

System Verification and Validation Test
Report: A Library of Simplex Method Solvers
(LoSMS)

Hanane Zlitni

December 18, 2018

1 Revision History

Date		Version	Notes
December 2018	15,	1.0	First draft

2 Symbols, Abbreviations and Acronyms

The following symbols, abbreviations and acronyms were used in this document:

symbol	description
LoSMS	Library of Simplex Method Solvers
T	Test
R	Requirements
NFR	Nonfunctional Requirements
V&V	Verification and Validation
<i>s. t.</i>	Subject to
Z	Optimal solution of the objective function
x_1, x_2, x_3	Decision variables
$\epsilon_{subscript}$	The relative error of the subscript
s	seconds

Contents

1	Revision History	i
2	Symbols, Abbreviations and Acronyms	ii
3	Functional Requirements Evaluation	1
4	Nonfunctional Requirements Evaluation	2
4.1	Usability	2
4.2	Portability	2
4.3	Accuracy	3
4.4	Correctness	3
4.5	Performance	4
4.6	Stability	4
5	Comparison to Existing Implementation	5
6	Unit Testing	5
7	Changes Due to Testing	5
8	Automated Testing	5
9	Trace to Requirements	6
10	Trace to Modules	6
11	Code Coverage Metrics	7
12	Appendix	9
12.1	Usability Survey Questions	9
12.1.1	Participant 1: Soumaya Zlitni	9

List of Tables

1	Relative Errors of Expected Outputs	3
2	Output Correctness for LoSMS and MatLab	4
3	Elapsed Time for Executions of LoSMS and MatLab	4

4	Traceability Between Test Cases and Requirements	6
5	Traceability Between Test Cases and Modules	7

This document reports all results obtained from testing the Library of Simplex Method Solvers (LoSMS) tool. The test cases under evaluation, along with the full documentation of the library, can be found at: <https://github.com/hananezlitni/HZ-CAS741-Project>

The document starts by outlining the results of the test cases related to the tool's functional requirements in Section 3. Then, the results of the test cases related to the non-functional requirements are reported in Section 4. This is followed by Sections 5, 6 and 7, which discuss comparing any previous implementations with the existing one, unit testing of the library and changes which occurred after testing, respectively. Section 8 discusses the tests that were executed automatically, while sections 9 and 10 provide traceability tables for the tests with the requirements and modules. Section 11 concludes the document by discussing the metrics used to achieve code coverage.

3 Functional Requirements Evaluation

T1 & T2 in the System V&V Plan verified the cases where a maximization linear program has and doesn't have an optimal solution. Both test cases have passed.

T3 & T4 in the System V&V Plan verified the cases where a minimization linear program has and doesn't have an optimal solution. Both test cases have passed.

T5 & T6 in the System V&V Plan verified the cases where the objective function and linear constraints, respectively, are missing. Both test cases have passed.

The System V&V Plan can be found at: <https://github.com/hananezlitni/HZ-CAS741-Project/blob/master/docs/VnVPlan/SystVnVPlan/SystVnVPlan.pdf>.

4 Nonfunctional Requirements Evaluation

4.1 Usability

A usability study was conducted to check whether LoSMS user-friendly or not (T7 in the System V&V Plan).

The participant, Soumaya Zlitni, was given the linear program to solve after reading the information provided in the project repository's README.md file (found at: <https://github.com/hananezlitni/HZ-CAS741-Project/blob/master/src/README.—HZ>). At the end of the usability testing, Soumaya filled in the usability survey that was handed to her. The filled survey can be found in the appendix (Section 12.1).

There might have been some bias in the usability testing since the participant is a relative.

4.2 Portability

T8 in the System V&V Plan specified the 3 operating systems that LoSMS should be operable on: Windows, MacOS and Linux.

The tool was ran twice in each operating system: once with the linear program in T1 in the System V&V Plan and once with the same linear program but with a missing input (the linear program goal). This was done to test the behaviour of the library in the correct and incorrect conditions in each operating system.

Running the library on Windows and Linux specified the Python libraries that the tool depends on to execute without errors. The details of the dependencies and how to download them are explained in the README.md file in the project's GitHub repository, found at: <https://github.com/hananezlitni/HZ-CAS741-Project/blob/master/src/README.md>. The portability test passed for all 3 operating systems by printing the correct optimum in the first execution and the missing input exception message in the second execution.

4.3 Accuracy

T9 in the System V&V Plan tests the accuracy of the solutions produced by LoSMS.

The following table reports the relative errors of the expected outputs for each test case detailed in the System V&V Plan that produces numbers.

T	Expected	Actual	Relative Error
T1	$Z = 3$	$Z = 3.00$	$\epsilon_Z = 0$
	$x_1 = 0$	$x_1 = 0$	$\epsilon_{x_1} = 0$
	$x_2 = 0$	$x_2 = 0$	$\epsilon_{x_2} = 0$
	$x_3 = 3$	$x_3 = 3.0$	$\epsilon_{x_3} = 0$
T3	$Z = -4.57$	$Z = -4.5714$	$\epsilon_Z = 0.0003$
	$x_1 = 2.29$	$x_1 = 2.2857$	$\epsilon_{x_1} = 0.00188$
	$x_2 = 0$	$x_2 = 0$	$\epsilon_{x_2} = 0$

Table 1: Relative Errors of Expected Outputs

4.4 Correctness

T10 in the System V&V Plan tests the correctness of the library. The test was done by solving the same linear program using LoSMS and MatLab, then comparing the output of both programs.

The following table reports the result:

Problem	LoSMS	MatLab
$\min Z = -x_1 - 2x_2$	$Z = -10.00$	$Z = -10.00$
$s. t. \quad 2x_1 + x_2 = 10$	$x_1 = 2.0$	$x_1 = 2.00$
$x_1 + x_2 = 6$	$x_2 = 4.0$	$x_2 = 2.00$
$-x_1 + x_2 = 2$		
$-2x_1 + x_2 = 1$		

Table 2: Output Correctness for LoSMS and MatLab

4.5 Performance

T11 in the System V&V Plan tests the performance of the library. The test was done by solving the same linear program using LoSMS and MatLab, then comparing the elapsed time (*time at the start of the execution - time at the end of the execution*) for both programs.

The following table reports the result:

Problem	LoSMS	MatLab
$\min Z = -x_1 - 2x_2$	0.000549 s	0.054326 s
$s. t. \quad 2x_1 + x_2 = 10$		
$x_1 + x_2 = 6$		
$-x_1 + x_2 = 2$		
$-2x_1 + x_2 = 1$		

Table 3: Elapsed Time for Executions of LoSMS and MatLab

4.6 Stability

Stress testing using a third-party tool was planned for LoSMS to verify its behaviour under heavy load. However, due to not finding a tool that automatically loads a very large number of inputs to the library, the other tests

were prioritized and this one was moved to a second phase of testing using other methods.

5 Comparison to Existing Implementation

This section is not applicable for LoSMS.

6 Unit Testing

The unit testing of the LoSMS tool is discussed in details in the Unit V&V Report found at: <https://github.com/hananezlitni/HZ-CAS741-Project/tree/master/docs/VnVReport/UnitVnVReport/UnitVnVReport.pdf>.

7 Changes Due to Testing

Initially, The output of LoSMS was a string consisting of the optimal solution and the points where it occurs. The string had to be changed to a *dictionary* (see <https://docs.python.org/3/tutorial/datastructures.html#dictionaries>) because the assert function in Unittest compared dictionaries and not strings.

The test cases for missing inputs in the System V&V Plan initially failed because there was a bug in the condition that checked for missing inputs and was producing an incorrect output. Upon modifying the condition, the correct output was produced and the tests have passed.

8 Automated Testing

All the test cases for functional requirements were automated using Unittest provided by Python. The implemented tests are executed by entering a command in the command line tool.

9 Trace to Requirements

Table 4 shows the traceability between the test cases in the System V&V Plan and the requirements.

T	Requirements
T1	R1, R4, R5, R6
T2	R1, R4, R5, R6
T3	R1, R4, R5, R6
T4	R1, R4, R5, R6
T5	R1, R2, R3
T6	R1, R2, R3
T7	NFR1
T8	NFR2
T9	NFR3
T10	NFR4
T11	NFR5
T12	NFR6

Table 4: Traceability Between Test Cases and Requirements

10 Trace to Modules

Table 5 shows the traceability between the test cases and the modules.

T	Modules
T1	Simplex Method Solver Module
T2	Simplex Method Solver Module, Exceptions Module
T3	Simplex Method Solver Module
T4	Simplex Method Solver Module, Exceptions Module
T5	Simplex Method Solver Module
T6	Simplex Method Solver Module
T7	Simplex Method Solver Module
T8	Simplex Method Solver Module
T9	Simplex Method Solver Module
T10	Simplex Method Solver Module
T11	Simplex Method Solver Module
T12	Simplex Method Solver Module

Table 5: Traceability Between Test Cases and Modules

11 Code Coverage Metrics

This section is discussed in details in the Unit V&V Report found at:
<https://github.com/hananezlitni/HZ-CAS741-Project/tree/master/docs/VnVReport/UnitVnVReport/UnitVnVReport.pdf>.

References

12 Appendix

This section provides additional content related to this document.

12.1 Usability Survey Questions

12.1.1 Participant 1: Soumaya Zlitni

1. Did LoSMS successfully provide all the services you requested?

(**Yes** / No)

Why have you chosen the above response? It was clear what the input needed to be.

2. How confident are you that LoSMS provided you with the correct results?

(1 / 2 / 3 / 4 / **5**) ; (1) *being not confident at all* and (5) *being very confident*

Why have you given the above score? Because everything was clear and left little room for any confusion.

3. How satisfied are you with the library's response time?

(1 / 2 / 3 / 4 / **5**) ; (1) *being not satisfied at all* and (5) *being very satisfied*

Why have you given the above score? The tool was very fast in giving me the output.

4. How likely are you to recommend LoSMS to a friend?

(1 / 2 / 3 / 4 / **5**) ; (1) *being very unlikely* and (5) *being very likely*

Why have you given the above score? The tool does what it is supposed to do.

5. Rate your overall satisfaction with LoSMS out of 10

(1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / **9** / 10)

Why have you given the above score? The tool did what it is supposed to do in a very short amount of time. The inputs that should be entered were clear.