Software Requirements Specification for Truss Tool: A Tool for Truss Analysis

Maryam Valian

February 3, 2023

Contents

Revision History

Date	Version	Notes
2023-01-30	1.0	Initial version of the SRS
Date 2	1.1	Notes

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

Throughout this document, SI (Système International d'Unités) is employed as the unit system. In addition to the basic units, several derived units are used as described below. For each unit, the symbol is given followed by a unit description and the SI name.

symbol	unit	SI
m	length	metre
N	force	newton
deg	angle	degree

1.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The choice of symbols was made to be consistent with the structural statics literature and with existing documentation for the truss analysis problem. The symbols are listed in alphabetical order.

symbol	unit	description
F_{i}	N	External force of joint i
M_{i}	Nm	Moment component of joint i
θ	deg	Angle between two members
F_{xi}	N	Force component in the x direction of joint i
F_{yi}	N	Force component in the y direction of joint i
S_{p}	-	Pin support
S_{r}	-	Roller support

1.3 Abbreviations and Acronyms

symbol	description
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
T	Theoretical Model

1.4 Mathematical Notation

In this document, we do not use any specific mathematical notation.

2 Introduction

A truss is a structure that consists of members organized into connected triangles to enable the distribution of loads and forces. Trusses are most commonly used for wide spans like bridges, and roofs. Truss Analysis shows whether the external forces are well-distributed among the members or not.
The following section provides an overview of the software requirements specification (SRS) for the truss tool. In this section, first, we explain the purpose of the document. Then we explain the scope of the requirements, the characteristics of the intended reader and the organization of the document.

2.1 Purpose of Document

The primary purpose of this document is to outline the software requirements of the truss analysis tool. To provide a good understanding of the system, different aspects of the system such as goals, assumptions, theoretical models, and definitions will be explained. The following SRS document will remain abstract exploring what is being solved rather than how it will be solved.
The following document will describe the system context and constraints, the specific problem definition and solution characteristics, requirements and likely and unlikely changes for the development of the tool.

2.2 Scope of Requirements

The scope of the requirements includes the analysis of the two-dimensional trusses where all members and nodes lie within a two-dimensional plane. for more details, you can also see the assumptions section (Section 4.2.1).

2.3 Characteristics of Intended Reader

Reviewers of this documentation should have a basic understanding of structure statics and high school physics and Mathematics. The users of the Truss Tool must have a higher level of expertise, as explained in Section: User Characteristics (Section 3.2).

2.4 Organization of Document

The organization of the document follows the template for an SRS for scientific computing software proposed by (**?**). The template will present the system's goals, theories, definitions, and assumptions. Readers interested in top-down reading can begin by reading the system's goal statements (Section 4.1.3). Subsequently, the theoretical models will elaborate on the goal statements. Lastly, readers can finish with a more understanding of the system by reading instance models of the system.

3 General System Description

This section provides general information about the system. It identifies the interfaces between the system and its environment, describes the user characteristics and lists the system constraints.

3.1 System Context

Figure 1 shows the system context. The circles represent a user that interacts with the software. The rectangle represents the software system for the truss tool. The arrows display the input data from the user and the output data that is useful for the user.

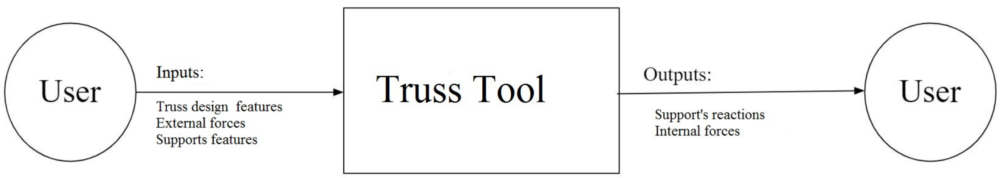

Figure 1: System Context

The interaction between the product and the user is through a user interface. The responsibilities of the user and the system are as follows:

- User Responsibilities:
  - Provide truss design features, supports and external forces.
  - Ensure the input data describes a correct truss.
- Truss Tool Responsibilities:
  - Detect data type mismatch, such as a string of characters instead of a floating point number.
  - Calculate external forces and support's reaction.

3.2 User Characteristics

The end user of Truss Tool should be an architecture/civil/mechanic engineer or should have an understanding of undergraduate Level 1 structural analysis.

3.3 System Constraints

There is no constraint on the development of the Truss Tool.

4 Specific System Description

This section first presents the problem description, which gives a high-level view of the problem to be solved. This is followed by the solution characteristics specification, which presents the assumptions, theories, definitions and finally the instance models.

4.1 Problem Description

Truss Tool is intended to solve a given truss with given external forces. By solving a truss, we mean that we are interested to calculate all internal forces among the members and the reactions of the supports. As a result, Truss Tool will help engineers to make a decision on whether the design of the given truss is proper or not.

4.1.1 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

- Planar truss: A planar truss is one where all members and nodes lie within a two-dimensional plane.
- Joint(nodes): A place where two or more members of the truss are connected.
- Force equilibrium: A body is in force equilibrium if the sum of all the forces acting on the body is zero.
- Moment equilibrium: A body is in moment equilibrium if the sum of all the moments of all the forces acting on the body is zero.
- Moment: The turning effect of a force is called the moment. The moment is the result of the force multiplied by the perpendicular distance from the line of action of the force to the pivot or point where the object will turn.
- Compression: When a member force points toward the joint it is attached to, the member is in compression
- Tension: When a member force points away from the joint it is attached to, the member is in tension.
- Pinned support: A kind of structural support that can have both a horizontal reaction and a vertical reaction.
- Roller support: A kind of structural support that can have only a vertical reaction.
- Zero force members: Members which do not have any force in them.

4.1.2 Physical System Description

The physical system of Truss Tool, as shown in Figure 2, includes the following elements:

- PS1: The joints (j_1,j_2,..,j_n).
- PS2: The members (m_1,m_2,..,m_k).
- PS3: The supports (S_1,S_2).

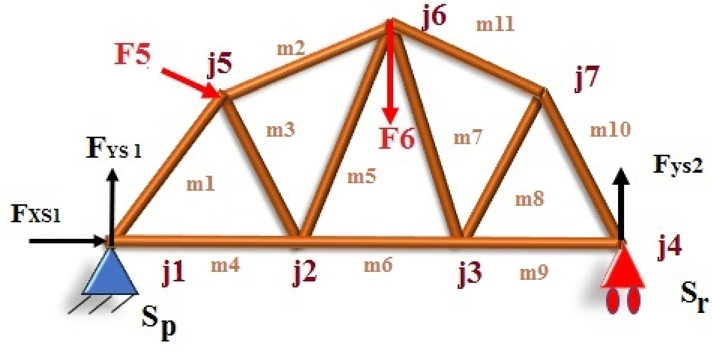

Figure 2: The physical system of Truss Tool

4.1.3 Goal Statements

Given the truss features and external forces, the goal statements are:

- GS1: Calculate the reactions of the supports.
- GS2: Calculate the internal forces for each member.

4.2 Solution Characteristics Specification

The instance models that govern Truss Tool are presented in Subsection 4.2.6. The information to understand the meaning of the instance models and their derivation is also presented so that the instance models can be verified.

4.2.1 Assumptions

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the theoretical model [T], the general definition [GD], data definition [DD], instance model [IM], or likely change [LC], in which the respective assumption is used.

- A1: All members and nodes lie within a two-dimensional plane.
- A2: Members are inter-connected only at their ends.
- A3: Members must be straight.
- A4: All joints are smooth and frictionless hinges.
- A5: All forces must only be applied at joints
- A6: All reactions must only be applied at joints
- A7: Self-weight of the member will be neglected
- A8: Members are subjected to axial forces only.
- A9: The number of supports at most is two.

4.2.2 Theoretical Models

This section focuses on the general equations and laws that Truss Tool is based on.

RefName: **T: EQUIL**

Label: Equilibrium Equations

Equation: $\sum F = 0$, $\sum M = 0$

Description: The equilibrium equation describes the static equilibrium of all forces of the system and the moment for the system so that $\sum M = 0$ and $\sum F = 0$.
F is any force in the system (N). M is a moment that is the turning effect of a force. Moments act about a point in a clockwise or anticlockwise direction(N m)

Notes: None.

Source: https://en.wikipedia.org/wiki/Mechanical_equilibrium/

Ref. By: GD??

Preconditions for T: EQUIL: None

Derivation for T: EQUIL: Not Applicable

4.2.3 General Definitions

This section collects the laws and equations that will be used in building the instance models.

Number	GD1
Label	**Equilibrium equations in planar trusses**
SI Units	All Forces are measured in N
Equation	$\sum F_x = 0$, $\sum F_y = 0$
Description	For any joint point in a planar truss, the equilibrium equations are satisfied horizontally and vertically on the direction of y-axis and y-axis.
Source	https://www.khanacademy.org/math/geometry/hs-geo-analytic-geometry/hs-geo-distance-and-midpoints/a/distance-formula
Ref. By	DD??

Detailed derivation of the member length

4.2.4 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given.

Number	DD1
Label	**Length of a straight Line**
Symbol	L
SI Units	m
Equation	$L = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
Description	For every two points such as X_1,X_2 with coordination (x_1, y_1) and (x_2, y_2) the length of line between two point is L.
Sources	https://www.cuemath.com/distance-formula/
Ref. By	IM1

Number	DD2
Label	**Finding angle by Law of cosine**
Symbol	θ
SI Units	Degree
Equation	$\theta = \arccos(\frac{a^2+b^2+c^2}{2ab})$
Description	The Law of Cosine helps us to find any angle for a given triangle with a known length of sides. Where θ is the angle between sides a and b. and c is the length of the opposite side.
Sources	https://en.wikipedia.org/wiki/Law_of_cosines
Ref. By	IM1

4.2.5 Data Types

This section collects and defines all the data types needed to document the models. For Truss Tool, all data types are real numbers or boolean numbers.

4.2.6 Instance Models

This section transforms the problem defined in Section 4.1 into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 4.2.4 to replace the abstract symbols in the models identified in Sections 4.2.2 and 4.2.3.

The goals G1 and G2 are solved by IM1, IM2, IM3.

Number	IM1
Label	**Find truss design features L_j, θ_i**
Input	J_i,M_j, S_k, F_i The input is constrained so that $k \leq 2$ (A9)
Output	$L(m_j)$ so that $0 \leq L$, θ between each two members
Description	$L(m_j)$ is the length of the member j. for each member Truss Tool should calculate the length from DD1. θ is the angle between two members that will be calculated from DD2.
Sources	Citation here
Ref. By	IM1

Number	IM2
Label	**Find support's reactions S_x, S_y**
Input	J_i,M_j, S_k, F_i
Output	S_ix, S_iy for each support on joint i, so that if support is a roller one, $S_{ix} = 0$
Description	
Sources	Citation here
Ref. By	IM??

Number	IM3
Label	**Find zero members** $F_{mi} = 0$
Input	Joints with only two members The input is constrained so that there is no external force or reaction on the joint
Output	$F_{m1} = 0$,$F_{m2} = 0$
Description	
Sources	Citation here
Ref. By	IM**??**

Number	IM4
Label	**Find zero member in joints with 3 members**
Input	Joints with 2 collinear member and one non-collinear The input is constrained so no external force or reaction exists on the joint
Output	$F_{m3} = 0$, $F_{m2} = -F_{m1}$
Description	
Sources	Citation here
Ref. By	IM**??**

Derivation of Zero members

A proof of zero members in truss can be found in https://learnaboutstructures.com/Identifying-Zero-Force-Members.

4.2.7 Input Data Constraints

Table 1 shows the data constraints on the input-output variables. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. The column for software constraints restricts the range of inputs to reasonable values. The software constraints will be helpful in the design stage for picking suitable algorithms. The constraints are conservative, to give the user of the model the flexibility to experiment with unusual situations. The column of typical values is intended to provide a feel for a common scenario. The uncertainty column provides an estimate of the confidence with which the physical quantities can be measured. This information would be part of the input if one were performing an uncertainty quantification exercise.
The specification parameters in Table 1 are listed in Table 2.

Table 1: Input Variables

Var	Physical Constraints	Software Constraints	Typical Value	Uncertainty
n	$n > 3$	$n_{\text{min}} \leq n \leq n_{\text{max}}$	6	0%

(*) The count of Joints in a given truss is an integer number. it must be greater or equal to 3 to be considered a triangle. For small trusses, the number of joints is around 6.

Table 2: Specification Parameter Values

Var	Value
n_{min}	3
n_{max}	20

4.2.8 Properties of a Correct Solution

Table 3 shows the physical constraints on the output.

5 Requirements

This section provides the functional requirements and the business tasks that the software is expected to be complete, and the nonfunctional requirements, the qualities that the software is expected to exhibit.

Table 3: Output Variables

Var	Physical Constraints
F_{mi}	(by A??)

5.1 Functional Requirements

- R1: Input the values from Table 4
- R2:Echoing inputs as part of output.
- R3: Calculate support reactions from IM1, and internal forces from IM1, IM2, IM3
- R4: Check summation of internal forces is zero in each joint

Table 4: Required Inputs

Symbol	Description	Data Type
F	External forces on each joint Index	Array of Integer (N).
J	The location of all joint	Array of Real tuples
M	A pair of joint index for all members	Array of Integer tuples
S_1, S_2	pair of Joint Index as Location of supports and type of support	(Integer, Char)

5.2 Nonfunctional Requirements

NFR1: **Accuracy**

NFR2: **Usability**

NFR3: **Maintainability**

NFR4: **Portability**

6 Likely Changes

LC1: Give the likely changes, with a reference to the related assumption (aref), as appropriate.

7 Unlikely Changes

LC2: Give the unlikely changes. The design can assume that the changes listed will not occur.

8 Traceability Matrices and Graphs

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 are marked with an "X" may have to be modified as well. Table 5 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 6 shows the dependencies of instance models, requirements, and data constraints on each other. Table 7 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions.

You will have to modify these tables for your problem.

The traceability matrix is not generally symmetric. If GD1 uses A1, that means that GD1's derivation or presentation requires invocation of A1. A1 does not use GD1. A1 is "used by" GD1.

The traceability matrix is challenging to maintain manually. Please do your best. In the future tools (like Drasil) will make this much easier.

	T??	T??	T??	GD??	GD??	DD??	DD??	DD??	DD??	IM??	IM??	IM??	IM
T??													
T??			X										
T??													
GD??													
GD??	X												
DD??				X									
DD??				X									
DD??													
DD??								X					
IM??					X	X	X				X		
IM??					X		X		X	X			X
IM??		X											
IM??		X	X				X	X	X		X		

Table 5: Traceability Matrix Showing the Connections Between Items of Different Sections

The purpose of the traceability graphs is also to provide easy references on what has to be additionally modified if a certain component is changed. The arrows in the graphs represent dependencies. The component at the tail of an arrow is depended on by the component at the head of that arrow. Therefore, if a component is changed, the components that it points to should also be changed. Figure **??** shows the dependencies of theoretical models, general definitions, data definitions, instance models, likely changes, and assumptions on each other.

	IM??	IM??	IM??	IM??	4.2.7	R??	R??
IM??		X				X	X
IM??	X			X		X	X
IM??						X	X
IM??		X				X	X
R??							
R??						X	
R??					X		
R2	X	X				X	X
R??	X						
R??		X					
R??			X				
R??				X			
R4			X	X			
R??		X					
R??		X					

Table 6: Traceability Matrix Showing the Connections Between Requirements and Instance Models

	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??
T??	X																		
T??																			
T??																			
GD??		X																	
GD??			X	X	X	X													
DD??							X	X	X										
DD??			X	X						X									
DD??																			
DD??																			
IM??											X	X		X	X	X			X
IM??												X	X			X	X	X	
IM??														X					X
IM??													X					X	
LC??				X															
LC??								X											
LC??									X										
LC??											X								
LC??												X							
LC??															X				

Table 7: Traceability Matrix Showing the Connections Between Assumptions and Other Items

Figure ?? shows the dependencies of instance models, requirements, and data constraints on each other.

9 Development Plan

This section is optional. It is used to explain the plan for developing the software. In particular, this section gives a list of the order in which the requirements will be implemented. In the context of a course this is where you can indicate which requirements will be implemented as part of the course, and which will be "faked" as future work. This section can be organized as a prioritized list of requirements, or it could should the requirements that will be implemented for "phase 1", "phase 2", etc.

10 Values of Auxiliary Constants

Show the values of the symbolic parameters introduced in the report.

The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

The value of FRACTION, for the Maintainability NFR would be given here.

The following is not part of the template, just some things to consider when filing in the template.
Grammar, flow and LaTeXadvice:

- For Mac users `*.DS_Store` should be in `.gitignore`
- LaTeX and formatting rules
  - Variables are italic, everything else not, includes subscripts (link to document)
    - Conventions
    - Watch out for implied multiplication
  - Use BibTeX
  - Use cross-referencing
- Grammar and writing rules
  - Acronyms expanded on first usage (not just in table of acronyms)
  - "In order to" should be "to"

Advice on using the template:

- Difference between physical and software constraints
- Properties of a correct solution means *additional* properties, not a restating of the requirements (may be "not applicable" for your problem). If you have a table of output constraints, then these are properties of a correct solution.
- Assumptions have to be invoked somewhere
- "Referenced by" implies that there is an explicit reference
- Think of traceability matrix, list of assumption invocations and list of reference by fields as automatically generatable
- If you say the format of the output (plot, table etc), then your requirement could be more abstract

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 this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, 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?