Test Report: Slope Stability analysis Program (SSP)

Brooks MacLachlan

December 10, 2018

1 Revision History

Date	Version	Notes
12/09/18	1.0	Initial write-up of document

2 Symbols, Abbreviations and Acronyms

The symbols, abbreviations, and acronyms used in this document include those defined in the table below, as well as any defined in the tables found in Section 2 of the Software Requirements Specification (SRS) document.

symbol	**description**
MIS	Module Interface Specification
MG	Module Guide
TC	Test Case
VnV	Verification and Validation

Contents

List of Tables

List of Figures

This document outlines the results of testing for SSP. Section 3 reports on the Test Cases (TCs) for functional requirements and Section 4 reports on the tests for non-functional requirements, all of which are described in the System Verification and Validation (VnV) Plan for this project. Section 5 compares this implementation of SSP to the original implementation. Section 6 reports on the results of the unit tests, which are described in the Unit VnV Plan for this project. Section 7 comments on changes to the project that that came as a result of the testing. Section 8 describes how the tests were implemented with an automated testing framework. Sections 9 and 10 show the traceability between test cases and requirements and modules. Supporting documents and other resources, such as the VnVPlans and original implementation, can be found on the GitHub repository for this project.

3 Functional Requirements Evaluation

The System VnV Plan described TC1 - TC30 for testing the requirement for verifying that inputs meet physical constraints. All of these tests passed.

The next test in the System VnV Plan, TC31 verified the calculation of the factor of safety by comparing it to literature sources for a common example problem. The factor of safety calculated by SSP for this example problem was 1.3200. The average relative errors between SSP and each literature source are shown in Table 1. The factor of safety calculated by SSP was within 1 percent of all but one of the literature sources. The relative tolerance was retroactively set as 0.1 for these tests to pass. Note that this test originally failed due to a bug for the case where there was no water table, which is discussed further in Section 7.

Source	**Factor of Safety**	**Relative Error**
Greco (1 July 1996)	1.3270	0.0086
Malkawi et al. (1 August 2001)	1.2380	0.063
Cheng et al. (27 Decemeber 2006)	1.3250	0.0071
Li et al. (25 June 2010)	1.3270	0.0086

Table 1: Relative error between SSP and literature for calculation of factor of safety

The next test in the System VnV Plan, TC32 verified the determination of the critical slip surface again by comparing it to literature sources. The average relative errors between SSP and each literature source are shown in Table 2. The critical slip surface calculated by SSP was within 2 percent of all but one of the literature sources. The relative tolerance was retroactively set as 0.1 for these tests to pass.

Source	**Relative Error**
Greco (1 July 1996)	0.013
Malkawi et al. (1 August 2001)	0.072
Cheng et al. (27 Decemeber 2006)	0.013
Li et al. (25 June 2010)	0.013

Table 2: Relative error between SSP and literature for calculation of critical slip surface

The next test in the System VnV Plan, TC33 verified the calculation of the factor of safety by comparing it to the original implementation for a specific example. The factor of safety calculated by SSP for this example problem was 0.9808, compared to 0.9835 for the original program. The average relative error between SSP and the original program was 0.0027. The relative tolerance was retroactively set as 0.01 for these tests to pass.

The next test in the System VnV Plan, TC34 verified the calculation of the critical slip surface by comparing it to the original implementation for a specific example. The critical slip surface calculated by SSP for this example problem is shown in Figure 1. The average relative error between SSP and the original program was 0.019 The relative tolerance was retroactively set as 0.05 for these tests to pass.

The next tests in the System VnV Plan, TC35 and TC36 verified the calculation of the interslice normal and shear forces by comparing them the original implementation for a specific example. Originally, the relative error between the current implementation and the original implementation was unexpectedly high, which led to the discovery of a bug in the current implementation,

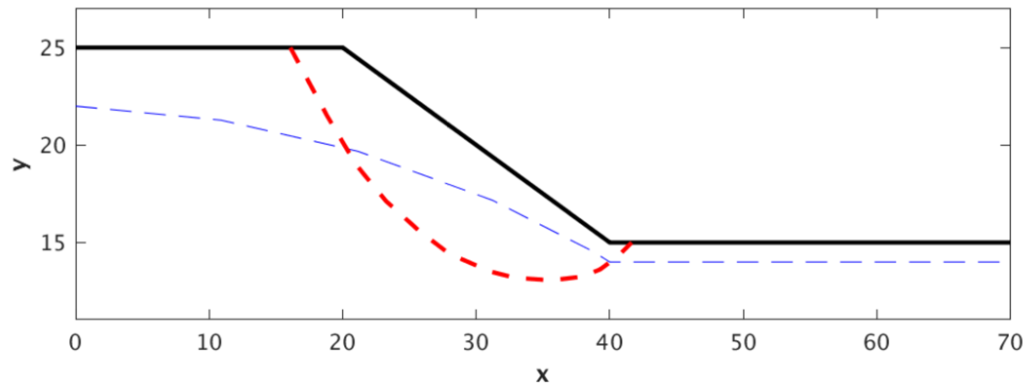

Figure 1: The critical slip surface for TC34 calculated by SSP

which was then fixed. This is discussed further in Section 7. After fixing the bug, the test was performed again. The interslice forces calculated by SSP for this example problem are shown in Figure 2. The average relative error between SSP and the original program was 0.099 for the normal forces and 0.072 for the shear forces. These relative errors were still higher than expected, but further investigation found that there is a high variance in these calculations, even in the original program. For example, the peak interslice normal force was seen to vary between about 140 kN and 200 kN for different runs of the original program. As a result, the tests were given a relatively high relative tolerance of 0.15 and were compared to outputs from multiple runs of the original program. As long as the relative error is less than the relative tolerance for at least one of the outputs, the tests pass. With these changes, both tests pass on the vast majority of runs.

The next test in the System VnV Plan, TC37, verified that changing the type of the function f to a constant has an impact on the result. This was done by visual inspection of the critical slip surface and interslice forces for the case where *const_f* is true. These plots are shown in Figure 3. The factor of safety was 0.8363 for this test. The results were similar overall to the case where *const_f* was false, with some notable differences: A slightly higher exit x-value for the slip surface, associated with notably different interslice forces near the exit, and an approximately 14 percent difference in factor of safety. These slight differences give confidence that SSP actually

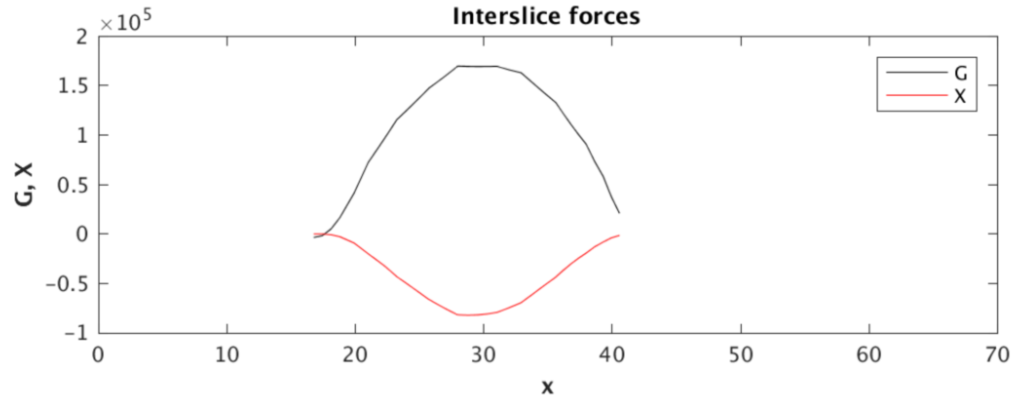

Figure 2: The interslice normal and shear forces for TC35 and TC36 calculated by SSP

translated the *const_f* input to a different f function, which is what this test was checking for, and thus the test was considered a pass.

The final functional requirement tests in the System VnV Plan, TC38 - TC49, were for verifying the requirements for verifying and delivering output. These tests all passed.

4 Nonfunctional Requirements Evaluation

4.1 Correctness

The success of the system tests for the calculations of SSP, specfically TC31 - TC36, gives confidence in the correctness of SSP.

4.2 Maintainability

TC50 in the System VnV Plan is a test for gaining confidence in the maintainability of SSP. The test question was asked to a volunteer test subject who was given the Module Guide (MG) for SSP as a resource to answer the question. The volunteer's response was the Input module, Morgenstern-Price calculation module, and potentially the Slice Property Calculation and Genetic Algorithm modules. This answer was correct with the exception

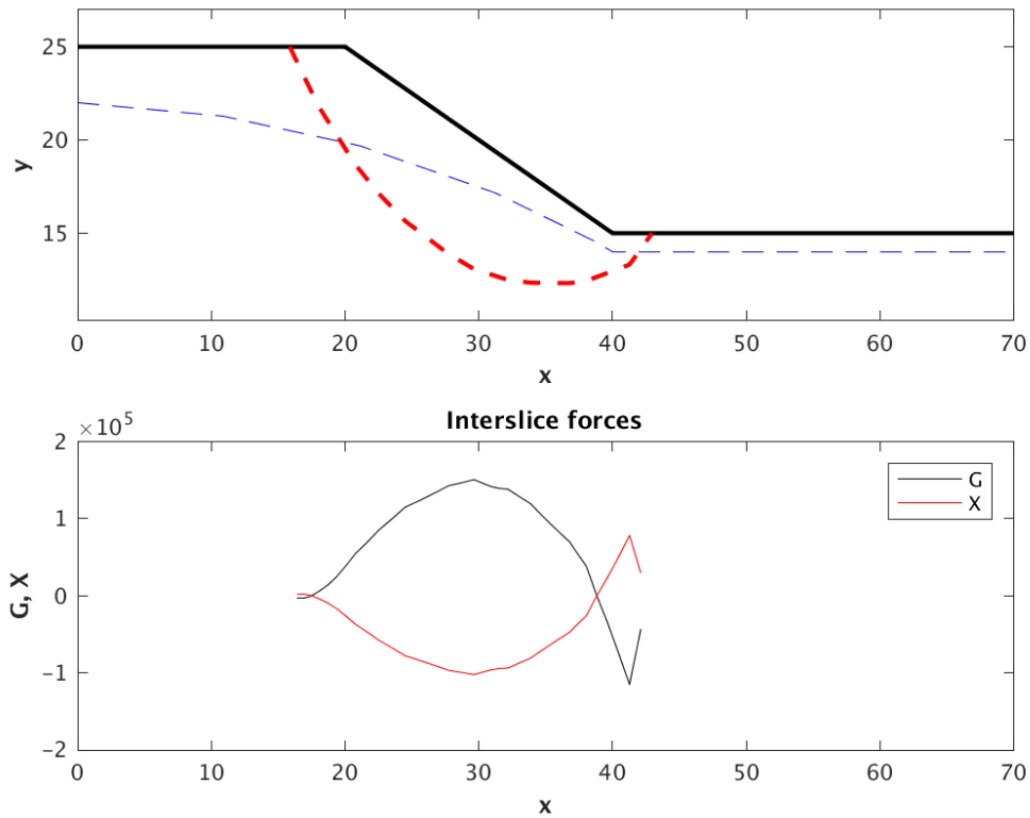

Figure 3: The critical slip surface (top) and interslice normal and shear forces (bottom) for TC37 calculated by SSP

of the Genetic Algorithm module. The volunteer's explanation for thinking this module might need to be changed was that, due to their lack of domain knowledge, they did not know if the changes to the Morgenstern-Price Calculation module would include changes to its interface. Since the Genetic Algorithm module uses the Morgenstern-Price Calculation module, it would need to be changed if the interface to the Morgenstern-Price Calculation module changed. This explanation was not only reasonable but also prudent, and thus the response was considered correct and this test result was considered a pass, giving confidence in the maintainability of SSP.

4.3 Reusability

TC51 in the System VnV Plan is a test for gaining confidence in the reusability of SSP. The alternative input module was written based off of the Input module from the previous implementation, which accepted some user inputs by keyboard input. This alternative Input Module is called "InputKeyboard.m" and can be found in the GitHub repository for this project. The current Input Module was temporarily replaced by this alternative Input Module. The system tests in "control_test.m" were then ran. The tests passed, showing that all of the other modules were reusable in this alternative version of SSP.

4.4 Understandability

Since maintainability and reusability are related to understandability, the success of the tests for those requirements gives confidence in the understandability of SSP.

5 Comparison to Existing Implementation

The results of TC33 - TC36 from the System VnV Plan showed how the current implementation provides similar results to the previous implementation, as discussed in Section 3. However, the current implementation improved upon the existing implementation in other regards. The existing implementation had tests, but they were implemented without a unit testing framework and did not work when the instructions for running them were followed. The tests also did not cover every module. The new implementation has working tests implemented with a unit testing framework, meaning they are easy to run and easy to add to in the future. Since the tests were developed in a systematic fashion based on the Module Interface Specification (MIS), they extensively cover all of the modules that are implemented by SSP. In addition, the tests uncovered an oversight in the original implementation, a case that was not accounted for in the code, which is discussed further in Sections 6 and 7.

6 Unit Testing

The Unit VnV Plan described TC1 - TC81, all of which were implemented in the unit testing framework and passed, with the exception of TC59, which failed on the first run, revealing that the Slip Slicing module had not been properly implemented for the case where the desired number of slices was indivisible by the starting number of slices. This prompted changes to the implementation, discussed further in Section 7. The test passed after these changes.

7 Changes Due to Testing

The failure of TC59 revealed that the implementation had not considered the case where the *soln.evnslc* variable (see MIS for more information) was false and the desired number of slices was not divisible by the initial number of slices. This prompted a change to the Slip Slicing module to properly handle this case.

The system test TC31 initially failed, revealing a bug in the case where there is no water table. In this case, the unit weight of water was set to 0 and thus violated the constraint that it must be greater than 0. The implementation was changed to only enforce this constraint when a water table exists.

The system tests TC35 and TC36 initially had very high relative errors, leading to the discovery of a bug where the interslice forces were not properly being mirrored in the case where expected soil motion is left-to-right. In this case, SSP is supposed to temporarily mirror the slope so that the direction of soil motion is right-to-left, and then mirror the slope and slip surface back to their original orientation before outputting the results. The interslice forces were not being mirrored back to the original implementation, leading to the addition of code to rectify this error.

8 Automated Testing

Every test, with the exception of TC37 and the tests for non-functional requirements, was automated so that the tests could be executed with a simple command. This was done by implementing the tests with MatLab's built-in unit testing framework. MatLab provides the "runtests" function for running all of the tests in a given directory.

9 Trace to Requirements

Tables 3 and 4 were taken directly from the System VnV Plan document for this project. It shows that the tests effectively cover every Requirement (R) and Non-Functional Requirement (NFR) from the SRS. Unless otherwise specified, the test cases referenced in the table refer to tests described in the System VnV Plan.

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11
UnitVnVPlan TC1 - TC25	X										
TC1 - TC30		X									
TC31 - TC34			X	X	X						
TC35									X		
TC36										X	
TC38 - TC44							X				
TC45								X			
TC46									X		
TC47										X	
TC48											X
TC49						X					

Table 3: Traceability matrix showing the connections between functional requirements and test cases

10 Trace to Modules

Table 5 was taken directly from the Unit VnV Plan document for this project, with the addition of one row for the System VnV Plan tests. It shows that the tests effectively cover every Module (M) from the MG and MIS, with the exception of those not implemented by SSP. Unless otherwise specified, the

	NFR1	NFR2	NFR3	NFR4
TC31 - TC36	X			
TC50		X		X
TC51		X	X	

Table 4: Traceability matrix showing the connections between non-functional requirements and test cases

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13
System VnV Plan		X											
TC1 - TC27			X										
TC28 - TC34				X									
TC35 - TC47					X								
TC48 - TC53						X							
TC55 - TC58							X						
TC60 - TC65								X					
TC66 - TC78									X				
TC79 - TC81										X			

Table 5: Traceability matrix showing the connections between modules and test cases

test cases referenced in the table refer to tests described in the Unit VnV Plan.

11 Code Coverage Metrics

Not applicable for SSP.

References

Y. M. Cheng, Liang Li, Shi chun Chi, and W. B. Wei. Particle swarm optimization algorithm for the location of the critical non-circular failure surface in two-dimensional slope stability analysis. *Computers and Geotechnics*, (24):92–103, 27 Decemeber 2006.

Venanzio R. Greco. Efficient monte carlo technique for locating critical slip surface. *J. Geotech. Engrg.*, (122):517–525, 1 July 1996.

Yu-Chao Li, Yun-Min Chen, Tony L.T Zhan, Dao-Sheng Ling, and Peter John Cleall. An efficient approach for locating the critical slip surface in slope stability analyses using a real-coded genetic algorithm. *Can. Geotech. J.*, (47):806–820, 25 June 2010.

Abdallah I. Husein Malkawi, Waleed F. Hassan, and Sarada K. Sarma. Global search method for locating general slip surface using monte carlo techniques. *J. Geotech. Geoenviron. Eng.*, (127):688–698, 1 August 2001.