Software Requirements Specification for Sun Catcher

Yu-Shiuan Wu

October 7, 2019

Contents

1 Reference Material **iii**
- 1.1 Table of Units . . . iii
- 1.2 Table of Symbols . . . iii
- 1.3 Abbreviations and Acronyms . . . iv

2 Introduction **1**
- 2.1 Purpose of Document . . . 1
- 2.2 Scope of Requirements . . . 1
- 2.3 Characteristics of Intended Reader . . . 2
- 2.4 Organization of Document . . . 2

3 General System Description **2**
- 3.1 System Context . . . 2
- 3.2 User Characteristics . . . 4
- 3.3 System Constraints . . . 4

4 Specific System Description **4**
- 4.1 Problem Description . . . 4
  - 4.1.1 Terminology and Definitions . . . 4
  - 4.1.2 Physical System Description . . . 5
  - 4.1.3 Goal Statements . . . 5
- 4.2 Solution Characteristics Specification . . . 6
  - 4.2.1 Assumptions . . . 6
  - 4.2.2 Theoretical Models . . . 6
  - 4.2.3 General Definitions . . . 7
  - 4.2.4 Data Definitions . . . 9
  - 4.2.5 Instance Models . . . 10
  - 4.2.6 Input Data Constraints . . . 11
  - 4.2.7 Properties of a Correct Solution . . . 12

5 Requirements **12**
- 5.1 Functional Requirements . . . 12
- 5.2 Nonfunctional Requirements . . . 12

6 Likely Changes **13**

7 Unlikely Changes **13**

8 Traceability Matrices and Graphs **13**

9 Values of Auxiliary Constants **13**

Revision History

Date	Version	Notes
Date 1	1.0	Notes
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 description of the unit and the SI name.

symbol	**unit**	**SI**
$^\circ$	angle	degree
$'$	angle	-
k	mass	kilo
m	length	metre
W	power	Watt ($\mathrm{W} = \mathrm{J\,s^{-1}}$)

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 heat transfer literature and with existing documentation for solar water heating systems. The symbols are listed in alphabetical order.

symbol	**unit**	**description**
$\theta_{S_{day}}$	$^\circ$	zenith angle of sun in the day
Φ_{SP}	$(^\circ\,')\mathrm{N/S}$	the latituade of the solar panel
δ_{day}	$^\circ$ N/S	the declination of the vertical noon sun in the day
$I_{S_{day}}$	$\frac{\mathrm{kW}}{\mathrm{m}^2}$	the intensity of the sun in a day

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
SC	Sun Catcher
T	Theoretical Model

[This template document assumes that a single program is being documented. If you are documenting a family of models, you should start with a commonality analysis. A separate template is provided for this. For program families you should look at Smith (2006); Smith et al. (2017). Single family member programs are often programs based on a single physical model. General purpose tools are usually documented as a family. Families of physical models also come up. —TPLT]

2 Introduction

Due to the increasing concepts of creating an earth-friendly environment, the kits using renewable energy bocome more popular in the market. In all type of renweable energy, solar energy is the most common type of renewable resourse for a home. However, it is an enpensive technology, and its cell efficiency is restricted by seaons. Therefore, Sun Catcher is created for home users to gain optimum energy from daily sunlight.

The following sections provides an overview of the Software Requirements Specification (SRS) for Sun Catcher. The developed program will be referred to as SC. This section explains the purpose of this document, the scope of the requirements, the characteristics of the intended reader, and the organization of the document.

2.1 Purpose of Document

The purpose of this document is to record the correct requirements of SC. The goal statement provided readers a consistent idea of what problem is solved. The theoretical models and the instance models, which state the mathematical terms supporting the theoretical models, are explained unambiguously for readers to reuse and verify the software. In the section of System Constraints, its contents will stay abstract because the content should only say what problem is being solved, but not how to solve it.

This document will be used as a starting point for subsequent development phases, including writing the design specification and the software verification and validation plan. The design document will show how the requirements are to be realized, including decisions on the numerical algorithms and programming environment. The verification and validation plan will show the steps that will be used to increase confidence in the software documentation and the implementation.

2.2 Scope of Requirements

The scope of the requirements includes stability analysis of a two-dimensional (2D) solar panel and the sun as the solar resource. The solar panel is assumed to place in a location near sea level, and the sky view above is unobstructed, with no trees, hills, clouds, dust, or haze ever blocking the sun.

2.3 Characteristics of Intended Reader

This section summarized the expectation of the readers' knowledge and skills for understanding this SRS. Readers should have a general knowledge of how a solar panel works and know the common factors that affect energy absorption.

2.4 Organization of Document

The organization of this document follows the template for an SRS for scientific computing software proposed by Koothoor [4] as well as Smith and Lai [8]. The presentation follows the standard pattern of presenting goals, theories, definitions, and assumptions. For readers that would like a more bottom up approach, they can start reading the instance models in Section: Instance Models) and trace back to find any additional information they require. The goal statements (Section: Goal Statements) are refined to the theoretical models and the theoretical models (Section: Theoretical Models) to the instance models (Section: Instance Models). The instance models provide the set of algebraic equations that must be solved.

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

System Context shows the design pattern of this program from users to the system than the output. The circle represents the users of this software. The rectangle represents the software system: Sun Catcher. The arrow represents the inputs that drive the software and the output expecting from the software.

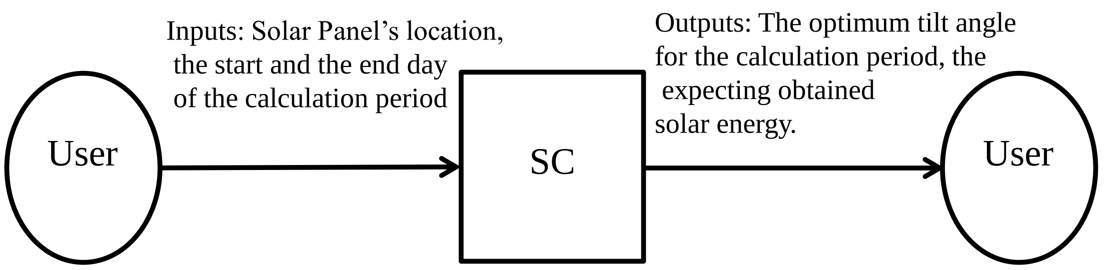

Inputs: Solar Panel's location, the start and the end day of the calculation period
User
SC
Outputs: The optimum tilt angle for the calculation period, the expecting obtained solar energy.
User

Figure 1: System Context

- User Responsibilities:
  - Provide the input data related to the solar panel.
  - Ensure the input date format matches the requirement from SC.
  - Ensure the condition of the solar panel, and the surrounding environment of the solar panel.
- ProgName Responsibilities:
  - Detect data type mismatch, such as a string of characters instead of a floating point number.
  - Comfirm the inputs quality to satisfy the required physical and software constraints.
  - Predict an optimum tilt angle, the expected gaining solar energy, and a report that shows the comparison of different results.

3.2 User Characteristics

The end-users of SC is expecting to understand the Level 1 Calculation method such as angle addition and subtraction theorems and understand the Level 1 celestial mechanics such as the formation of the latitude and the earth's tilt angle in the four seasons.

3.3 System Constraints

There are no system constraints.

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

Sun Catcher is intended to solve the unpredictable energy efficiency of solar panels. Due to the tilt angle of the earth when it rotates by axis, the latitude of the sun will consistently move. With the fixed location of the solar panel, it is unlikely to get the direct the direct sunlight for the maximum output. Therefore, it causes inadequate performance in gaining energy.

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:

- Declination of the Sun: The angle between the rays of the Sun and the plane of the Earth's equator.
- Tilt angle: The angle for adjusting the solar panel result in the panel get the direct sunlight.
- The period of days: The period of days depends on the input days of the users.
- Sun panel: The adjustable panel, which has solar cells on it, able to converse solar energy to power.
- Sun Intensity: The amount of incoming solar energy, or radiation, that reaches the Earth's surface.

- Latitude: Latitude is an angle which ranges from 0 ° at the Equator to 90° (North or South) at the poles.
- Zenith Angle: The angle between the sun and the vertical.

4.1.2 Physical System Description

The physical system of ProgName, as shown in Figure ?, includes the following elements:

PS1: Solar Panel: The panel with solar cell that able to absorb solar energy from sun.

PS2: Sun: Proving solar energy to solar panel

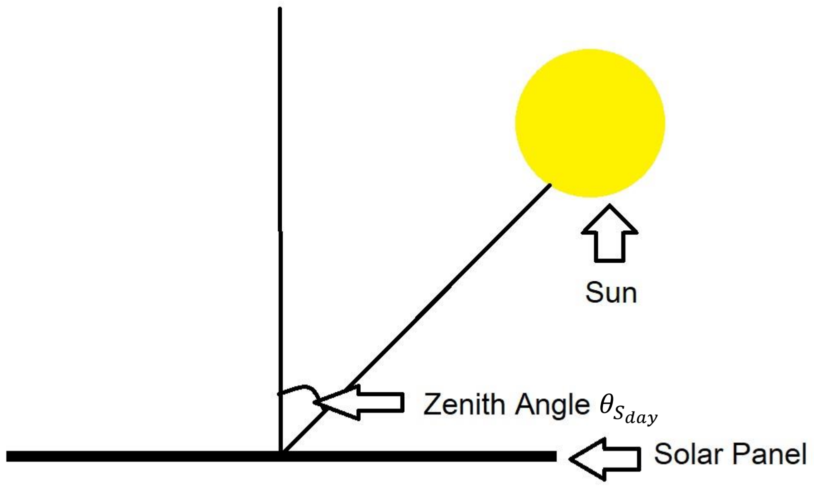


Figure 2: The Physic System

4.1.3 Goal Statements

Given the user located latitude, the day started to estimate angle,the day when the estimation end, the goal statements are:

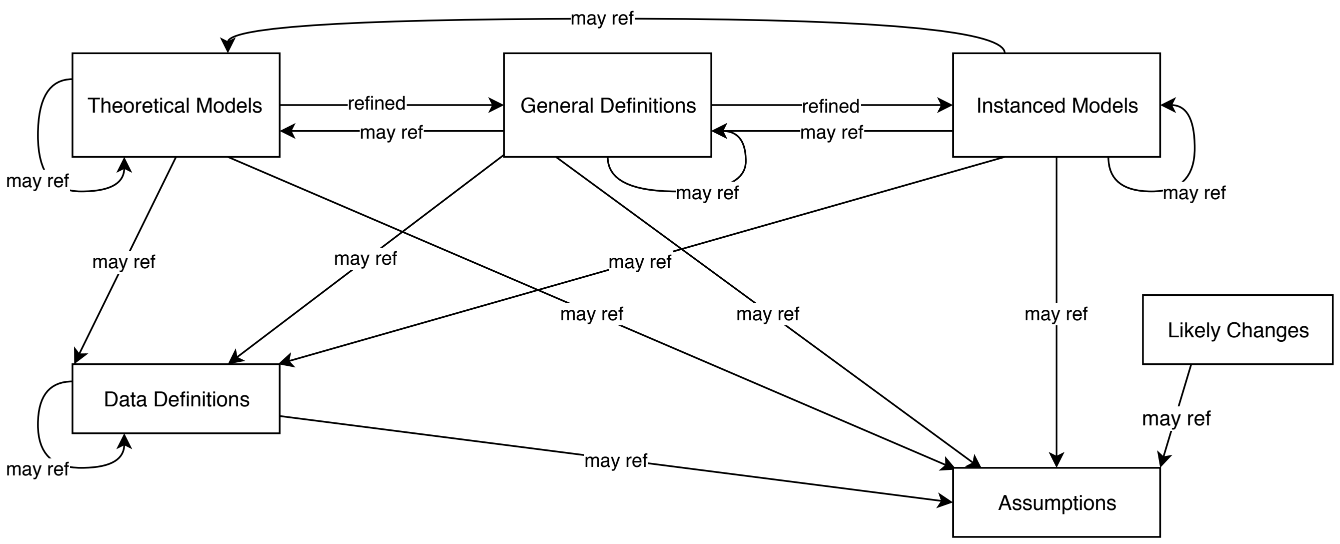
GS1: Predict the optimum tilt angle in the period of days.

GS2: Predict the optimum produced solar energy in the period of days.

GS3: Predict the possibility the amount of money user might save.

4.2 Solution Characteristics Specification

The instance models that govern SC are presented in Section: Instance Models. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.



The instance models that govern ProgName are presented in Subsection 4.2.5. 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], general definition [GD], data definition [DD], instance model [IM], or likely change [LC], in which the respective assumption is used.

A1 : The environmental condition of the solar panel location assumed as users have an unobstructed view of the sky, with no trees, hills, clouds, dust, or haze ever blocking the sun.(RefBy: IM1)

A2 : The default solar intensity is 1.35 kW. It was measured by the satellites.(RefBy: IM1)

4.2.2 Theoretical Models

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

Number	T1
Label	**Lambert's cosine law**
Equation	$I = I \times \frac{\cos\theta \times \partial\Omega \times \partial A}{\cos\theta \times \partial\Omega_0 \times \partial A_0}$
Description	I is the liuminous intensity. θ is the angle between the illuminating source and the normal source. Ω is the solid angle subtended by the aperture from the viewpoint of the emitting area element. A is the area of the observing aperture.
Source	https://en.wikipedia.org/wiki/Lambert%27s_cosine_law
Ref. By	RefBy: IM1

4.2.3 General Definitions

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

Number	GD1
Label	**Calculate Zenith Angle**
SI Units	$\circ$
Equation	$\theta_{S_{day}} = \begin{cases} \Phi_{SP} - \delta_{day}, & \text{if } 0 \leq \Phi_{SP}, \delta_{day} \leq 90°\text{N} \ \vee \ 0 \leq \Phi_{SP}, \delta_{day} \leq 90°\text{S} \\ \Phi_{SP} + \delta_{day}, & \text{if } \begin{cases} 0 \leq \Phi_{SP} \leq 90°\text{N} \ \wedge \ 0 \leq \delta_{day} \leq 90°\text{S} \\ 0 \leq \Phi_{SP} \leq 90°\text{S} \ \wedge \ 0 \leq \delta_{day} \leq 90°\text{N} \end{cases} \end{cases}$
Description	This equation describes the zenith angle associated with the solar panel. $\theta_{S_{day}}$ is the zeith angle of the sun in the day. Φ_{SP} is the local latitude of the solar panel. δ_{day} is the declination of the vertical noon sun in the day.
Source	https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19680025707.pdf
Ref. By	DD1

Detailed derivation of simplified rate of change of temperature

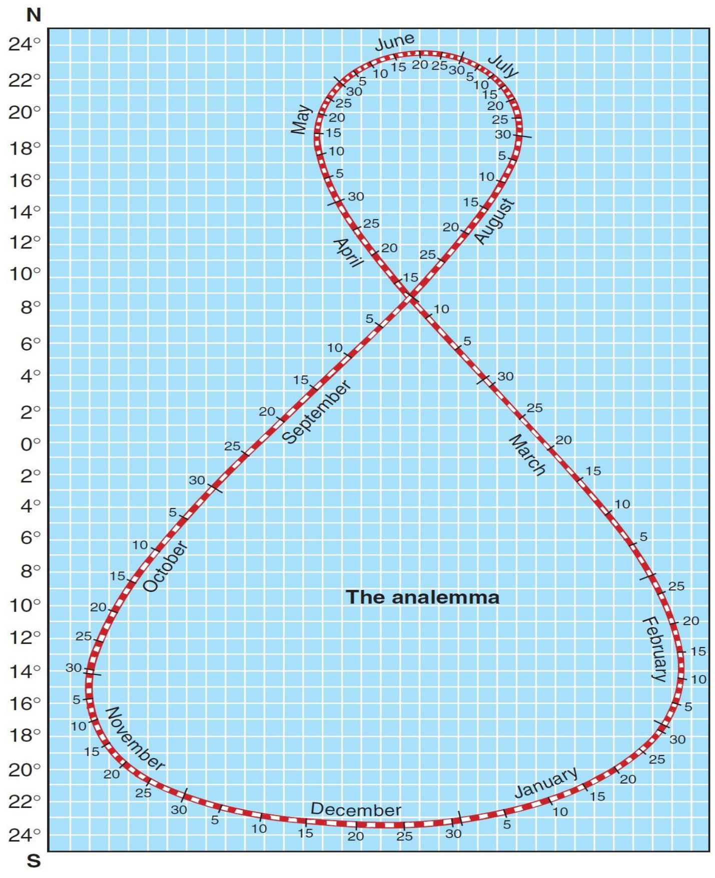
According to the resource [cite], it states the the solar elevation angle $\sin\alpha$ is determined by the relationship
$\sin\alpha = \sin\Phi \times \sin\delta + \cos\Phi \times \cos\delta \times \cos \text{h}$
where
α is the solar elevation.
Φ is the latituade.
δ is the solar declination.
h is the solar hour angle.
Base on the Level 1 celestial mechanics, the solar zenith angle is equal to 90° - the solar elevation angle
which can write as
$\cos\theta = \sin\alpha = \sin\Phi \times \sin\delta + \cos\Phi \times \cos\delta \times \cos \text{h}$ Base on the Assumptions section, it assumes that SC calculate the solar zenith angle at noon which mean solar hour angle = 0°.
Therefore, $\cos \text{h} = 0$
Rewrite the equation as,
$\cos\theta = \sin\alpha = \sin\Phi \times \sin\delta + \cos\Phi \times \cos\delta \times 1$
$\Rightarrow \cos\theta = \sin\alpha = \sin\Phi \times \sin\delta + \cos\Phi \times \cos\delta$
Base on the Level 1 calcucltion, we have

$\cos\Phi \pm \delta = \sin\alpha = \sin\Phi \times \sin\delta + \cos\Phi \times \cos\delta$
Therefore, we have $\theta = \Phi \pm \delta$

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	**The Declination of the Sun**
Symbol	δ_{day}
SI Units	degree°
Equation	-
Description	"day" means every day in the period of days which depends on the user's input. The degree of δ_{day} can be determinated by the the Analemma Figure.
Sources	https://www.cengage.com/resource_uploads/downloads/0495555061_137179.pdf
Ref. By	IM1



4.2.5 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 Goal Statements are solved by IM1.

Number	IM1
Label	**Calculating the Sun intensity**
Input	Φ_{SP}
Output	$I_{S_{day}} = 1.35 \cdot \frac{1.00}{1.35}^{sec(\theta_{S_{day}})}$
Description	The above equation gives the daily sun intensity at noon. 1.35 kW is the solar intensity measured by satellites. sec is the second of arc. $(\theta_{S_{day}})$ is the zenith angle of the sun in the day. The equation of $(\theta_{S_{day}})$ had described in GD1. This equation assumes that the earth is flat, so a factor was applied to account for the curvature of the earth(and therefore the earth's atmosphere). These factors, and the angle of the sun with respect to the panel, then determine the insolation on the panel.
Sources	https://www.solarpaneltilt.com/#other
Ref. By	-

Derivation of ...

[The derivation shows how the IM is derived from the TMs/GDs. In cases where the derivation cannot be described under the Description field, it will be necessary to include this subsection. —TPLT]

4.2.6 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.

(*) [you might need to add some notes or clarifications —TPLT]

Table 1: Input Variables

Var	Physical Constraints	Software Constraints	Typical Value	Uncertainty
L	$L > 0$	$L_{\text{min}} \leq L \leq L_{\text{max}}$	1.5 m	10%

Table 2: Specification Parameter Values

Var	Value
L_{min}	0.1 m

4.2.7 Properties of a Correct Solution

A correct solution must exhibit [fill in the details —TPLT]. [These properties are in addition to the stated requirements. There is no need to repeat the requirements here. These additional properties may not exist for every problem. Examples include conservation laws (like conservation of energy or mass) and known constraints on outputs (which are usually summarized in tabular form. A sample table is shown in Table 3 —TPLT]

Table 3: Output Variables

Var	Physical Constraints
T_W	$T_{\text{init}} \leq T_W \leq T_C$ (by A??)

5 Requirements

[The requirements refine the goal statement. They will make heavy use of references to the instance models. —TPLT]

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

5.1 Functional Requirements

R1: [Requirements for the inputs that are supplied by the user. This information has to be explicit. —TPLT]

R2: [It isn't always required, but often echoing the inputs as part of the output is a good idea. —TPLT]

R3: [Calculation related requirements. —TPLT]

R4: [Verification related requirements. —TPLT]

R5: [Output related requirements. —TPLT]

6 Likely Changes

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

7 Unlikely Changes

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

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 4 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 5 shows the dependencies of instance models, requirements, and data constraints on each other. Table 6 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. —TPLT]

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

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

9 Values of Auxiliary Constants

[Show the values of the symbolic parameters introduced in the report. —TPLT]

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

	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 4: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM??	IM??	IM??	IM??	4.2.6	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 5: 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 6: Traceability Matrix Showing the Connections Between Assumptions and Other Items

References

W. Spencer Smith. Systematic development of requirements documentation for general purpose scientific computing software. In *Proceedings of the 14th IEEE International Requirements Engineering Conference, RE 2006*, pages 209–218, Minneapolis / St. Paul, Minnesota, 2006. URL http://www.ifi.unizh.ch/req/events/RE06/.

W. Spencer Smith, John McCutchan, and Jacques Carette. Commonality analysis for a family of material models. Technical Report CAS-17-01-SS, McMaster University, Department of Computing and Software, 2017.

[The following is not part of the template, just some things to consider when filing in the template. —TPLT]
[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"

—TPLT]
[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 invokations 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

—TPLT]