System Verification and Validation Plan for Truss Tool

Maryam Valian

February 26, 2023

1 Revision History

Date	Version	Notes
13-02-2023	1.0	First Draft of VnV plan
14-02-2023	1.1	Updating Test cases
16-02-2023	1.2	Updating Tests for NFRs
25-02-2023	1.3	Updating According to first reviewer feedback
26-02-2023	1.4	Updating According to second reviewer feedback

Contents

2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test
SRS	Software Requirements Specification
R	Functional Requirements
NFR	Nonfunctional Requirements
N/A	Not Applicable
VnV	Verification and Validation

This document provides an overview of the Verification and Validation plan for Truss Tool. The general information is introduced in section 3. The verification plans and the test descriptions are explained in section 4 and section 5, respectively.

3 General Information

3.1 Summary

In this document, we explain how Truss Tool is being tested. Truss Tool is software for truss analysis. Users can define a specific truss, by giving the exact location of the joints, members, supports and external forces. Truss Tool calculates the reaction of the supports and internal forces.

3.2 Objectives

Our objective is to build confidence in the software's correctness and increase its reliability. The functional requirements will be tested according to section 5.1, and Non-functional requirements will be tested according to section 5.2.

3.3 Relevant Documentation

- Problem Statement
- SRS
- VnV report

4 Plan

In This section, we explain the VnV plan of the Truss Tool. The first section will introduce VnV team members and their roles in the project's verification. Next, the verification plans of SRS, design, VnV verification plan, implementation, Automated testing, and software validation plan will be given.

4.1 Verification and Validation Team

This section lists VnV team members and describes the role of each member in the project's verification.

- Maryam Valian reviews the whole project as the author.
- Dr. Spencer Smith reviews the whole project as supervisor.
- Lesley Wheat reviews the whole project as a domain expert.
- Joachim de Fourestier reviews the VnV plan as a secondary reviewer.
- Jason Balaci reviews the MIS as a secondary reviewer.

4.2 SRS Verification Plan

The SRS will be reviewed by Dr. Spencer Smith and Lesely Wheat. Reviewers can give feedback and revision suggestions to the author by creating issues on GitHub. It is the author's responsibility to check the submitted issues regularly and make necessary modifications.

4.3 Design Verification Plan

The design document MIS will be reviewed by Dr. Spencer Smith, Lesley Wheat and Jason Balaci. Reviewers can give feedback and revision suggestions to the author by creating issues on GitHub. It is the author's responsibility to check the submitted issues regularly and make necessary modifications.

4.4 Verification and Validation Plan Verification Plan

The VnV plan document will be reviewed by Dr. Spencer Smith, Lesley Wheat and Joachim de Fourestier. Reviewers can give feedback and revision suggestions to the author by creating issues on GitHub. It is the author's responsibility to check the submitted issues regularly and make necessary modifications.

4.5 Implementation Verification Plan

The implementation of the software will be verified using several techniques involving manual or automated interactions. The software will be developed in Python programming language. The code evaluation will involve unit and system testing. The tools for the evaluations will be mentioned in section 4.6. Furthermore, the test cases used in system and unit testing will be stated in sections 5 and 6, respectively.

4.6 Automated Testing and Verification Tools

Truss Tool will be tested and verified by Pytest as an automated tool for unit testing and system testing. Automated testing will be implemented for individual units as well as the integrated system.

4.7 Software Validation Plan

Validation is the process of comparing the outputs of models to experimental values. The validation of the Truss Tool will be done by comparing the outputs of the software to several outputs from an online tool called 2D-Truss Analysis developed by Valdivia. The Euclidean distance between both results from the two software will be reported as an error measurement for some cases. The euclidean distance formula will be calculated as follows:

$$d\left(p,q\right)=\sqrt{\sum_{i=1}^{n}\left(q_i-p_i\right)^2}$$

Where p is the vector of internal forces from the Truss Tool and q is the vector of internal forces from the 2D-Truss Analysis for a specific test case.

5 System Test Description

5.1 Tests for Functional Requirements

The functional requirements are described in the SRS. Truss Tool will detect invalid inputs and assure that the calculated outputs are correct. Testing R1 and R2 will be explained in the input verification section. R3 and R4 will be explained in the output verification test.

TC	N	J,M	F	$\{S(p,r)\}$	**Output**
1-1	3	$J = \{(0,0),(1,1),(2,0)\}$ $M = \{(1,2),(2,3),(1,3)\}$	$\{(2,-50)\}$	(1,3)	(25,25) (-35,-35,25)
2-1	10	$J = \{(0,0),..,(2,0)\}$ $M = \{(1,2),..,(1,3)\}$	$\{(2,-50)\}$	(1,3)	Error
2-2	2	$J = \{(0,0),(2,0)\}$ $M = \{(1,2)\}$	$\{(2,-50)\}$	(1,2)	Error
3-1	3	$J = \{(0,0),(1,1),(2,0)\}$ $M = \{(1,2),(2,3),(1,3)\}$	$\{(2,-36000)\}$	(1,3)	Error
3-2	3	$J = \{(0,0),(1,1),(2,0)\}$ $M = \{(1,2),(2,3),(1,3)\}$	$\{(2,36000)\}$	(1,3)	Error
4-1	3	$J = \{(0,0),(1,1),(2,0)\}$ $M = \{(1,2),(2,3),(1,3)\}$	$\{(2,-50)\}$	(0,0)	Error
4-2	3	$J = \{(0,0),(1,1),(2,0)\}$ $M = \{(4,5),(9,3),(1,3)\}$	$\{(2,-50)\}$	(1,3)	Error
4-3	3	$J = \{(0,0),(1,1),(2,0)\}$ $M = \{(1,2),(2,3),(1,3)\}$	$\{(4,-50)\}$	(1,3)	Error
4-4	3	$J = \{(0,0),(1,1),(2,0)\}$ $M = \{(1,2),(2,3),(1,3)\}$	$\{(2,-50)\}$	(4,5)	Error

Table 1: Test cases For input Verification.

5.1.1 Input Verification

The inputs will be tested to satisfy R1 and R2 from Truss Tool SRS. Specifically, this test will ensure values of the inputs align with the input constraints. Table 1 displays the inputs and outputs of test cases for the input constraints tests.

Input Verification test

1. T-1: Expected Input

   Control: Automatic

Initial State: N/A

Input: TC1-1

Output: Support Reactions (25,25), Internal forces (-35,-35,25)

Test Case Derivation: Derived from existing software 2D-Truss Analysis.

How the test will be performed: It will be performed by test classes built with the help of Pytest

2. T-2: Invalid Number of joints N

   Control: Automatic

   Initial State: N/A

   Input: TC2-1, TC2-2

   Output: Exception: InputError

   Test Case Derivation: N input is out of bound for 10 and 2.

   How the test will be performed: It will be performed by test classes built with the help of Pytest.

3. T-3: Invalid amount of external force N

   Control: Automatic

   Initial State: N/A

   Input: TC3-1, TC3-2

   Output: Exception: InputError

   Test Case Derivation: Amount of the external force/forces out of bound.

   How the test will be performed: It will be performed by test classes built with the help of Pytest.

4. T-4: Invalid index of joints N

   Control: Automatic

   Initial State: N/A

Input: TC4-1 to TC4-4

Output: Exception: InputError

Test Case Derivation: Any reference to joints must be in the range of 1 to N. The support Index can be zero in case of no existing support, but both indices of support cannot be zero at the same time.

How the test will be performed: It will be performed by test classes built with the help of Pytest.

5.1.2 Output verification

The output will be tested as follows:

Output Verification test

1. T-5: Zero summation for the last joint

   Control: Automatic

   Initial State: All internal forces are already calculated.

   Input: Internal forces array IF

   Output: Exception: Output Error

   How the test will be performed: It will be performed by Truss Tool. The summation of internal forces in the last joint will be zero. Otherwise, it will generate an exception message.

2. T-6: Pseudo-oracle test
   Control: Manual

   Initial State: Both Truss Tool and 2D-Truss Analyse are running.

   Input: The result of internal forces and support reaction from both tools

   Output: The euclidean distance between results.

   How the test will be performed: It will be performed by the Truss Tool team to verify results of the software is similar to the previous software introduced in section 4.7.

Test cases	N	**input**
TC5-1	3	$J = \{(0,0),((1,1),(2,0)\}$ $M = \{(1,2),((1,3),(2,3)\}$ $F = \{(2,-50)\}$,$S = (2,0)$
TC5-2	4	$J = \{(0,0),((1,1),(2,0),(3,1)\}$ $M = \{(1,2),((1,3),(2,3),(2,4),(3,4)\}$
TC5-3	5	$J = \{(0,0),((1,1),(2,0),(3,1),(4,0)\}$ $M = \{(1,2),((1,3),(2,3),(2,4),(3,4),(3,5),(4,5)\}$
TC5-4	6	$J = \{(0,0),((1,1),(2,0),(3,1),(4,0),(5,1)\}$ $M = \{(1,2),((1,3),(2,3),(2,4),(3,4),(3,5),(4,5)(4,6),(5,6)\}$
TC5-5	7	$J = \{(0,0),((1,1),(2,0),(3,1),(4,0),(5,1),(6,0)\}$ $M = \{(1,2),((1,3),(2,3),(2,4),(3,4),(3,5),(4,5)(4,6),$ $(5,6),(5,7),(6,7)\}$
TC5-6	8	$J = \{(0,0),((1,1),(2,0),(3,1),(4,0),(5,1),(6,0),$ $(7,1)\}$ $M = \{(1,2),((1,3),(2,3),(2,4),(3,4),(3,5),(4,5)(4,6),$ $(5,6),(5,7),(6,7),(6,8),(7,8)\}$
TC5-7	9	$J = \{(0,0),((1,1),(2,0),(3,1),(4,0),(5,1),(6,0),$ $(7,1),(8,0)\}$ $M = \{(1,2),((1,3),(2,3),(2,4),(3,4),(3,5),(4,5)(4,6),$ $(5,6),(5,7),(6,7),(6,8),(7,8),(7,9),(8,9)\}$

Table 2: Test cases for time performance.

5.2 Tests for Nonfunctional Requirements

This section will define the tests to ensure Truss Tool satisfies the nonfunctional requirements seen in the SRS document of Truss Tool.

5.2.1 Reliability

Performance measurement

1. T-7: Time performance

Type: Manual

Initial State: Truss Tool is running. Input/Condition: The user has entered the inputs correctly.

Output/Result: Time duration for calculating all the internal forces.

How the test will be performed: run-time of test cases from table 2 will be stored as a pair of run-time and the number of Joints. The VnV team members will analyze the log and report the results.

5.3 Traceability Between Test Cases and Requirements

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the column of that component that is marked with an X may have to be modified as well. Table 4 shows which test cases are supporting which requirements.

Test Cases	R1	R2	R3	R4	NFR1
T-1	X	X			
T-2	X	X			
T-3	X	X			
T-4	X	X			
T-5				X	
T-6			X	X	
T-7					X

Table 3: Traceability Matrix between tests and requirements.

6 Unit Test Description

This section should not be filled in until after the MIS (detailed design document) has been completed.

6.1 Unit Testing Scope

In this project, we will use SymPy library of Python to solve our equations. Testing this library is not in the scope of this document.

6.2 Tests for Functional Requirements

6.2.1 Module 1

1. test-id1

   Type: Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic

   Initial State:

   Input:

   Output: The expected result for the given inputs

   Test Case Derivation: Justify the expected value given in the Output field

   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic

   Initial State:

   Input:

   Output: The expected result for the given inputs

   Test Case Derivation: Justify the expected value given in the Output field

   How test will be performed:

6.3 Tests for Nonfunctional Requirements

6.3.1 Module ?

1. test-id1

   Type: Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic

   Initial State:

   Input/Condition:

   Output/Result:

   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

   How test will be performed:

6.3.2 Module ?

...

6.4 Traceability Between Test Cases and Modules

References

Anton Valdivia. 2d-truss analysis. URL https://valdivia.staff.jade-hs.de/fachwerk_en.html.

7 Appendix

Not applicable.

7.1 Symbolic Parameters

Symbol	Meaning
J	Tuples of the joint's coordination. Each tuple is a pair of two-dimensional geometric coordination.
M	Tuples of members. Each tuple is pair of joint index
F	External forces. Tuples of the vertical and horizontal amount of force for each joint
$S = (S_p, S_R)$	Pair of two indices. Index of pinned support and index of roller support

Table 4: Traceability Matrix between tests and requirements.

7.2 Usability Survey Questions?

Not Applicable.

Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning. Please answer the following questions:

Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning. Please answer the following questions:

1. What knowledge and skills will the team collectively need to acquire to successfully complete the verification and validation of your project? Examples of possible knowledge and skills include dynamic testing knowledge, static testing knowledge, specific tool usage etc. You should look to identify at least one item for each team member.

2. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?