Software Requirements Specification for Solar Cooker Energy Calculation: A Software for calculating heat loss and observed heat

Deesha Patel

February 1, 2023

Contents

Revision History

Date	Version	Notes
01/22/2023	1.0	Initial Release

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**
m	length	metre
kg	mass	kilogram
s	time	second
°C	temperature	centigrade
J	energy	joule
W	power	watt ($\mathrm{W} = \mathrm{J\,s^{-1}}$)

Derived units, like newtons, pascal, etc, should show their derivation (the units they are derived from) if their constituent units are in the table of units (that is, if the units they are derived from are used in the document). For instance, the derivation of pascals as $\mathrm{Pa} = \mathrm{N\,m^{-2}}$ is shown if newtons and m are both in the table. The derivations of newtons would not be shown if kg and s are not both in the table.

The symbol for units named after people use capital letters, but the name of the unit itself uses lower case. For instance, pascals use the symbol Pa, watts use the symbol W, teslas use the symbol T, newtons use the symbol N, etc. The one exception to this is degree Celsius. Details on writing metric units can be found on the NIST web-page.

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**
A_C	m^2	coil surface area
A_{in}	m^2	surface area over which heat is transferred in

Use your problems actual symbols. The si package is a good idea to use for units.

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
	put an expanded version of your program name here (as appropriate)
T	Theoretical Model

Add any other abbreviations or acronyms that you add

1.4 Mathematical Notation

This section is optional, but should be included for projects that make use of notation to convey mathematical information. For instance, if typographic conventions (like bold face font) are used to distinguish matrices, this should be stated here. If symbols are used to show mathematical operations, these should be summarized here. In some cases the easiest way to summarize the notation is to point to a text or other source that explains the notation.

This section was added to the template because some students use very domain specific notation. This notation will not be readily understandable to people outside of your domain. It should be explained.

This SRS template is based on **(author?)** (4**?**). It will get you started. You should not modify the section headings, without first discussing the change with the course instructor. Modification means you are not following the template, which loses some of the advantage of a template, especially standardization. Although the bits shown below do not include type information, you may need to add this information for your problem. If you are unsure, please can ask the instructor.

Feel free to change the appearance of the report by modifying the LaTeX commands.

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 (**? ?**). 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.

The SRS is not generally written, or read, sequentially. The SRS is a reference document. It is generally read in an ad hoc order, as the need arises. For writing an SRS, and for reading one for the first time, the suggested order of sections is:

- Goal Statement
- Instance Models
- Requirements
- Introduction
- Specific System Description

Guiding principles for the SRS document:

- Do not repeat the same information at the same abstraction level. If information is repeated, the repetition should be at a different abstraction level. For instance, there will be overlap between the scope section and the assumptions, but the scope section will not go into as much detail as the assumptions section.

The template description comments should be disabled before submitting this document for grading.

You can borrow any wording from the text given in the template. It is part of the template, and not considered an instance of academic integrity. Of course, you need to cite the source of the template.

When the documentation is done, it should be possible to trace back to the source of every piece of information. Some information will come from external sources, like terminology. Other information will be derived, like General Definitions.

An SRS document should have the following qualities: unambiguous, consistent, complete, validatable, abstract and traceable.

The overall goal of the SRS is that someone that meets the Characteristics of the Intended Reader (Section 2.3) can learn, understand and verify the captured domain knowledge. They should not have to trust the authors of the SRS on any statements. They should be able to independently verify/derive every statement made.

2 Introduction

As fossil fuels adversely affect the environment, many countries like India, where they have good Solar rays throughout the year, have started to implement devices for utilizing solar energy and transforming it into valuable energy. It includes Solar Water heaters, Solar panels, and Solar Cookers. It is indeed a fact that the demand for renewable energy has increased to deal with the problem. Focusing on solar cooker, several design such as Solar Panel Cooker, Solar Parabolic Cooker, and Solar Box Cooker has been proposed to utilize more solar energy. Using internal reflectors, we would like to improve the utilization of solar energy during cooking a food.

The following section provides an overview of Software Requirement Specification (SRS) for a box-type Solar Cooker. This section explains the purpose of the document, the scope of requirements, the characteristic of the intended reader, and the organization of a document.

2.1 Purpose of Document

We are going to use the Solar Box Cooker in this software. The main purpose of this document is to describe a mathematical model of a Solar Cooker which can provide a calculation for internal reflector in solar box that can help to improve the temperature in the box. It includes a variety of parameters that attempts to define the intended functionality required. Thus, this document provides detailed requirements of the software which will be used in planing for design stage. Therefore, this document is intended to be used as a reference to provide ad hoc access to all information necessary to understand and verify the model. This document describes goals, assumptions, theoretical models, and important definitions to understand the problem. The SRS is abstract because the content here says *what* the problem is. But it does not say anything related to *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. Although the SRS fits in a series of documents that follow the so-called waterfall model, the actual development process is not constrained in any way. Even when the waterfall model is not followed, as Parnas and Clements (3) point out, the most logical way to present the documentation is still to “fake” a rational design process.

2.2 Scope of Requirements

There are Numerous algorithm has been proposed to improve the efficiency of Solar Cooker. The scope of the requirement includes a mathematical model to determine the thermal function of a box-type solar cooker with an internal reflectors. With the help of different inputs, this system calculates the achieved temperature by implementing the proposed solution. However, this project not focusing on more than one iteration for the reflections.

2.3 Characteristics of Intended Reader

Firstly, Intended Reader or Reviewer should have knowledge of heat transfer theory and radiant solar energy. A person should have completed a Heat Transfer course during their bachelor of engineering (Mechanical Engineering expected). A reader should have knowledge about the coupled differential equation; offered in the Calculus course.

2.4 Organization of Document

The organization of this document follows the template for an SRS for scientific computing software proposed by (2) and (4). 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 4.2.5 and trace back to find any additional information they require. The goal statements are refined to the theoretical models, and the theoretical models to the instance models. The instance model (Section 4.2.5) to be solved is referred to as IM1. The instance model provides the Ordinary Differential Equation (ODE) that model the solar water heating system. SWHS solves this ODE.

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

The system context is shown in Figure 1 below. The circles represent the user, who is both responsible for handling the inputs and the outputs. The box represents the program itself, and the arrows indicate what data and information is passed from the user to the program.

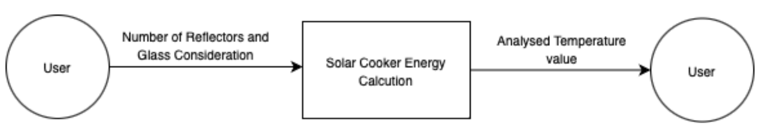

Figure 1: System Context

- User Responsibilities:
  - Provide required inputs including number of reflectors, glass area, thickness, dimension and size.
  - Ensure all inputs are in correct format.

- Responsibilities:
  - Detect data type mismatch, such as a string of characters instead of a floating point number
  - Determine if the inputs satisfy the required physical and software constraints such as thickness of the glass can not be a negative value
  - Calculate and plot the required outputs of temperature

3.2 User Characteristics

The end user of the system is expected to be familiar with Undergraduate level Calculus and basic physics. They should also know basics about the Reflector angle and heat flows in solar cooker.

3.3 System Constraints

There are no system constraints for this project.

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 (ODE) that models the Solar Cooker Reflections.

4.1 Problem Description

Solar Cooker Energy Calculation is intended to investigate the temperature inside the solar cooker box with internal reflectors.

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:

- Reflectivity: The fraction of radiation reflected by the surface is called the reflectivity (ρ)
- Absorptivity: The fraction of irradiation absorbed by the surface is called the absorptivity (α)
- Transmittivity: The fraction of radiation transmitted is called the transmissivity (τ)

- Emittance: the energy radiated by the surface of a body per second per unit area (ϵ)
- Reflector Angle: the angle between a reflected ray and the normal drawn at the point of incidence to a reflecting surface (θ)
- Heat flow Convection: Convection is the transfer of heat from one place to another due to the movement of fluid
- Heat flow radiation: a process where heat waves are emitted that may be absorbed, reflected, or transmitted through a colder body
- Heat Convection Coefficients: The rate of heat transfer between a solid surface and a fluid per unit surface area per unit temperature difference (h)
- Steffan-Boltzman constant: is a physical constant expressing the relationship between the heat radiation emitted by a black body and its absolute temperature (σ)
- Heat Flux: the amount of heat energy passing through a certain surface

4.1.2 Physical System Description

The physical system of Solar Cooker Energy Calculation, as shown in Figure 2, includes the following elements:

PS1: A cover with two flat glasses (glass 1 and glass 2)

PS2: Lead of the recipient and recipient itself

PS3: Fluid inside the recipient

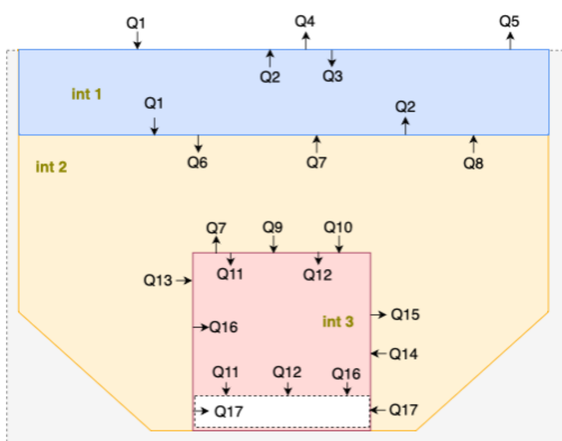

Figure 2: Heat Flows in Solar Cooker

4.1.3 Goal Statements

Given the Area of glass, Incident solar radiation, difference in temperature, Reflection angle, Steffan-Boltzman constant, heat transfer convection coefficient, the goal statements are:

GS1: predicts the Balance of energy in the recipient.

GS2: predicts the cooking power in the recipient.

4.2 Solution Characteristics Specification

The instance model (ODE) that governs Solar Cooker Energy Calculation is presented in Subsection 4.2.5. The information to understand the meaning of the instance model and its derivation is also presented, so that the instance model 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: Properties like emissivity(ϵ) reflectivity (ρ) and transmisivity (τ) have been considered constant [GD2, IM1]

A2: The material in the recipient (Container) is liquid for this case (For us, it's water). This implies that the temperature will not drop below melting point and not rise above the boiling point [IM1, T4.2.2]

A3: The temperature T_{int2} are obtained in function of others temperatures by means of the following supposition: [IM1]

$$T_{\text{int2}} = \frac{T_{\text{g2}}+T_t+T_r}{3}$$

A4: The solar radiation impact over the solar cooker occurs in perpendicular way. [IM1]

4.2.2 Theoretical Models

This section focuses on the general equations and laws that Solar Cooker Energy Calculation is based on. Theoretical models are sets of abstract mathematical equations or axioms for solving the problem described in Section "Physical System Description" (Section 4.1.2). Examples of theoretical models are physical laws, constitutive equations, relevant conversion factors, etc.

RefName: **T:CHTC**

Label: Convective Heat Transfer Coefficient

Equation: $\mathrm{h} = \frac{q}{\triangle T}$

Description: The above equation is used to calculate the heat transfer typically by convection or phase transition.

h is the heat transfer convection coefficient ($\mathrm{W/m^2\ K}$).

q is the thermal flux vector ($\mathrm{W\,m^{-2}}$).

$\triangle$T is the change in temperature.

Notes: None.

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

Ref. By: GD2, IM1

Preconditions for T:CHTC: None

Derivation for T:CHTC: Not Applicable

RefName: **T:SHE**

Label: Sensible Heat energy Calculation

Equation: Q $= Cm\triangle T$

Description: This calculation occurs until the highest or lowest temperature reach, as assumed in [A2]
Q is the quantity of heat transferred to or from the object
m is the mass of the object
C is the specific heat capacity of the material the object is composed of
$\triangle$ T is the resulting temperature change of the object

Notes: None.

Source: https://www.physicsclassroom.com/class/thermalP/Lesson-2/Measuring-the-Quantity

Ref. By: IM1

Preconditions for T:SHE: None

Derivation for T:SHE: Not Applicable

4.2.3 General Definitions

General Definitions (GDs) are a refinement of one or more TMs, and/or of other GDs. The GDs are less abstract than the TMs. Generally the reduction in abstraction is possible through invoking (using/referencing) Assumptions. For instance, the TM could be Newton's Law of Cooling stated abstracting. The GD could take the general law and apply it to get a 1D equation.

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

Some projects may not have any content for this section, but the section heading should be kept. Modify the examples below for your problem, and add additional definitions as appropriate.

Number	GD1
Label	**Heat Flow Convection**
SI Units	W
Equation	$Q(t) = hA\Delta T(t)$
Description	Heat Flow Convection is related to Newton's law of cooling describes convective cooling from a surface. The law is stated as: the rate of heat loss from a body is proportional to the difference in temperatures between the body and its surroundings. $Q(t)$ is the thermal flux ($\mathrm{W\,m^{-2}}$). h is the heat transfer coefficient ($\mathrm{W\,m^{-2}\,{}^{\circ}C^{-1}}$). A is the exposed surface area ($\mathrm{m^2}$). $\Delta T(t)$ is the time-difference (°C).
Source	https://en.wikipedia.org/wiki/Convection_(heat_transfer)
Ref. By	IM1

Number	GD2
Label	**Heat Flow Radiation**
SI Units	J
Equation	$Q(t) = \sigma e A \triangle T^4$
Description	An object emits radiant energy in all directions unless its temperature is absolute zero. If this energy strikes a receiver, part of it may be absorbed, part may be transmitted, and part may be reflected. Heat transfer from a hot to a cold object in this manner is known as radiation heat transfer . The higher the temperature, the greater is the amount of energy radiated. $Q(t)$ is the heat flow radiation ($\mathrm{W\,m^{-2}}$). σ is the Steffan-Boltzman constant ($5.669X10^{-8}W/m^2K^4$). e is the emissivity of object(W/m). A is the exposed surface area (m^2) $\triangle T(t)$ is the time-difference (°C).
Source	(1)
Ref. By	IM1

Detailed derivation of simplified rate of change of temperature

This may be necessary when the necessary information does not fit in the description field. Derivations are important for justifying a given GD. You want it to be clear where the equation came from.

4.2.4 Data Definitions

The Data Definitions are definitions of symbols and equations that are given for the problem. They are not derived; they are simply used by other models. For instance, if a problem depends on density, there may be a data definition for the equation defining density. The DDs are given information that you can use in your other modules.

All Data Definitions should be used (referenced) by at least one other model.

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given. Modify the examples below for your problem, and add additional definitions as appropriate.

Number	DD1
Label	**Heat flux over all different object**
Symbol	q
SI Units	$\mathrm{W\,m^{-2}}$
Equation	$q(t) = -\lambda \frac{\triangle T}{\triangle x}$
Description	It is necessary to know the thermal conductivity of a material if you want to calculate the heat energy transferred through it q is the heat flux λ is the thermal conductivity of the material $\triangle T$ is the temperature difference across the object $\triangle x$ is the distance of heat transfer (the thickness of the object)
Sources	`https://www.omnicalculator.com/physics/thermal-conductivity`
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.

Number	IM1
Label	**Balance of energy on the recipient to find T_r**
Input	A_r, ϵ_r, T_r, T_f, T_{int2}, A_m, T_f The input is constrained so that $\epsilon_t \leq 0$
Output	$T_f(t)$, $t \geq 0$, such that $\mathbf{m_r c_r \frac{dT_r}{dt}} = Q13 + 4Q14 - Q15 - Q16 - Q17$
Description	**Q13** $= A_r h_{\text{r-int2}}(T_{\text{int2}} - T_r)$ - Heat flow convection of recipient to the inner 2 **Q14** $= 4\sum_{i=1}^{n} \rho A_{\text{ref,n}} G \tau_g^2 cos(90 - \theta_{\text{ref,n}})$ - Heat flow reflection of incident radiation on the reflectors **Q15** $= A_r \sigma \epsilon_r (T_r^4 - T_{\text{g2}}^4)$ - Heat flow radiation of recipient toward glass 2 **Q16** $= A_r \sigma \epsilon_r (T_r^4 - T_{\text{f}}^4)$ - Heat flow radiation of recipient toward the fluid **Q17** $= A_m h_{\text{r-f}}(T_r - T_f)$ - Heat flow convection of recipient toward the fluid A_r is the Area of Reflector (m^2). ϵ_r is the Emittance of recipient (W/m). T_r is the Reflector temperature (°C). T_{g2} is a temperature of glass 2 (°C). T_{int2} is a temperature inside the box (°C). G is the Incidental solar radiation (W/m^2). T_f is a current temperature of Fluid (°C). The above equation applies as long as the fluid is in liquid form, $0 < T_f < 100^o\text{C}$, where 0^oC and 100^oC are the melting and boiling points of water(as we are using water for testing), respectively.
Sources	(5)
Ref. By	None

Number	IM2
Label	**Heat energy in the recipient**
Input	C_W, m_W, T_{init}, $T_W(t)$
Output	$E_W(t)$, $0 \leq t \leq t_{\text{final}}$, such that $E_W(t) = C_W m_W (T_W(t) - T_{\text{init}})$
Description	The above equation is derived using T4.2.2. E_W is the change in thermal energy of the liquid water relative to the energy at the initial temperature (T_{init}). C_W is the specific heat capacity of liquid water and m_W is the mass of the water. The change in temperature is the difference between the temperature at time t, T_W, and the initial temperature, T_{init}, this equation applies as long as $0 < T_W < 100^o$C (A2).
Sources	(?)
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.

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

4.2.7 Properties of a Correct Solution

A correct solution must exhibit fill in the details. These properties are in addition to the stated requirements. There is no need to repeat the requirements here. These additional

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

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

Table 3: Output Variables

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

This section is not for test cases or techniques for verification and validation. Those topics will be addressed in the Verification and Validation plan.

5 Requirements

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

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.

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

R3: Calculation related requirements.

R4: Verification related requirements.

R5: Output related requirements.

Every IM should map to at least one requirement, but not every requirement has to map

5.2 Nonfunctional Requirements

List your nonfunctional requirements. You may consider using a fit criterion to make them verifiable. The goal is for the nonfunctional requirements to be unambiguous, abstract and verifiable. This isn't easy to show succinctly, so a good strategy may be to give a "high level" view of the requirement, but allow for the details to be covered in the Verification and Validation document. An absolute requirement on a quality of the system is rarely needed. For instance, an accuracy of 0.0101 % is likely fine, even if the requirement is for 0.01 % accuracy. Therefore, the emphasis will often be more on describing now well the quality is achieved, through experimentation, and possibly theory, rather than meeting some bar that was defined a priori. You do not need an entry for correctness in your NFRs. The purpose of the SRS is to record the requirements that need to be satisfied for correctness. Any statement of correctness would just be redundant. Rather than discuss correctness, you can characterize how far away from the correct (true) solution you are allowed to be. This is discussed under accuracy.

NFR1: **Accuracy** Characterize the accuracy by giving the context/use for the software. Maybe something like, "The accuracy of the computed solutions should meet the level needed for <engineering or scientific application>. The level of accuracy achieved by shall be described following the procedure given in Section X of the Verification and Validation Plan." A link to the VnV plan would be a nice extra.

NFR2: **Usability** Characterize the usability by giving the context/use for the software. You should likely reference the user characteristics section. The level of usability achieved by the software shall be described following the procedure given in Section X of the Verification and Validation Plan. A link to the VnV plan would be a nice extra.

NFR3: **Maintainability** The effort required to make any of the likely changes listed for should be less than FRACTION of the original development time. FRACTION is then a symbolic constant that can be defined at the end of the report.

NFR4: **Portability** This NFR is easier to write than the others. The systems that should run on should be listed here. When possible the specific versions of the potential operating environments should be given. To make the NFR verifiable a statement could be made that the tests from a given section of the VnV plan can be successfully run on all of the possible operating environments.

- Other NFRs that might be discussed include verifiability, understandability and reusability.

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 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.

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??	DD1	DD??	DD??	DD??	IM1	IM??	IM2	IM??
T??													
T??			X										
T??													
GD??													
GD??	X												
DD1				X									
DD??				X									
DD??													
DD??								X					
IM1					X	X	X				X		
IM??					X		X		X	X			X
IM2		X											
IM??		X	X				X	X	X		X		

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

	IM1	IM??	IM2	IM??	4.2.6	R??	R??
IM1		X				X	X
IM??	X			X		X	X
IM2						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													
DD1							X	X	X										
DD??			X	X						X									
DD??																			
DD??																			
IM1											X	X		X	X	X			X
IM??												X	X			X	X	X	
IM2														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

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 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.

References

[1] Ibrahim Dinçer and Marc A. Rosen. *Thermal Energy Storage : Systems and Applications*. John Wiley Sons, Canada, 2010.

[2] Nirmitha Koothoor. A document drive approach to certifying scientific computing software. Master's thesis, McMaster University, Hamilton, Ontario, Canada, 2013.

[3] David L. Parnas and P.C. Clements. A rational design process: How and why to fake it. *IEEE Transactions on Software Engineering*, 12(2):251–257, February 1986.

[4] W. Spencer Smith and Lei Lai. A new requirements template for scientific computing. In J. Ralyté, P. Ågerfalk, and N. Kraiem, editors, *Proceedings of the First International Workshop on Situational Requirements Engineering Processes – Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05*, pages 107–121, Paris, France, 2005. In conjunction with 13th IEEE International Requirements Engineering Conference.

[5] Hilario Terres, Arturo Lizardi, Raymundo Lpez, Mabel Vaca, and Sandra Ch°vez. Mathematical model to study solar cookers box-type with internal reflectors. *Energy Procedia*, 57:1583–1592, 2014. 2013 ISES Solar World Congress.

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?