Verification and Validation Report: Truss Tool

Maryam Valian

April 19, 2023

1 Revision History

Date	Version	Notes
15/04/2023	1.0	Draft version
18/04/2023	1.1	Update report
19/04/2023	1.2	Update figures and tables
19/04/2023	1.3	Update code coverage

2 Symbols, Abbreviations and Acronyms

symbol	**description**
T	Test

For more detailed Table of Symbols please see SRS

Contents

List of Tables

List of Figures

This document is a report on the results of a testing plan for Truss Tool. Detailed descriptions of the tests can be found in VnV Plan

3 Functional Requirements Evaluation

All functional requirements have been met.

4 Nonfunctional Requirements Evaluation

4.1 Reliability

The outputs generated by Truss Tool were compared to the Existing software with the same input data. The results were the same and the mean error between the expected value and generated value was less than 0.1.

4.2 Portability

Truss Tool runs successfully on Windows. The test was done manually.

5 Comparison to Existing Implementation

The outputs generated by Truss Tool were compared to the Existing software. the comparison was based on measuring mean error.

6 Unit Testing

The detail of the unit tests can be found in VnV Plan section. There were 12 Tests designed and all test cases are performed by test classes built with the help of Pytest. All tests succeed.

7 Changes Due to Testing

For the first time running the output test failed. The results didn't completely match the expected value. For some of the members, the results were correct and for some others not. We found the bug in the code was that we did not consider that the force vector's direction is different in two end joints of a member. Figure 1 shows the difference. So at the first implementation, we assumed that for each member the direction of force is fixed. After code inspection and debugging, we changed the code to consider a different direction for the member force based on that for which joint we are writing the equations.

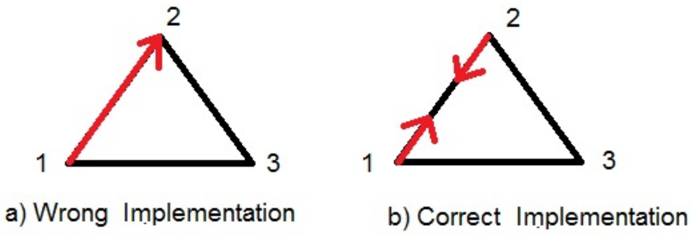

Figure 1: Changes Due To Testing

8 Automated Testing

Tools used for automated testing are mentioned in VnV Plan section 4.5. All automated tests were passed successfully.

9 Trace to Requirements

We provide traceability matrices to make it easy for referencing what has to be additionally modified if a certain component is changed. Table 1 shows the dependencies between the test cases and the requirements.

10 Trace to Modules

The traceability between test cases and modules are provided in section 6.4 of VnV Plan

Requirements	**Test section**
R1	section 3
R2	section 3
R3	section 3
R4	section 3
NFR1	section 4
NFR2	section 4

Table 1: Traceability Between Test Cases and Requirements

11 Code Coverage Metrics

Code coverage is a white-box testing technique performed to verify the extent to which the code has been executed. The following results are provided by using the Coverage library of Python:

Modules	**Code coverage**
Inputs.py	71%
Internals.py	87%
Outputs.py	79%
Total	**79%**

Table 2: Results of Code coverage