INTERNAL EROSION AND SIMPLIFIED BREACH ANALYSIS (UPGRADED VERSION 2012)

by

VIJAY SADHU

B.E., Osmania University, 2010

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

2012

Approved by:

Major Professor
Dr. Mitchell L. Neilsen

Abstract

In recent years, headline news has been overwhelmed with stories about dam and levee failures including the 2005 levee breaches in New Orleans and the 2006 Kaloko Damfailure in Hawaii that resulted in seven deaths. Since 2000, state and federal agencies have reported 92 dam failures in the United States to the National Performance of Dams Program. Incidents such as these have brought both national and worldwide attention to the need for improved flood warning systems and breach prediction tools for dam embankment and levee failures. (G. J. Hanson, 2010)

IESIMBA 2012 is an upgraded version of **SIMBA**, which has been upgraded from VB6 to C#.NET. The Microsoft Windows-based **SIM**plified **B**reach **A**nalysis software (**SIMBA**) was developed by the USDA Agricultural Research Service in cooperation with Kansas State University. The software was developed for the purpose of analyzing internal erosion, earth embankment breach test data and extending the understanding of the underlying physical processes of breach of an overtopped earth embankment. It is a research tool that is modified routinely to test the sensitivity of the output to various sub-models and assumptions. This software is a test version for use in validation testing of the simplified breach model based on stress and mass failure driven headcut movement. It runs under Microsoft Windows 98SE, Windows 2000, NT, XP, or Vista.

The following **Input Screens** are used to guide the user through development of input data sets.

- Model Properties , Dam Profile , Structure Table, Spillway Rating and Hydrograph Data

After an input data set has been entered, the data is saved and simulation can be performed on the data stored in memory at any time by selecting Build option. Input and output files are stored in a fixed ASCII text format. The results of the simulation can be viewed in graphical format which are of interest to the researchers at Oklahoma State University, Stillwater by selecting View option.

Table of Contents

List of Figures

Acknowledgements

I offer my sincere gratitude first to my major professor **Dr. Mitchell L. Neilsen** for giving me an opportunity to work on this challenging project and for his constant encouragement throughout the project. I truly feel privileged to have worked under him.

I would like to thank Dr. Daniel Andresen and Dr. Torben Amtoft for graciously accepting to serve on my committee.

I would like to thank my family members and friends for supporting and encouraging me to pursue this degree. Their encouragement was a key factor in successful completion of my degree.

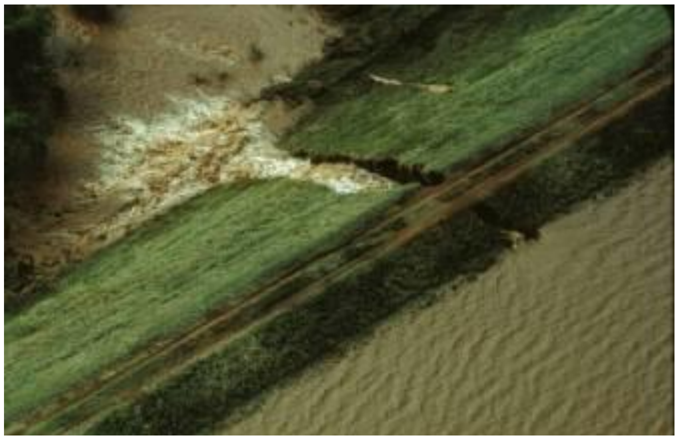
Finally, I would like to thank the administrative and technical support staff of the department of CIS for their support throughout my graduate study.

Dedication

I would like to dedicate this project to my parents Surender & Dhanalakshmi.

Chapter 1 - Introduction

The two most common causes of earthen embankment and levee failure are embankment overtopping and internal erosion. Internal erosion occurs when water flows through a cavity, crack, and/or other opening within the embankment. These openings may be a result of inadequate compaction during construction, differential settlement, desiccation, earthquakes, burrowing animals, and/or vegetation roots. Figure 1.1 illustrates an internal erosion event at Natural Resources Conservation Service (NRCS) structure.



(G. J. Hanson, 2010)
Figure 1.1 Internal Erosion Event, Upper Red Rock

The Microsoft Windows-based **I**nternal **E**rosion and **SIM**plified **B**reach **A**nalysis software (**IESIMBA**) was developed by the USDA Agricultural Research Service in cooperation with Kansas State University.

The software was developed for the purpose of analyzing

- Internal Erosion
- Earth embankment breach test data
- Extending the understanding of the underlying physical processes of an overtopped earth embankment breach.

It is a research tool that is modified routinely to test the sensitivity of the output to various sub-models and assumptions. This software is a test version for use in validation testing of the simplified breach model based on stress and mass failure driven headcut movement.

IESIMBA 2012 is an upgraded version of **IESIMBA 2010**, which has been upgraded from VB6 to C#.NET

1.1 Motivation

IESIMBA is used to determine the breach rate and outflow hydrograph for overtopped homogeneous earth embankments. It is a simplified model developed from the concept of using fundamental building blocks and testing the sensitivity of output to refinement in the overall model. In its present form, IESIMBA is limited to the evaluation of overtopping of homogeneous earth embankments with negligible protection on the downstream face. The model simulates the four stages of the failure process observed for these embankments. These stages are: 1) surface erosion leading to development of a headcut on the downstream face of the embankment, 2) headcut advance through the crest to initiate the breach, 3) breach formation as the headcut advances into the reservoir, and 4) breach expansion during reservoir drawdown.

1.2 Project Description

The following **Input Screens** are used to guide the user through development of input data sets.

- Model Properties
- Dam Profile
- Structure Table
- Spillway Rating
- Hydrograph Data

After an input data set has been entered, the data should be saved. A simulation can be performed on the data stored in memory at any time by selecting Build. Input and output files are stored in a fixed ASCII text format. The results of the simulation can be viewed in graphical format by selecting View and the corresponding graph of interest.

1.3 Graphical Representation of Upgrade Process

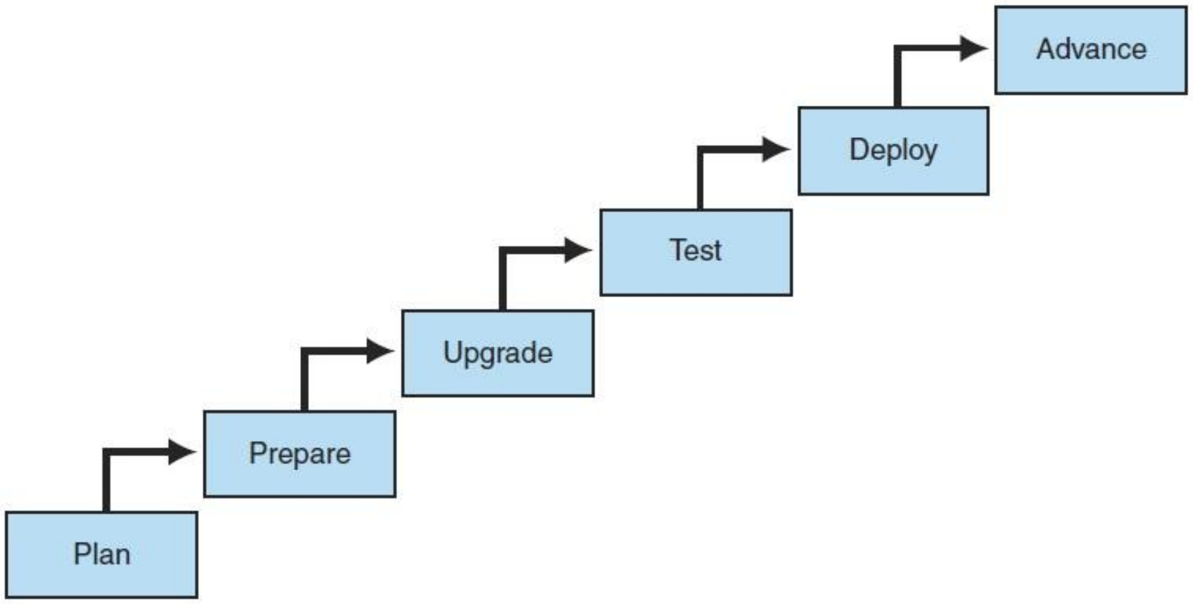


Figure 1.2 Upgrade Process

1.4 Planning the Upgrade

Apart from normal software development life cycle being followed upgrading project should consider to follow these steps

- Defining the project scope
- Performing an application analysis
- Assessing the current and the target architecture
- Analyzing and designing new functionality
- Selecting an upgrade strategy
- Making an inventory of source code
- Preparing the source code
- Preparing to handle upgrade issues
- Unit testing
- Validating functional equivalence and acceptance
- Implementing and testing new functionality
- Deploying the C# .NET application

Chapter 2 - User Interface

Requirement analysis is done by Reverse Engineering of SIMBA software. SIMBA was developed in VB6 which is a component-oriented and event-driven programming language. It consists of seven windows forms, various variables, data structures involving operations on them and finally generating graphs for the resulting values. The following are the forms which can get the input for their textboxes by reading a specific format text document or by user entered values.

2.1 Home

This is the Home Page of the application, parent of all other forms.

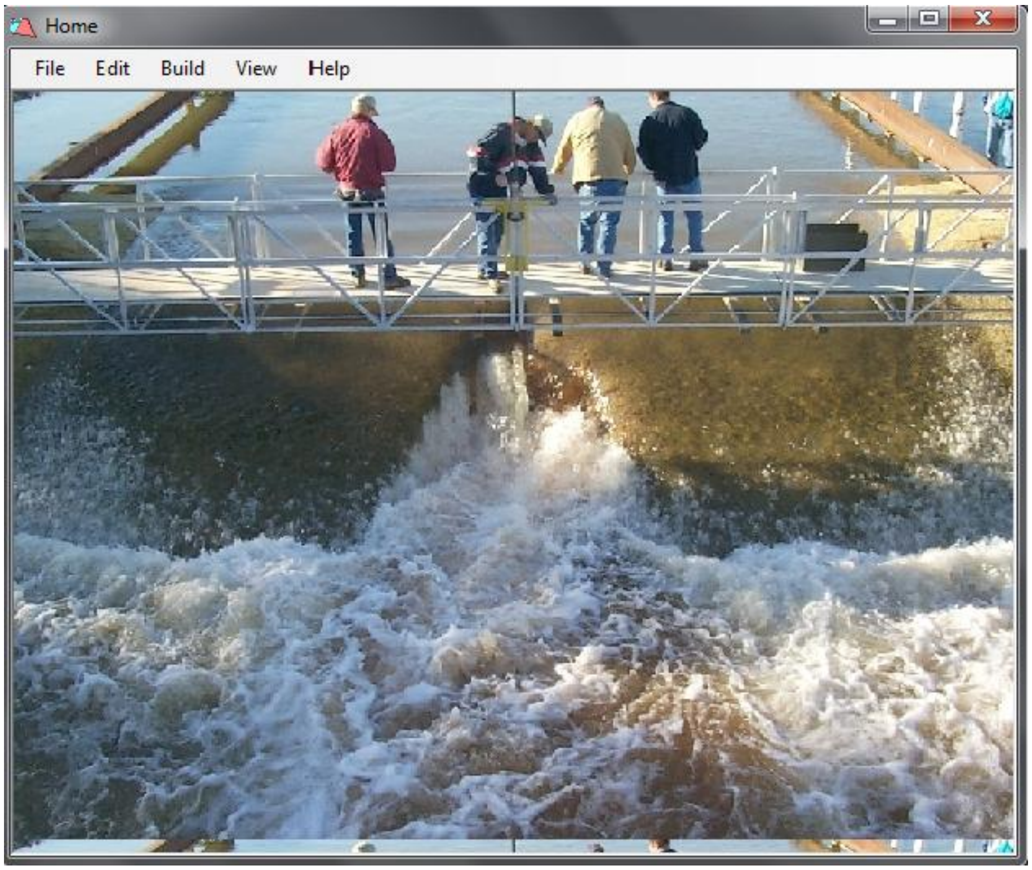


Figure 2.1 Home Screen

2.2 Model Properties

Model Properties is the first child of Home. It is organized with three tabs: Options and Constants, Computation Entry Point, and Headcut Advance Model.

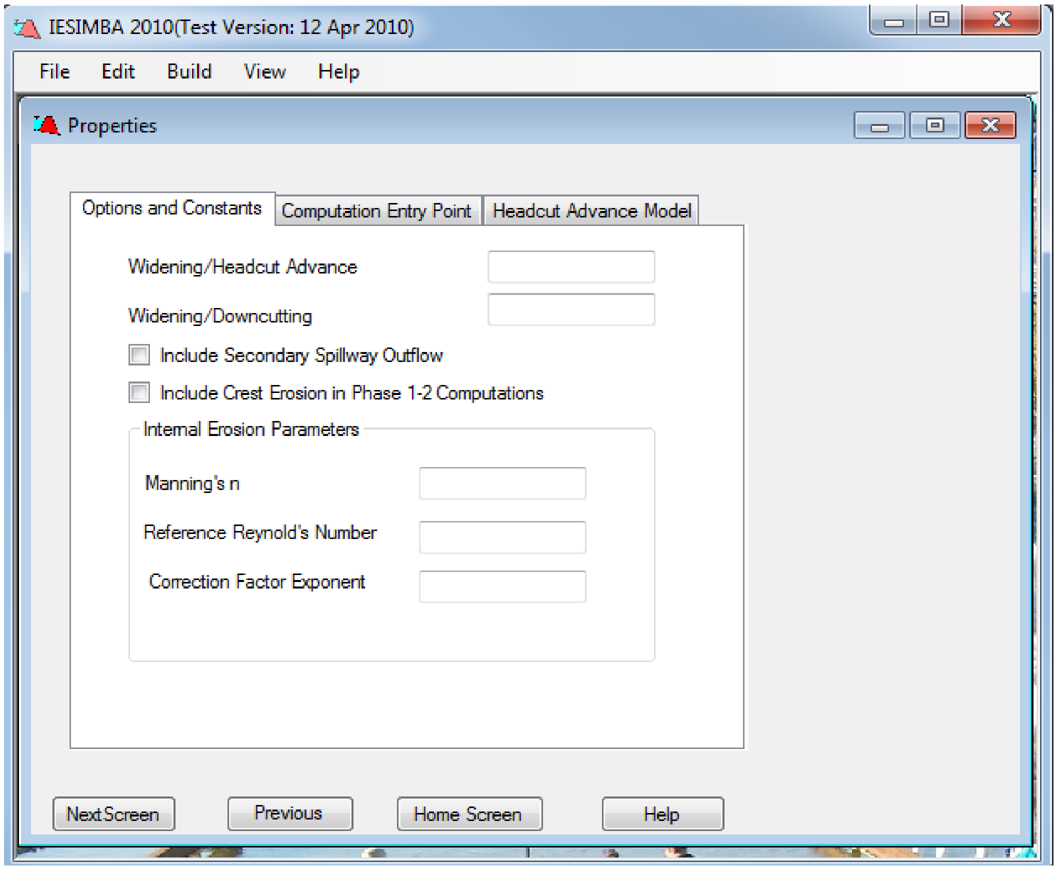

IESIMBA 2010(Test Version: 12 Apr 2010)
File
Edit
Build
View
Help
Properties
Options and Constants
Computation Entry Point
Headcut Advance Model
Widening/Headcut Advance
Widening/Downcutting
Include Secondary Spillway Outflow
Include Crest Erosion in Phase 1-2 Computations
Internal Erosion Parameters
Manning's n
Reference Reynold's Number
Correction Factor Exponent
Next Screen
Previous
Home Screen
Help

Figure 2.2 Model Properties Screen

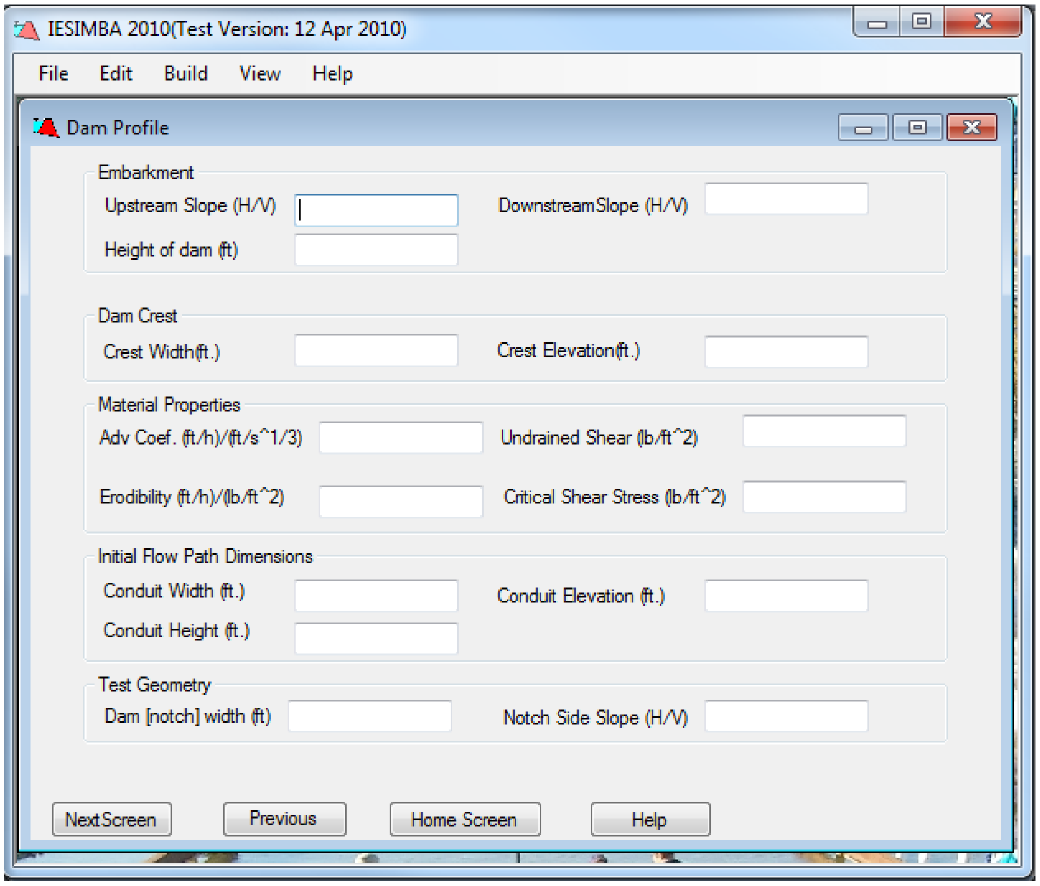
Under *Options and Constants*, the user can indicate whether there is a secondary spillway outflow.

Under *Computation Entry Point*, the user is currently limited to the default option of Stage 1 Entry with an Intact Embankment for entry at stage 1 of the erosion process.

Under *Headcut Advance Model*, the user is currently limited to two options and both are preselected. The Headcut Advance Form is set to continuous movement, and the Headcut Model is set to the Hanson/Robinson Model.

2.3 Dam Profile

Dam Profile is the second child of Home and the successor of Model Properties form. This screen is used to input the description of the embankment dimensions and material properties. Note that only a simple embankment cross-section is presently available and no provision is made for such things as vegetal protection of the face. Only homogeneous embankments are evaluated.



Embankment

Upstream Slope (H/V)		DownstreamSlope (H/V)	
Height of dam (ft)			

Dam Crest

Crest Width(ft.)		Crest Elevation(ft.)	

Material Properties

Adv Coef. (ft/h)/(ft/s^1/3)		Undrained Shear (lb/ft^2)	
Erodibility (ft/h)/(lb/ft^2)		Critical Shear Stress (lb/ft^2)	

Initial Flow Path Dimensions

Conduit Width (ft.)		Conduit Elevation (ft.)	
Conduit Height (ft.)			

Test Geometry

Dam [notch] width (ft)		Notch Side Slope (H/V)	

Figure 2.3 Dam Profile Screen

These are the different properties captured in Dam Profile

Embankment

- Upstream Slope (H/V)
- Downstream Slope (H/V)
- Height of Dam (ft)

Dam Crest

- Crest Width (ft)
- Crest Elevation (ft)

Material Properties

- Total Unit Weight (lb/ft^3)
- Undrained Shear (lb/ft^2)
- Erodibility (ft/h)/(lb/ft^2)
- Critical Shear Stress (lb/ft^2)

Test Geometry

- Dam (Notch) Width (ft)
- Notch Side Slope (H/V)

2.4 Structure Table

Structure table is the successor of Dam Profile form. This screen contains the stage storage information in tabular form with ordered pairs of elevation in feet (ft) and volume in cubic feet (ft^3) or acre feet (ac-ft), units to be selected by user. Entries in the table are interpolated linearly during calculation.

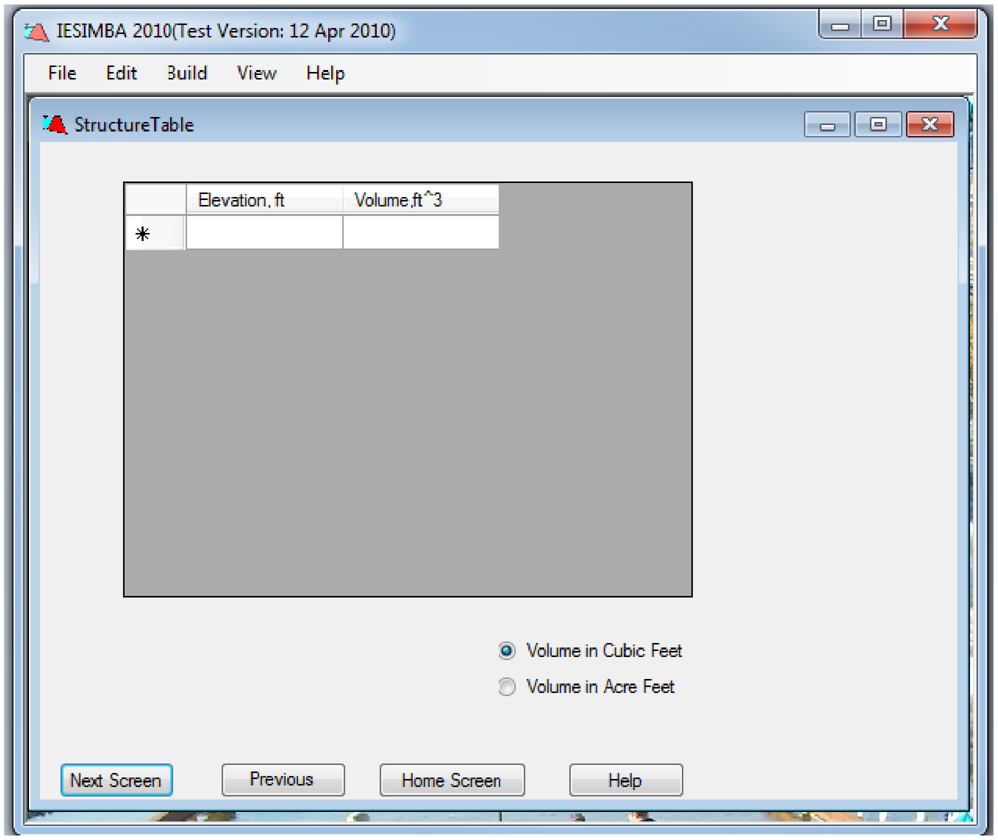


Figure 2.4 Structure Table screen

Here the fields include

- Elevation (ft)
- Volume (ft^3 or ac-ft)
- Elevation to Start Routing (ft)

2.5 Spillway Rating

Spillway is the successor of Structure Table Form, Sometimes this form can be skipped if there it is not required. This screen includes the Spillway Rating Table consisting of ordered pairs of reservoir water surface elevation in feet (ft) and discharge through the spillway in cubic feet per second (cfs). The table is interpolated linearly during program execution. Elevations are referenced to the same coordinate system as used for other elevation entries.

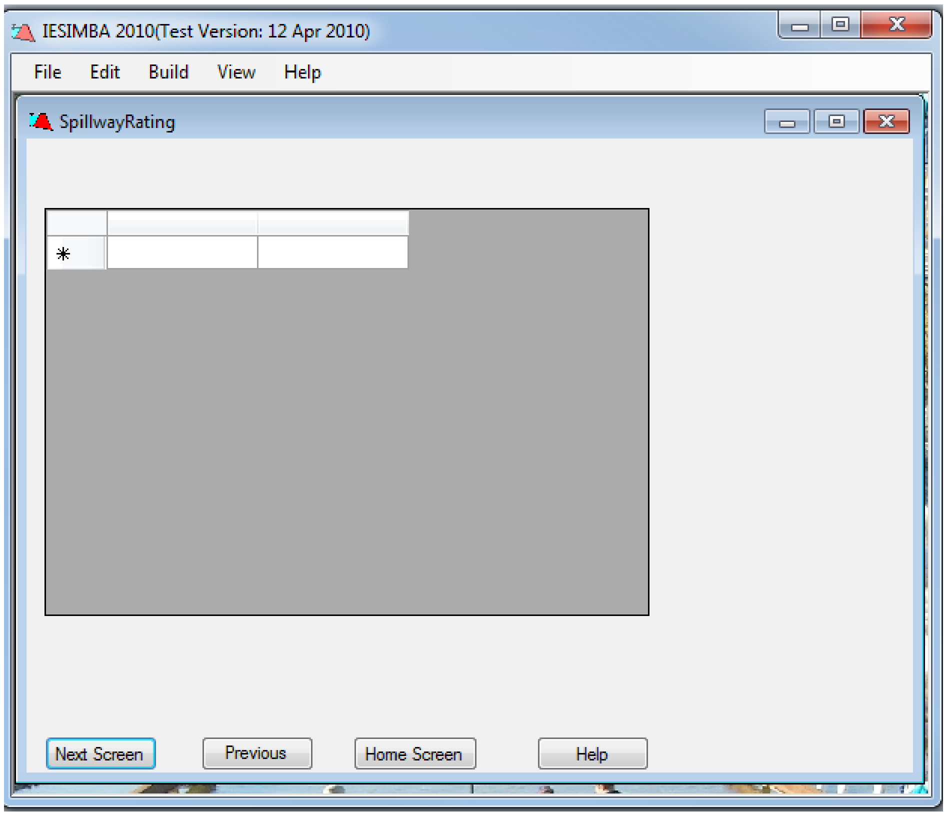


Figure 2.5 Spillway Rating Screen

Here the fields include

- Elevation (ft)
- Discharge (cfs)

2.6 Hydrograph Data

Hydrograph Data is the successor of either Structure Table or Spillway Rating form. This screen is used to enter the inflow hydrograph for the test or flood. A computational time increment must be provided. The specified time increment will be used in routing the flow through the reservoir and computing the erosion in the breach area. The user can enter a hydrograph with variable time steps but SIMBA will convert the variable time step hydrograph to a constant time increment hydrograph for computation purposes. The discharge for a given time step will be determined by interpolation of the input hydrograph and will be provided in the text output file. The computational time increment is entered in hours. The initial time of the inflow hydrograph must start with time zero.

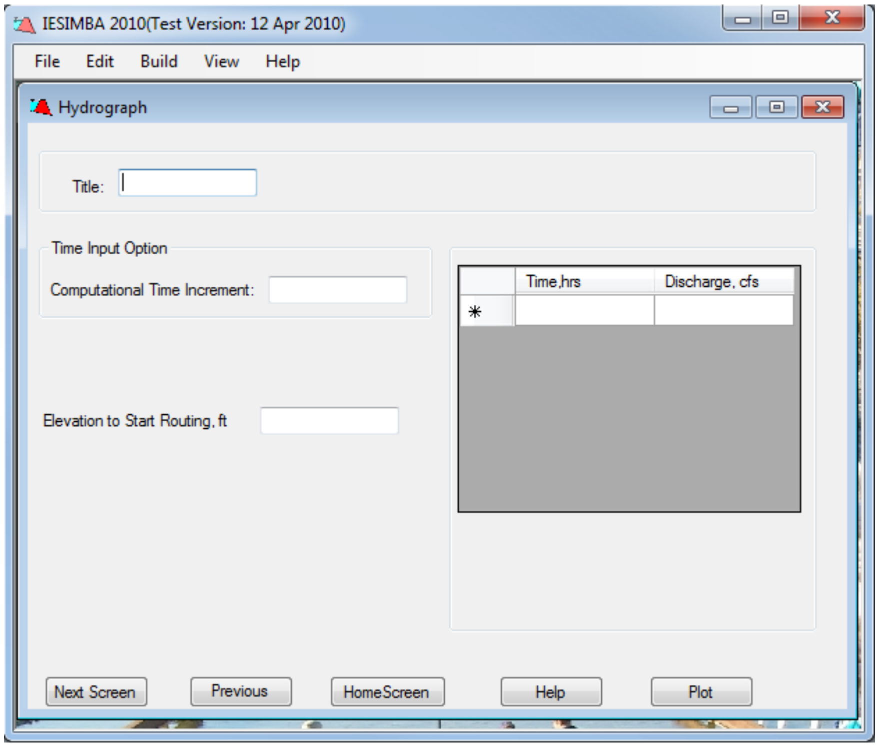


Figure 2.6 Hydrograph Data Screen

Here the fields include

- Title
- Computational Time Increment
- Time (hrs)
- Discharge (cfs)

The Plot button generates graph for Time/Discharge values in the grid table of Hydrograph. Help button all over the application opens a compiled HTML file with information specific to the form currently active.

2.7 Input / Output Files

Users are advised to make all entries through the IESIMBA user interface. All inputs are written to an ASCII text file with a default “.txt” extension. Inputs have been defined as they appear in the user interface. When a simulation is performed, two output ASCII text files are created with the same filename, but extensions “.out” and “.evt”. The “.evt” file contains a copy of the elevation-volume table.

Chapter 3 - Implementation

This is completely a new application incorporating the business logic of its previous version. It is developed in C#.NET using Visual Studio 2010 (IDE).Graphs are generated with the help of Windows Chart Control. The application has 7 Windows Forms and one Variable class where entire logic resides. It uses text documents for input to the application and also the results are written to various text documents with extensions ".evt" ,".out" and ".sof".

Lines of Code (LOC) written in C# to implement this application are approximately 7500 excluding code generated by the system and CSS. A total of 900 working hours spanning over 7 months have been spent on developing this application. Testing of this application took around 100 hours for its correct functionality after the application has been fully developed.

3.1 Lesson's learned

3.1.1 Things to remember for converting an application from VB6 to C#.NET

The most profound changes arise from the fact that C#.NET is based on the new Common Language Runtime (CLR). The CLR is the execution platform for all .NET apps, including those developed in VB, managed C++, C#, and other languages. VB has undergone the most significant changes ever in the language as a result of the new programming model in the CLR. In fact, you should rewrite an app rather than port it to take best advantage of VB.NET's new features and structures.

- There is no tool available in market that can convert whole VB6 application to C#.NET
- Converting VB6 -> VB.NET -> C#.NET with the help of available tools is not feasible for large applications like SIMBA involving many operations. It requires a lot of manual work to be done.
- The best way is to start building the whole new application in C#.
- This involves converting an application from a component-oriented, event-driven programming language to an object oriented programming language.
- One should have knowledge of what the application is doing before starting the project.

- Start with designing the forms, using exact names used in the previous version for naming the entities.
- There are many methods in the application for drawing graphs which we don’t use. Don’t panic by looking at them. Windows Chart Control is a very good package which makes our work easy.
- In VB6 the default way of passing parameters to methods is pass by reference. If you have to pass by value use ByVal keyword.
- When initializing the arrays use the same index number as in VB6. If index is declared Public T(1 To 5000) As Double *Use* public static double[] T = new double[5001]; *T[0] remains unused*.
- The same thing applies for iterating the loops.
- Be sure you will not miss any line in code while converting, testing phase would be harder.
- Be sure to end the loops correctly, should be lot careful when you are not sure where the loop is going to be end.
- Don’t start with converting the back-end code without knowing from where it is called from or from where it is invoked. Do the other way.
- Type casting differences:
  int/int=int in c#
  int/int can be a double in vb6
  For Example: dc = ((qbl * qbl) / gravity) ^ (5 / 3) and
  dc = Math.Pow((qbl * qbl) / gravity, (5 / 3)); equivalent? NO
  5/3 is 1.67 in vb6
  5/3 is 1 in C #
  We have to use 5.0/ 3.0 in C#
- Accessing datagridview in C# differs from datagrid of MSFlexControl in VB6. Code should be written newly with the help of logic in the previous version.
- Don’t forget to include comments while converting, as this is a research tool people will be updating it.
- Maintaining a book with daily progress of the project should save time to get back to work when you come back after a break.

- In Windows Chart Control use FastLine graph instead of Spline, this increases the through-put when used for large data sets.

3.1.2 Typical Example

```
If Flcd=2 then (loop 1 starts)
If  qd(i)=0 then cms(i)=0.0(loop 2 starts and ends here )
…
..
.
If qb(i + 1) > 0# Then
   If Flcd < 3 Then  'only pipe flow - Only average stress is applicable
    Cms(i + 1) = 0#         ' Set Critical flow stress to zero
   Else
    dcd = (qb(i + 1) * qb(i + 1) / g) ^ (1# / 3#)
    rcd = dcd
    vcd = qb(i + 1) / dcd
    scd = (vcd * n / (1.49 * rcd ^ (2# / 3#))) ^ 2
    Cms(i + 1) = 62.4 * dcd * scd ' critical flow stress for wide channel condition
   End If
   If Flcd < 4 Then
    rcd = (Wb(i + 1) * Ch(i + 1)) / (2 * (Wb(i + 1) + Ch(i + 1)))
    vcd = Qbr / (Wb(i + 1) * Ch(i + 1))
    scd = (vcd * n / (1.49 * rcd ^ (2# / 3#))) ^ 2
    Cs(i + 1) = 62.4 * rcd * scd   'mean conduit stress
   ElseIf Xc(8) > Xc(1) Then
    dcd = (qb(i + 1) * qb(i + 1) / g) ^ (1# / 3#)
    rcd = (dcd * Wb(i + 1)) / (Wb(i + 1) + 2# * dcd)
    vcd = qb(i + 1) / dcd
    scd = (vcd * n / (1.49 * rcd ^ (2# / 3#))) ^ 2
    Cs(i + 1) = 62.4 * rcd * scd    ' mean stress on rectangular channel at critical depth
   **Else**
    **Cs(i + 1) = 0#**
   **End If**
 Else
   Cms(1) = 0#
   Cs(1) = 0#
 End If

Else(loop 1 ends here – don’t confuse this with the end of loop2)
 Cms(1) = 0#
 Cs(1) = 0#
```

Chapter 4 - Testing

4.1 Manual Testing

Manual testing is the process of manually testing software for defects. Functionality of this application is manually tested to ensure the correctness. (http://en.wikipedia.org/wiki/Software_performance_testing)

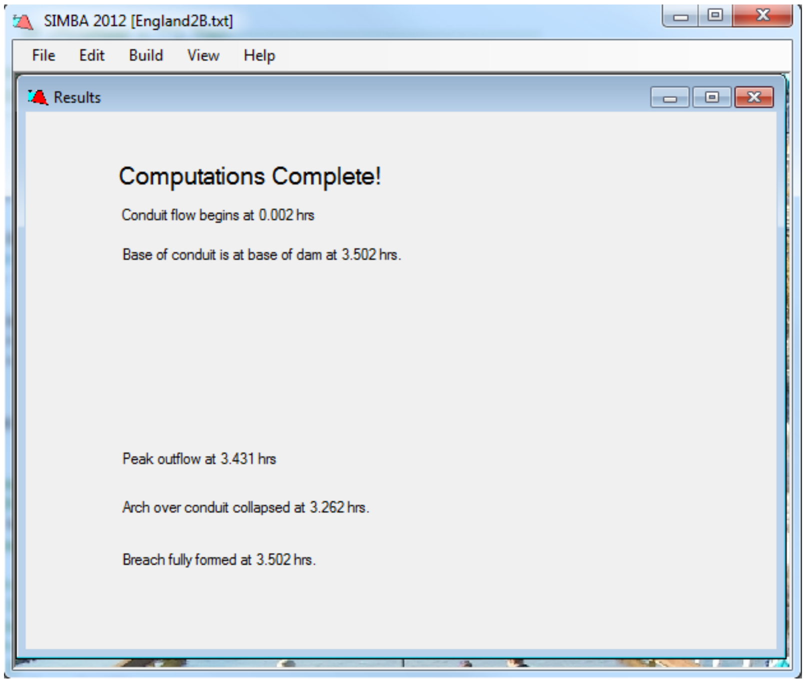
Most crucial phase of the application is testing if the algorithm is working well and this certifies that the things analyzed in the requirement phase are correct and we are in the right path. I did testing in three phases, first is the correct working of the application without algorithm (i.e. reading the files and application progress), second is getting the right results after the compute function is executed and checking if the text files are updated with specific format, finally getting the graphs right completed successful testing of the application.

Few example Test cases are listed below

- Each and every line is converted
- Verify every form is reading the right values what it has to read after parsing the line
- Write the information on to a file in the correct format and crosscheck the values
- Are the forms in the right path
- For each and every data set available, match the results of a data set with the results its previous version
- No Exceptions caught
- Getting the graphs right

Chapter 5 - Results and Challenges

5.1 Result Window



Computations Complete!

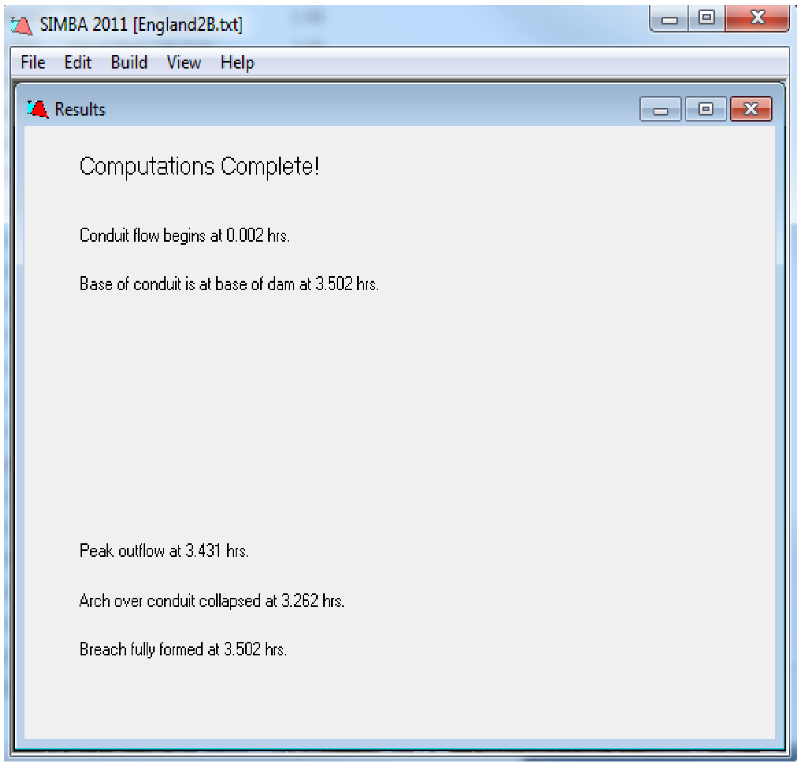
Conduit flow begins at 0.002 hrs

Base of conduit is at base of dam at 3.502 hrs.

Peak outflow at 3.431 hrs

Arch over conduit collapsed at 3.262 hrs.

Breach fully formed at 3.502 hrs.



Computations Complete!

Conduit flow begins at 0.002 hrs.

Base of conduit is at base of dam at 3.502 hrs.

Peak outflow at 3.431 hrs.

Arch over conduit collapsed at 3.262 hrs.

Breach fully formed at 3.502 hrs.

This output includes:

- The time that overtopping of the embankment begins referenced to zero time at the first point on the inflow hydrograph. Prior to this point flow is being stored in the reservoir or passing out through the secondary spillway.
- The time that the headcut begins to advance. This is the point where the computed erosion depth exceeds the critical depth of the flow passing over the forming headcut.
- The time that breach begins is the beginning of Stage 3, when the headcut enters the reservoir.
- The time when the breach is complete is the time that the embankment is entirely removed in the breach area and the toe elevation is the hydraulic control in the breach area.

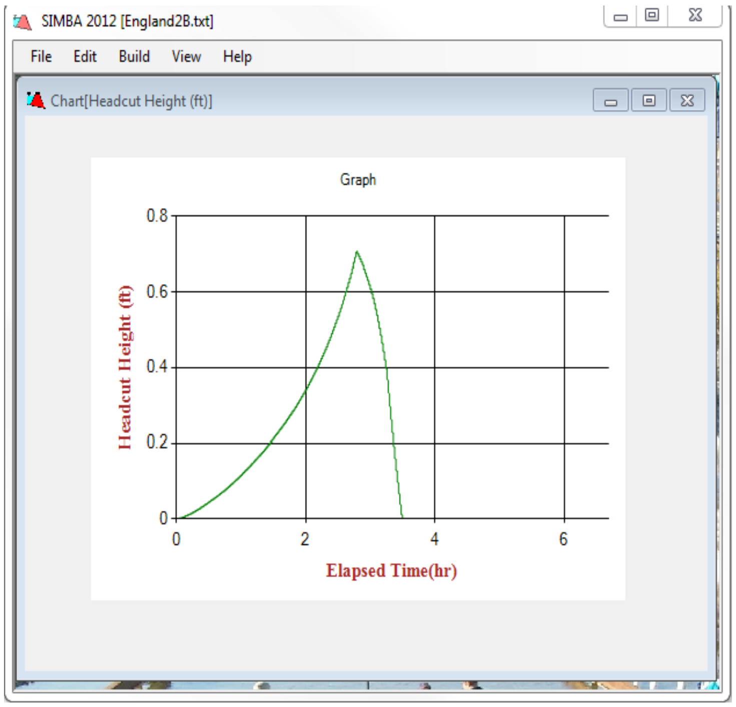
In the absence of a complete breach, the final headcut height and headcut position relative to the downstream edge of crest are reported.

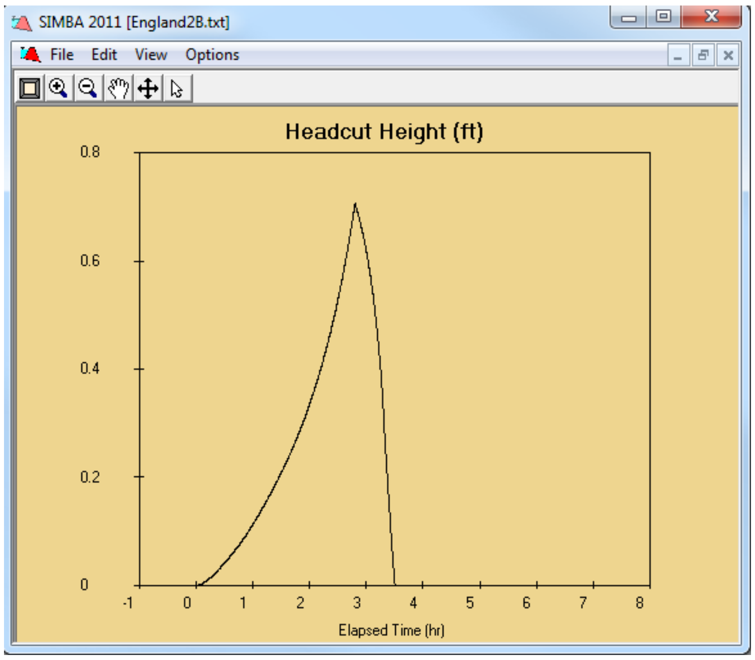
- The maximum headcut height (most often equal to dam height) and the final computed width of the breach are also displayed on the output screen.

5.2 Graphical Output

Additional information is written to a text output ".out" file and may be viewed graphically. These include:

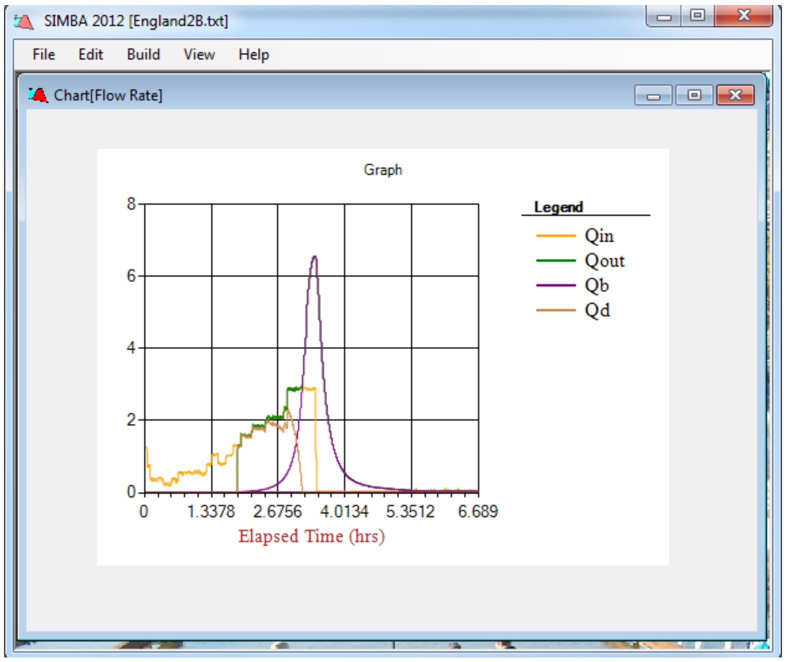
The **headcut height** as a function of time.

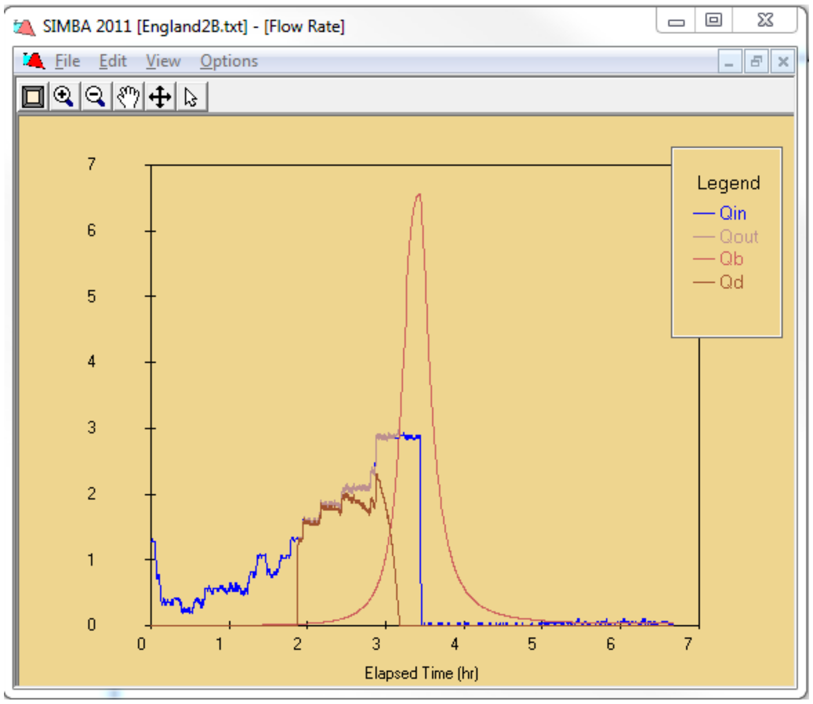

SIMBA 2012 [England2B.txt]
File Edit Build View Help
Chart[Headcut Height (ft)]
Graph
Headcut Height (ft)
Elapsed Time(hr)
0.8
0.6
0.4
0.2
0
0
2
4
6



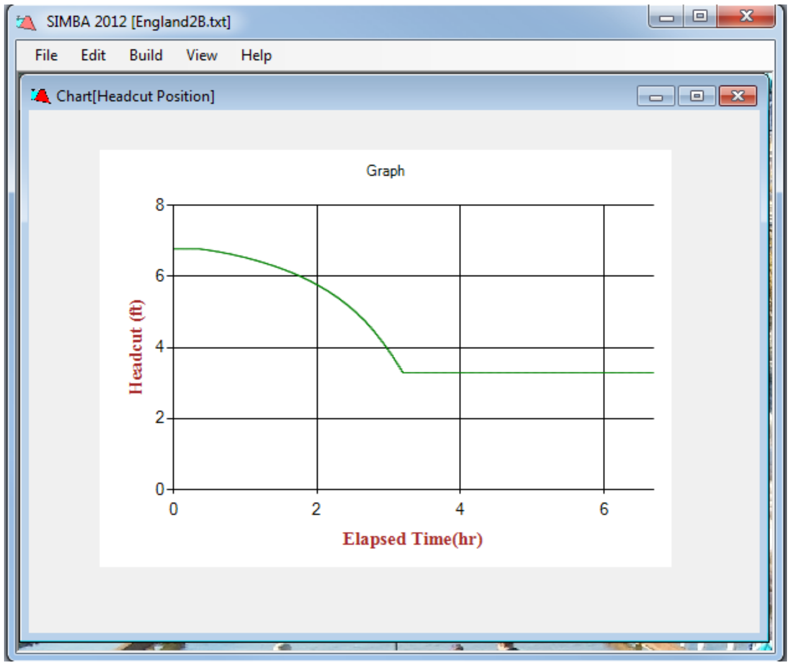
The **inflow and outflow hydrographs** including:

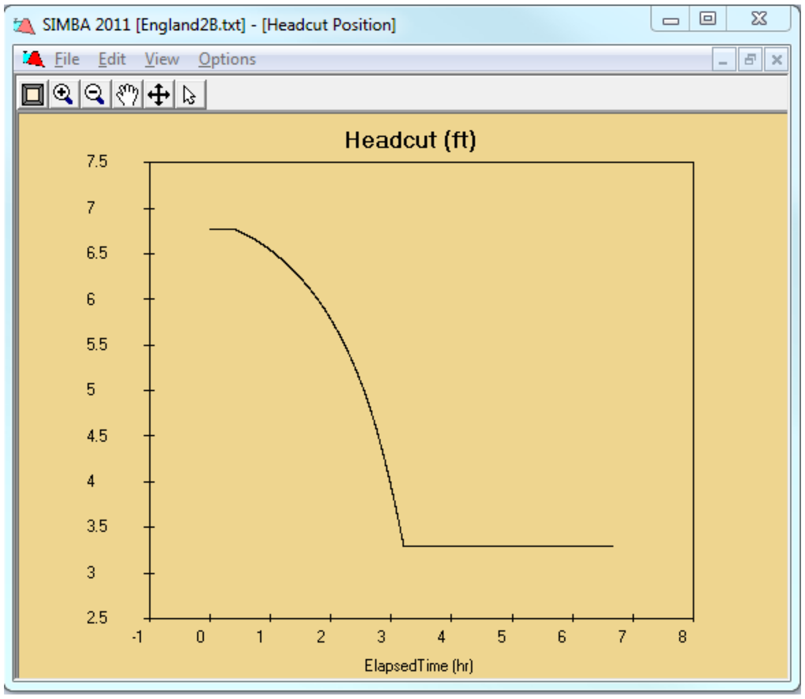
- outflow through the breach area (Qb),
- outflow over the top of the dam that is not through the breach width (Qd),
- the total outflow, and
- the inflow.



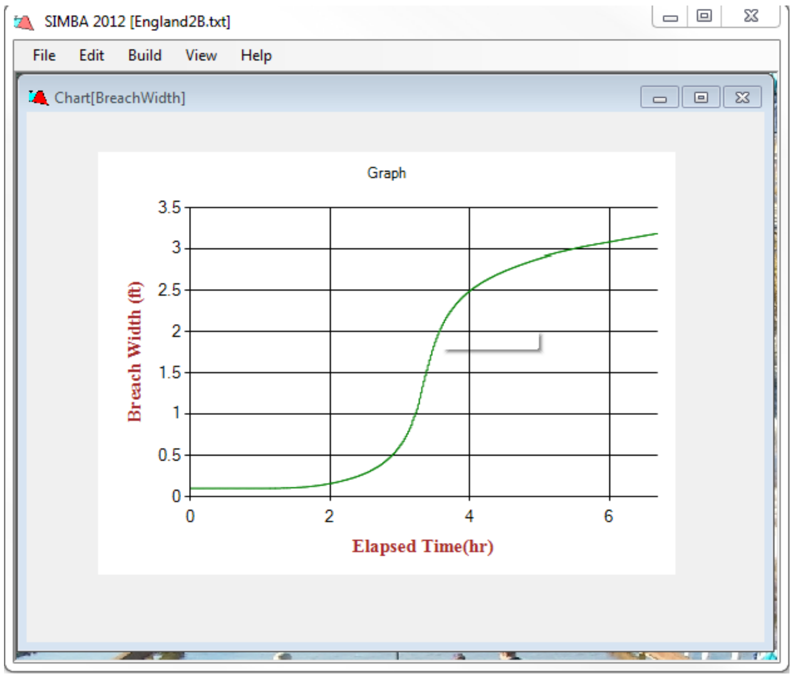


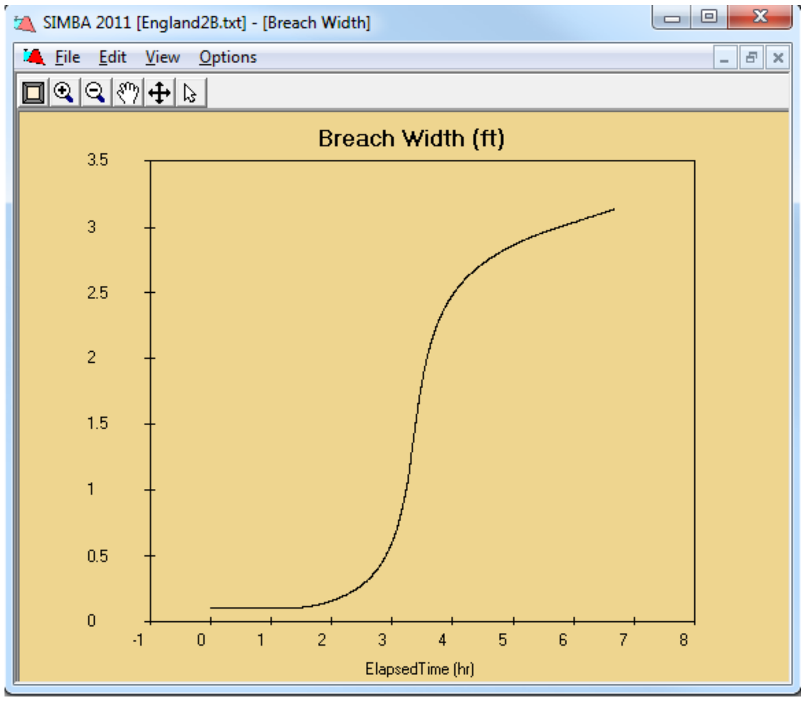
The **position of the headcut** as a function of elapsed time, relative to distance from the upstream toe. Initial position of the headcut is the downstream edge of the crest and the position of the upstream toe is zero.



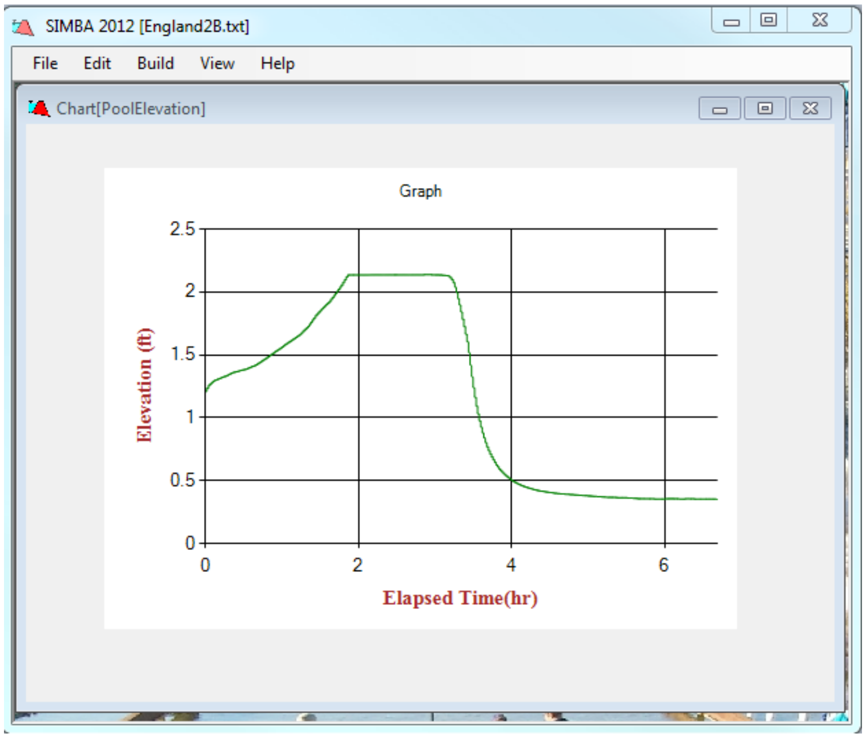


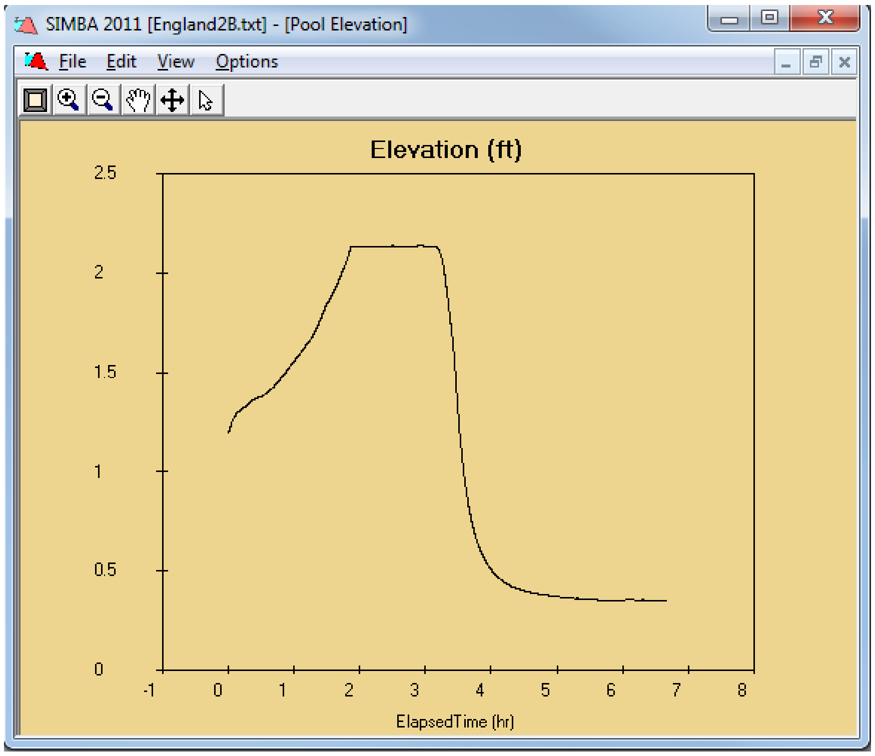
The **breach width** as a function of time. A period of constant or near-constant breach width may occur during stage 2 as a result of the breach width being constrained to not be larger than the overtopping notch width at ½ flow depth.





The **water surface elevation** in the reservoir as a function of elapsed time . The elevation uses the same reference as the structure and rating table inputs.





5.3 Text Output

All text output is written to an ASCII text file with the ".out" extension containing the following columns:

- Time: elapsed time in hours at intervals as determined by "Computational Time Increment."
- Qin: Flow into reservoir in cubic feet per second (cfs).
- Qout: Sum of flows out of reservoir in cubic feet per second (cfs). Constiturents of this flow rate are overtopping flow through breach, overtopping flow outside of breach, and flow routed through secondary spillway.
- Qb; Flow through breach width in cubic feet per second (cfs).
- Qd: Flow over top of dam, but outside of defined breach width in cubic feet per second (cfs)
- El: Reservoir water elevation in feet (ft).
- HC Pos: The station of the headcut in feet (ft), measured as a positive distance from upstream toe.
- Attack: Does not apply to the current model.
- Hh: Headcut height in feet (ft).
- Width: Breach width in feet (ft).
- Hc: Headcut elevation in feet (ft).

At the conclusion of the ".out" file, the times of a few key events are summarized. All times are in hours (hr).

- OTBegins: Time at which overtopping has begun.
- HCAdv : Time at which headcut has begun to advance.
- BreachSt: Time at which breach begins, i.e. headcut enters reservoir.
- BrEnd: Time of breach completion, i.e. erosion has reach upstream toe.

If the event did not occur, a zero will be printed in the output.

5.4 Challenges

- Understanding what the application is doing by reverse engineering the software
- Getting to know the syntax of VB
- Testing the application
- Making it run faster
- Converting without missing any single line
- Estimating time for a task

Chapter 6 - Scope, Conclusions and Future Work

6.1 Scope of the Project

- Now the advancements could be made to this functionally equivalent application in C#.NET using a better IDE.
- Resources are readily available to work on this application.
- Graphs could be made to look more interesting as Widows Chart Control is capable to produce 3D and drill down graphs.
- Could be integrated in an IDE to the set of all flood warning and breach prediction tools.
- Application code and report could be used by others performing upgrade of applications like IESIMBA to get a vague idea to start the process.

6.2 Conclusions

The application has been designed successfully to meet all the requirements. Anyone who has access to sites group can download and run on a windows machine.

This software has not been fully developed and tested for field application. This application has all the required comments which are helpful when updating the software.

6.3 Future Work

I would like to include my experiences, challenges faced and guidelines for converting applications from VB6 to C# in my blog.

Chapter 7 - Bibliography

1. *Internal Erosion and Impact of Erosion Resistance.* **Hanson, G.J.** 2011.

2. *INTERNAL EROSION AND IMPACT OF EROSION RESISTANCE.* **G. J. Hanson, R.D. Tejral, S. L. Hunt, D.M. Temple.** 2010. USSD. pp. 773-784.

3. http://en.wikipedia.org/wiki/Software_performance_testing. [Online]

4. **Microsoft.** Chart Control. [Online] http://msdn.microsoft.com/en-us/library/dd456632.aspx.

5. vb6. [Online] http://www.vb6.us/.

6. **Microsoft.** MS Chart Control. [Online] http://msdn.microsoft.com/en-us/library/system.windows.forms.datavisualization.charting.aspx.

7. **DeveloperFusion.** Df. [Online] http://www.developerfusion.com/tools/convert/vb-to-csharp/.

8. **Anonymous.** codeproject. [Online] http://stackoverflow.com/questions/2089689/row-copy-paste-functionality-in-datagridviewwindows-application.

9. *Code Project.* [Online] http://www.codeproject.com/Articles/65803/A-Guide-to-using-MSChart-for-NET.

10. **Anonymous.** [Online] http://ittecture.wordpress.com/2009/01/11/tip-of-the-day-89/.