run							
6	IEMODEL	2	0100.0000	0000.0500	0100.0000	0000.0040			
7	HYD	Triangular	0.05	0	0C				
8			0	100	200	300	400		
9			500	600	700	800	900		
10			1000	1100	1200	1300	1400		
11			1500	1600	1700	1800	1900		
12			2000	2100	2200	2300	2400		
13			2500	2600	2700	2800	2900		
14			3000	3100	3200	3300	3400		
15			3500	3600	3700	3800	3900		
16			4000	4100	4200	4300	4400		
17			4500	4600	4700	4800	4900		
18			5000	5100	5200	5300	5400		
19			5500	5600	5700	5800	5900		
20			6000	5950	5900	5850	5800		

**Figure 15: Processed WDC File**

This file is run against WinDAM and the output file example.out gets created. It contains the output generated by the WinDAM simulation. It contains values about the peak outflow possible and other relevant details.

```

76
77 ***** Elevation Summary 1 *****
78 Dam Base Elevation:                100.00 feet
79 Max Tailwater Elevation:           100.00 feet
80 Prin Spwy Crest Elevation:         105.00 feet
81 Start Routing Elevation:           110.00 feet
82 Aux Spwy # 1 Crest Elevation:      none
83 Aux Spwy # 2 Crest Elevation:      none
84 Aux Spwy # 3 Crest Elevation:      none
85 Max Pool Elevation:                121.96 feet
86 Initial Low Point in Dam Crest:    120.00 feet
87 Ending Low Point in Dam Crest:     120.00 feet
88 *****
89
90
91 ***** Flow Summary 2 *****
92 Peak Inflow:                       6000.0000 cfs @      3.00 hours
93 Peak Outflow:                      5957.3396 cfs @      3.05 hours
94 Max Flow Principal Spwy:           103.8871 cfs @      3.05 hours
95 Ending Low Point in Dam Crest:      0.00 feet @
96 Ending Low Point in Dam Crest:      0.00 feet @
97 Ending Low Point in Dam Crest:      0.00 feet @

```

**Figure 16: Example OUT File**

This output file is again parsed by parseWindamout.pl which accumulates all the values we have specified in the file. All these output values, new input params and the execution instance are all collected in a tabular format under Dakota\_tabular.dat.

1	%eval_id	TW	KD	US	CS	obj_fn_1	obj_fn_2
2	1	100	0.05	100	0	0	0
3	2	60	0.05	100	0	0	0
4	3	70	0.05	100	0	0	0
5	4	80	0.05	100	0	0	0
6	5	90	0.05	100	0	0	0
7	6	110	0.05	100	0	0	0
8	7	120	0.05	100	0	0	0
9	8	130	0.05	100	0	0	0
10	9	140	0.05	100	0	0	0
11	10	100	0.01	100	0	0	0
12	11	100	0.02	100	0	0	0
13	12	100	0.03	100	0	0	0
14	13	100	0.04	100	0	0	0
15	14	100	0.06	100	0	0	0
16	15	100	0.07	100	0	0	0
17	16	100	0.08	100	0	0	0
18	17	100	0.09	100	0	0	0
19	18	100	0.05	80	0	0	0
20	19	100	0.05	85	0	0	0
21	20	100	0.05	90	0	0	0
22	21	100	0.05	95	0	0	0
23	22	100	0.05	100	0	0	0

**Figure 17: Dakota\_tabular.dat File**

Dakota\_tabular.dat file is the final output. We will use it with graphical software package and plot the variations. In the above figure, it can be noted that it contains lots of execution instances. It shows that DAKOTA is taking control over the WinDAM and execute multiple instances of example.WDC to check the variation in WinDAM output as shown in Figure 1.

### ***Functional Classes:***

The whole project has been coded into multiple classes and files. Major classes used are FileOperation, InternalFrame.java and MyInternalFrame.java. Each one of these files defines a class that deals with a specific function. InternalFrame.java class helps in running the whole project. InternalFrame.java class provides a user interface where a user can select a file and execute the Dakota code. FileOperation class mostly deals with the reading and writing of files and executing a few commands on the client. MyInternalFrame.java has all the panels for the user interface where a user can enter his/her inputs and can be used for processing.

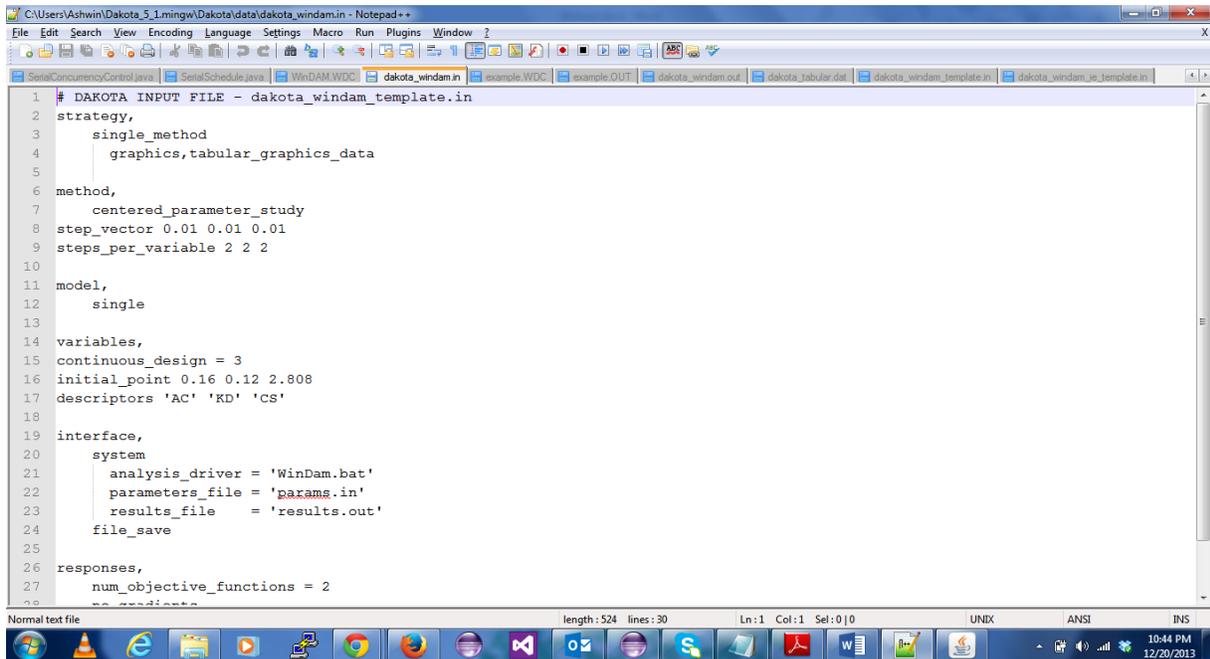
InternalFrame.java class contains the main function which will start the execution of the code. The InternalFrame class instantiates an object of MyInternalFrame class. MyInternalFrame deals with the user interface and a few functions like setValues, writetoDakotaFile. InternalFrame class provides a basic menubar which provides a few basic operations like open a file, save a file and close the frame. It will read a .WDC file selected by the user and fill up the values in the textboxes for extra input and manipulation. If a user changes any value then it is parsed by setValues function and check for changes. WriteDakotafilename function plays a vital role where it creates the appropriate input file for execution in Dakota. Also MyInternalFrame instantiate an object of FileOperations and Output. As a user selects a file as input, the whole file is parsed over and the parameters appear on input frame.

The Fileoperation class deals with all the basic function required on a file and execution. It includes functions like readfile which parses the inputs from the file selected by the user. WriteToFile function deals with the save option provided in the MyInternalFrame action where it can save it in the same file or create a new file and save the modifications. It also contains execution of command build() from InternalFrame which takes the Dakota input file and executes it in its working directory by using the exec function. The output class has 2 vectors which act as a dictionary for saving the values generated during the execution. It has a labels field and output field. The labels field contains all the labels corresponding to the WinDAM

model. The output contains all the different combinations of the values and their respective output.

After we select an input file, the file is parsed and the values are set in the textboxes provided on the user interface. Users provide the step size and the number of steps to be taken in each direction.

When Run + Build is selected, the software takes these inputs and generates a Dakota file and later executes this file to obtain the output. During the execution, first a template file is created. For simpler execution purpose, we have named it example.WDT. It's a template file similar to the .WDC file with the tags like {AC} being replaced at the place of the value. These new values based on the type of parameter study are used in generating the input file that can be processed by DAKOTA. The input file which was created by writeToDakotaFile function looks like:



```
1 # DAKOTA INPUT FILE - dakota_windam_template.in
2 strategy,
3   single_method
4   graphics, tabular_graphics_data
5
6 method,
7   centered_parameter_study
8   step_vector 0.01 0.01 0.01
9   steps_per_variable 2 2 2
10
11 model,
12   single
13
14 variables,
15   continuous_design = 3
16   initial_point 0.16 0.12 2.808
17   descriptors 'AC' 'KD' 'CS'
18
19 interface,
20   system
21     analysis_driver = 'WinDam.bat'
22     parameters_file = 'params.in'
23     results_file = 'results.out'
24   file_save
25
26 responses,
27   num_objective_functions = 2
28
```

**Figure 18: Dakota Input File**

This file is sent as an input to the DAKOTA. DAKOTA supports interface mechanisms by which we are able to custom declare the type of processing needed. In the above screen shot, you will notice that we have defined the analysis drives as the .BAT file which contain the commands that will be executed. It takes the parameter from param.in file generated by the

DAKOTA. The below screenshot is the file generated using a windam.bat script during execution:

```

1 WINDAM 01/01/2009my project
2 OPTION SIMPLE EARTH BREACH
3 *
4 my comments go here
5 HCMODEL 1 0000.1600 0000.1200 0002.8280
6 HYD OUTLET 10.1263 6.770 0C
7 20 20.93423 21.71476 22.08148 22.47298
8 23.05004 23.64339 24.3587 25.09186 25.9587
9 26.84333 27.86116 28.89655 30.06337 31.24766
10 32.56278 33.89583 35.36321 36.84967 38.47906
11 40.12945 41.93718 43.76858 45.7772 47.81266
12 50.04906 52.31606 54.81213 57.34338 60.13809
13 62.97355 66.11409 69.3031 72.85506 76.46843
14 80.54167 84.69919 89.48779 95.10393 101.92789
15 110.6035 125.42834 145.84958 176.53256 213.48643
16 265.39757 321.61276 387.04935 448.71741 514.58731
17 575.74231 642.13202 704.26764 771.15637 835.117
18 896.4975 954.49841 1003.4917 1052.373 1086.8638
19 1120.277 1138.4196 1155.2197 1157.25 1158.3575
20 1148.2125 1137.0265 1114.7015 1091.7234 1061.0431
21 1029.825 992.73773 955.16608 912.16333 869.14227
22 824.73688 780.96033 741.21149 702.31598 669.16699
23 636.6319 608.12976 580.15302 555.27832 530.81848
24 509.07535 487.62347 468.0379 448.69345 430.91629
25 413.43588 397.85358 382.54761 368.98175 355.60303
26 343.50748 331.56912 320.7374 310.05023 300.40158
27 290.87363 282.22141 273.67336 265.8904 258.20702
28 251.26033 244.39922 238.17485 232.01765 226.34099
29 220.73138 215.56706 210.46646 205.73174 201.05701
30 196.69017 192.38396 188.38417 184.44266 180.80757
31 177.22429 173.92267 170.66464 167.66078 164.69786
  
```

**Figure 19: WDC File with Parameters**

With the help of the analysis driver, we replace the DAKOTA generated parameters in the template file and executed it against the WinDAM. The output generated are in results.out. This file is processed to extract a couple of values and save it in Dakota\_tabular.dat (Figure 12). It contains a table like structure which contains the execution run identification, the values of the parameters used and the output generated and stored in the working directory.

We can use these generated values and use it to plot the various possibilities of parameter values and understand the variations it may cause which can be avoided and used to avoid any issues with unforeseen circumstances.

## **Chapter 4 – Conclusions**

WinDAM has been developed over in stages to evaluate the performance of earth dams. The system provides tools that can be used to better understand the structure, function, and dynamics of such structures. In this report, I have described the development of a new software package which links the WinDAM development with DAKOTA for performing a variety of different parameter analysis studies.

For future work, we plan to incorporate a Graphics Module for plotting the output generated from the software and WinDAM using visualization software.

## References:

- [1] M.L. Neilsen, "Global sensitivity analysis of dam erosion models", in Proceedings of the 10<sup>th</sup> International Conference on Scientific Computing, Paper No. CSC-3502, July 22-25, 2013.
- [2] Brian M. Adams, Keith R. Dalbey, Michael S. Eldred, David M. Gay, Laura P. Swiler, William J. Bohnhoff, John P. Eddy, Karen Haskell, & Patricia D. Hough, Sophia Lefantzi, DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis, 2014.
- [3] D.M. Temple and G. J. Hanson, "Earth dam overtopping and breach outflow", In *Proc. of the World Water and Environmental Resources Congress*, Anchorage, Alaska, ASCE, 8 pp., 2005.
- [4] G.J. Hanson, K.M. Robinson, and K.R. Cook, "Prediction of headcut migration using a deterministic approach. Trans. ASAE, 44(3): pp. 525-531, 2001.
- [5] M.L. Neilsen, D.M. Temple, and J.L. Wibowo, "A distributed hydrologic simulation environment with latin hypercube sampling", In *Proc. of the Intl. Conf. on Env. Modelling and Simulation*, No. 432-032, St. Thomas, USVI, Nov. 22-24, 2004.
- [6] M. D. McKay, W.J. Conover, and R. J. Beckman, "A comparison of three methods for selecting values in the analysis of output from a computer code", *Technometrics*, 21(2):239-245, 1979.
- [7] K. Visser, R.D. Tejral, and M. Neilsen, "WinDAM C Earthen Embankment Internal Erosion Analysis Software", in Proceedings of the 30<sup>th</sup> Annual Conference of the Association of State Dam Safety Officials, Providence, RI, Sept. 8-12, 2013.