Software Requirements Specification for Solar Water Heating Systems

Thulasi Jegatheesan

June 2, 2019

Contents

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

The unit system used throughout is SI (Système International d'Unités). In addition to the basic units, several derived units are also used. For each unit, the table lists the symbol, a description and the SI name.

Symbol	Description
°C	temperature (centigrade)
J	energy (joule)
kg	mass (kilogram)
m	length (metre)
s	time (second)
W	power (watt)

Table 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	Description	Units
A_C	Heating coil surface area	m^2
A_{in}	Surface area over which heat is transferred in	m^2
A_{out}	Surface area over which heat is transferred out	m^2
C	Specific heat capacity	$\frac{\mathrm{J}}{(\mathrm{kg}^{\circ}\mathrm{C})}$
C^L	Specific heat capacity of a liquid	$\frac{\mathrm{J}}{(\mathrm{kg}^{\circ}\mathrm{C})}$
C_W	Specific heat capacity of water	$\frac{\mathrm{J}}{(\mathrm{kg}^{\circ}\mathrm{C})}$
D	Diameter of tank	m
E	Sensible heat	J
E_W	Change in heat energy in the water	J
g	Volumetric heat generation per unit volume	$\frac{\mathrm{W}}{\mathrm{m}^3}$

Symbol	Description	Units
h	Convective heat transfer coefficient	$\frac{\mathrm{W}}{(\mathrm{m^2 \, ^\circ C})}$
h_C	Convective heat transfer coefficient between coil and water	$\frac{\mathrm{W}}{(\mathrm{m^2 \, ^\circ C})}$
L	Length of tank	m
m	Mass	kg
m_W	Mass of water	kg
$\mathbf{n}$	Normal Vector	–
q	Heat flux	$\frac{\mathrm{W}}{\mathrm{m^2}}$
q_C	Heat flux into the water from the coil	$\frac{\mathrm{W}}{\mathrm{m^2}}$
q_{in}	Heat flux input	$\frac{\mathrm{W}}{\mathrm{m^2}}$
q_{out}	Heat flux output	$\frac{\mathrm{W}}{\mathrm{m^2}}$
$\mathbf{q}$	Thermal flux vector	$\frac{\mathrm{W}}{\mathrm{m^2}}$
S	Surface	$\mathrm{m^2}$
T	Temperature	$^\circ$C
t	Time	s
ΔT	Change in temperature	$^\circ$C
T_C	Temperature of the heating coil	$^\circ$C
T_{env}	Temperature of the environment	$^\circ$C
t_{final}	Final time	s
T_{init}	Initial temperature	$^\circ$C
T_W	Temperature of the water	$^\circ$C
V	Volume	$\mathrm{m^3}$
V_W	Volume of water	$\mathrm{m^3}$
π	Circumference to diameter ratio	–
ρ	Density	$\frac{\mathrm{kg}}{\mathrm{m^3}}$
ρ_W	Density of water	$\frac{\mathrm{kg}}{\mathrm{m^3}}$
τ_W	ODE parameter for water	s
∇	Gradient	–

Table 2

1.3 Abbreviations and Acronyms

Abbreviation	Full Form
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
ODE	Ordinary Differential Equation
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
SWHS	Solar Water Heating System
TM	Theoretical Model
UC	Unlikely Change
Uncert.	Typical Uncertainty

Table 3

2 Introduction

Due to increasing cost, diminishing availability, and negative environmental impact of fossil fuels, there is a higher demand for renewable energy sources and energy storage technology. Solar water heating systems provide a novel way of storing energy. The following section provides an overview of the Software Requirements Specification (SRS) for solar water heating systems. The developed program will be referred to as Solar Water Heating System (SWHS). This section explains the purpose of this document, the scope of the system, the characteristics of the intended reader, and the organization of the document.

2.1 Purpose of Document

The main purpose of this document is to describe the modelling of solar water heating system. The goals and theoretical models used in the SWHS code are provided, with an emphasis on explicitly identifying assumptions and unambiguous definitions. 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. The SRS is abstract because the contents say what problem is being solved, but do not say 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 point out [4], the most logical way to present the documentation is still to "fake" a rational design process.

2.2 Scope of Requirements

The scope of the requirements includes thermal analysis of a single solar water heating tank. Given the appropriate inputs, SWHS predicts the temperature and thermal energy histories for the water.

2.3 Characteristics of Intended Reader

Reviewers of this documentation should have an understanding of heat transfer theory from level 3 or 4 mechanical engineering and differential equations from level 1 and 2 calculus. The users of SWHS can have a lower level of expertise, as explained in Section: User Characteristics.

2.4 Organization of Document

The organization of this document follows the template for an SRS for scientific computing software proposed by [**dParnas1972**] and [**parnasClements1984**]. 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 model to be solved is referred to as IM: eBalanceOnWtr. 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

Fig:SysCon shows the system context. A circle represents an external entity outside the software, the user in this case. A rectangle represents the software system itself (SWHS). Arrows are used to show the data flow between the system and its environment. SWHS

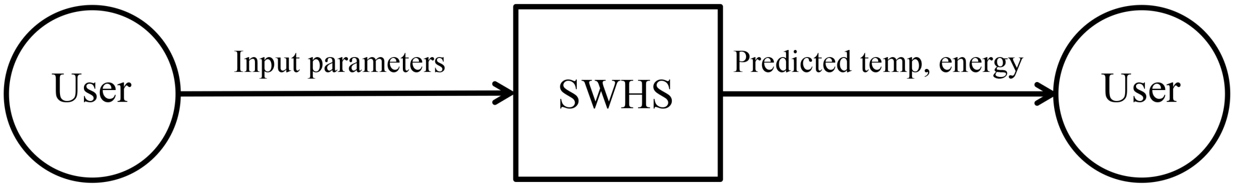

Figure 1: Fig:SysCon: System Context

is mostly self-contained. The only external interaction is through the user interface. The responsibilities of the user and the system are as follows:

- User Responsibilities:
  - Provide the input data to the system, ensuring no errors in the data entry
  - Take care that consistent units are used for input variables
- SWHS 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
  - Calculate the required outputs

3.2 User Characteristics

The end user of SWHS should have an understanding of undergraduate Level 1 Calculus and Physics.

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, and definitions that are used.

4.1 Problem Description

SWHS is a computer program developed to investigate the heating of water in a solar water heating tank.

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.

- Heat flux: The rate of thermal energy transfer through a given surface per unit time
- Specific heat capacity: The amount of energy required to raise the temperature of the unit mass of a given substance by a given amount
- Thermal conduction: The transfer of heat energy through a substance
- Transient: Changing with time

4.1.2 Physical System Description

The physical system of SWHS, as shown in Fig:Tank, includes the following elements:

PS1: Tank containing water.

PS2: Heating coil at bottom of tank. (q_C represents the heat flux into the water from the coil.)

4.1.3 Goal Statements

Given the temperature of the heating coil, the initial temperature of the water, and the material properties, the goal statements are:

ater-Temperature: Predict the temperature of the water over time.

dict-Water-Energy: Predict the change in heat energy in the water over time.

4.2 Solution Characteristics Specification

The instance models that govern SWHS 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.

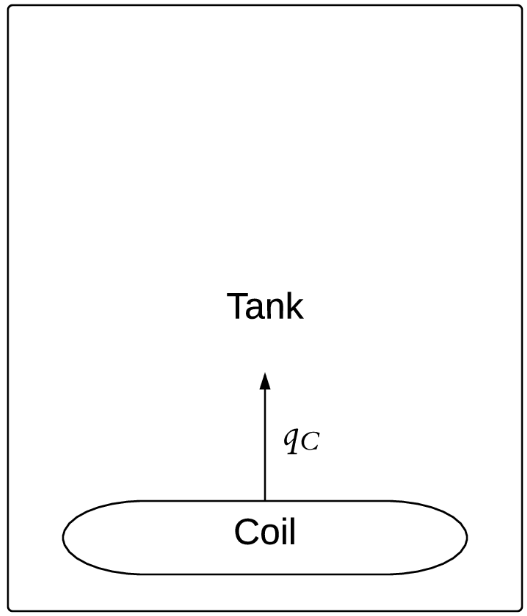

Figure 2: Solar water heating tank, with heat flux from heating coil of q_C

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 Models Section: Theoretical Models, General Definitions Section: General Definitions, Data Definitions Section: Data Definitions, Instance Models Section: Instance Models, Likely Changes Section: Likely Changes, or Unlikely Changes Section: Unlikely Changes, in which the respective assumption is used.

rmal-Energy-Only: The only form of energy that is relevant for this problem is thermal energy. All other forms of energy, such as mechanical energy, are assumed to be negligible. TM: consThermE.

er-Coeffs-Constant: All heat transfer coefficients are constant over time. GD: nwtnCooling.

Temp-Across-Tank: The water in the tank is fully mixed, so the temperature of the water is the same throughout the entire tank. GD: rocTempSimp.

stant-over-Volume: The density of water has no spatial variation; that is, it is constant over their entire volume. GD: rocTempSimp.

stant-over-Volume: The specific heat capacity of water has no spatial variation; that is, it is constant over its entire volume. GD: rocTempSimp.

Cooling-Coil-Water: Newton's law of convective cooling applies between the heating coil and the water. DD: ht_flux_C.

onstant-over-Time: The temperature of the heating coil is constant over time. LC: Temperature-Coil-Variable-Over-Day DD: ht_flux_C.

stant-over-Length: The temperature of the heating coil does not vary along its length. LC: Temperature-Coil-Variable-Over-Length DD: ht_flux_C.

o-Temp-Discharge: The model only accounts for charging of the tank, not discharging. The temperature of the water can only increase, or remain constant; it cannot decrease. This implies that the initial temperature is less than (or equal to) the temperature of the heating coil. LC: Discharging-Tank IM: eBalanceOnWtr.

ter-Always-Liquid: The operating temperature range of the system is such that the material (water in this case) is always in liquid state. That is, the temperature will not drop below the melting point temperature of water, or rise above its boiling point temperature. UC: Water-Fixed-States TM: sensHtE IM: heatEInWtr IM: eBalanceOnWtr.

ct-Insulation-Tank: The tank is perfectly insulated so that there is no heat loss from the tank. LC: Tank-Lose-Heat.

neration-By-Water: No internal heat is generated by the water; therefore, the volumetric heat generation per unit volume is zero. UC: No-Internal-Heat-Generation IM: eBalanceOnWtr.

eric-Pressure-Tank: The pressure in the tank is atmospheric, so the melting point temperature and boiling point temperature of water are 0°C and 100°C, respectively IM: heatEInWtr.

4.2.2 Theoretical Models

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

Refname	**TM:consThermE**
Label	Conservation of thermal energy
Equation	$$-\nabla \cdot \mathbf{q} + g = \rho C \frac{\partial T}{\partial t}$$
Description	∇ is the gradient (Unitless) $\mathbf{q}$ is the thermal flux vector ($\frac{\text{W}}{\text{m}^2}$) g is the volumetric heat generation per unit volume ($\frac{\text{W}}{\text{m}^3}$) ρ is the density ($\frac{\text{kg}}{\text{m}^3}$) C is the specific heat capacity ($\frac{\text{J}}{(\text{kg}^\circ\text{C})}$) t is the time (s) T is the temperature (°C)
Notes	The above equation gives the law of conservation of energy for transient heat transfer in a material of specific heat capacity C ($\frac{\text{J}}{(\text{kg}^\circ\text{C})}$) and density, ρ ($\frac{\text{kg}}{\text{m}^3}$), where $\mathbf{q}$ is the thermal flux vector ($\frac{\text{W}}{\text{m}^2}$), g is the volumetric heat generation per unit volume ($\frac{\text{W}}{\text{m}^3}$), T is the temperature (°C), t is time (s), and ∇ is the degree of steepness of a graph at any point. For this equation to apply, other forms of energy, such as mechanical energy, are assumed to be negligible in the system (A: Thermal-Energy-Only).
Source	Fourier Law of Heat Conduction and Heat Equation
RefBy	GD: rocTempSimp.

Refname	TM:sensHtE
Label	Sensible heat energy (no state change)
Equation	$$E = C^L m \Delta T$$
Description	E is the sensible heat (J) C^L is the specific heat capacity of a liquid ($\frac{\text{J}}{(\text{kg}^\circ\text{C})}$) m is the mass (kg) ΔT is the change in temperature ($^\circ$C)
Notes	E occurs as long as the material does not reach a temperature where a phase change occurs, as assumed in A: Water-Always-Liquid.
Source	Definition of Sensible Heat
RefBy	IM: heatEInWtr.

4.2.3 General Definitions

This section collects the laws and equations that will be used to build the instance models.

Refname	GD:nwtnCooling
Label	Newton's law of cooling
Units	$\frac{\text{W}}{\text{m}^2}$
Equation	$$q\left(t\right)=h\Delta T\left(t\right)$$
Description	q is the heat flux ($\frac{\text{W}}{\text{m}^2}$) t is the time (s) h is the convective heat transfer coefficient ($\frac{\text{W}}{(\text{m}^2{}^\circ\text{C})}$) ΔT is the change in temperature (${}^\circ$C)
Notes	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. $\mathbf{q}\left(t\right)$ is the thermal flux ($\frac{\text{W}}{\text{m}^2}$). h is the heat transfer coefficient, assumed independant of T (A: Heat-Transfer-Coeffs-Constant) ($\frac{\text{W}}{(\text{m}^2{}^\circ\text{C})}$). $\Delta T\left(t\right)=T\left(t\right)-T_{env}\left(t\right)$ is the time-dependant thermal gradient between the environment and the object (${}^\circ$C).
Source	[1, (pg. 8)]
RefBy	

Refname	**GD:rocTempSimp**
Label	Simplified rate of change of temperature
Equation	$$mC\frac{d\,T}{d\,t}=q_{in}A_{in}-q_{out}A_{out}+gV$$
Description	m is the mass (kg) C is the specific heat capacity ($\frac{\text{J}}{(\text{kg}^\circ\text{C})}$) t is the time (s) T is the temperature ($^\circ$C) q_{in} is the heat flux input ($\frac{\text{W}}{\text{m}^2}$) A_{in} is the surface area over which heat is transferred in (m^2) q_{out} is the heat flux output ($\frac{\text{W}}{\text{m}^2}$) A_{out} is the surface area over which heat is transferred out (m^2) g is the volumetric heat generation per unit volume ($\frac{\text{W}}{\text{m}^3}$) V is the volume (m^3)
Notes	The basic equation governing the rate of change of temperature, for a given volume V, with time. m is the mass (kg). C is the specific heat capacity ($\frac{\text{J}}{(\text{kg}^\circ\text{C})}$). T is the temperature ($^\circ$C) and t is the time (s). q_{in} and q_{out} are the in and out heat transfer rates, respectively ($\frac{\text{W}}{\text{m}^2}$). A_{in} and A_{out} are the surface areas over which the heat is being transferred in and out, respectively (m^2). g is the volumetric heat generated ($\frac{\text{W}}{\text{m}^3}$). V is the volume (m^3).
Source	–
RefBy	GD: rocTempSimp IM: eBalanceOnWtr.

Detailed derivation of simplified rate of change of temperature. Integrating TM: consThermE over a volume (V), we have:

$$-\int_V \nabla\cdot\mathbf{q}\,dV+\int_V g\,dV=\int_V \rho C\frac{\partial\,T}{\partial\,t}\,dV$$

Applying Gauss's Divergence Theorem to the first term over the surface S of the volume, with $\mathbf{q}$ as the thermal flux vector for the surface and $\mathbf{n}$ as a unit outward normal vector for a surface:

$$-\int_S \mathbf{q} \cdot \mathbf{n}\, dS + \int_V g\, dV = \int_V \rho C \frac{\partial\, T}{\partial\, t}\, dV$$

We consider an arbitrary volume. The volumetric heat generation per unit volume is assumed constant. Then (1) can be written as:

$$q_{in} A_{in} - q_{out} A_{out} + gV = \int_V \rho C \frac{\partial\, T}{\partial\, t}\, dV$$

Where q_{in}, q_{out}, A_{in}, and A_{out} are explained in GD: rocTempSimp. Assuming ρ, C and T are constant over the volume, which is true in our case by A: Constant-Water-Temp-Across-Tank, A: Density-Water-Constant-over-Volume, and A: Specific-Heat-Energy-Constant-over-Volume, we have:

$$\rho C V \frac{d\, T}{d\, t} = q_{in} A_{in} - q_{out} A_{out} + gV$$

Using the fact that ρ=m/V, (2) can be written as:

$$mC \frac{d\, T}{d\, t} = q_{in} A_{in} - q_{out} A_{out} + gV$$

4.2.4 Data Definitions

This section collects and defines all the data needed to build the instance models.

Refname	DD:ht.flux.C
Label	Heat flux into the water from the coil
Symbol	q_C
Units	$\frac{\text{W}}{\text{m}^2}$
Equation	$$q_C = h_C \left(T_C - T_W \left(t\right)\right)$$
Description	q_C is the heat flux into the water from the coil ($\frac{\text{W}}{\text{m}^2}$) h_C is the convective heat transfer coefficient between coil and water ($\frac{\text{W}}{(\text{m}^2{}^\circ\text{C})}$) T_C is the temperature of the heating coil (°C) T_W is the temperature of the water (°C) t is the time (s)
Notes	A: Newton-Law-Convective-Cooling-Coil-Water A: Temp-Heating-Coil-Constant-over-Time A: Temp-Heating-Coil-Constant-over-Length
Source	[2]
RefBy	IM: eBalanceOnWtr.

4.2.5 Instance Models

This section transforms the problem defined in Section: Problem Description into one which is expressed in mathematical terms. It uses concrete symbols defined in Section: Data Definitions to replace the abstract symbols in the models identified in Section: Theoretical Models and Section: General Definitions. The goal GS: Predict-Water-Temperature is met by IM: eBalanceOnWtr and the goal GS: Predict-Water-Energy is met by IM: heatEInWtr.

Refname	IM:eBalanceOnWtr
Label	Energy balance on water to find the temperature of the water
Input	T_C, T_{init}, t_{final}, A_C, h_C, C_W, m_W
Output	T_W
Input Constraints	$$T_{init} \leq T_C$$
Output Constraints	$$0 < t < t_{final}$$
Equation	$$\frac{d\,T_W}{d\,t} = \frac{1}{\tau_W} \left(T_C - T_W\left(t\right)\right)$$
Description	t is the time (s) T_W is the temperature of the water (°C) τ_W is the ODE parameter for water (s) T_C is the temperature of the heating coil (°C)
Notes	T_W is the temperature of the water (°C). T_C is the temperature of the heating coil (°C). $\tau_W = \frac{m_W C_W}{h_C A_C}$ is a constant (s). The above equation applies as long as the water is in liquid form, $0 < T_W < 100$ (°C) where 0 (°C) and 100 (°C) are the melting and boiling point temperatures of water, respectively (A: Water-Always-Liquid).
Source	[2, (with PCM removed)]
RefBy	UC: No-Internal-Heat-Generation FR: Output-Input-Derived-Quantities FR: Find-Mass FR: Calculate-Temperature-Water-Over-Time.

Derivation of the energy balance on water: To find the rate of change of T_W, we look at the energy balance on water. The volume being considered is the volume of water V_W, which has mass m_W and specific heat capacity of water, C_W . Heat transfer occurs in the water from the coil as q_C, over area A_C. No heat transfer occurs to the outside of the tank, since it has been assumed to be perfectly insulated (A: Charging-Tank-No-Temp-Discharge) . Assuming no volumetric heat generation per unit volume (A: No-Internal-Heat-Generation-By-Water), $g = 0$. Therefore, the equation for GD: rocTempSimp can be written as:

$$m_W C_W \frac{d\,T_W}{d\,t} = q_C A_C$$

Using DD: ht_flux_C , this can be written as:

$$m_W C_W \frac{d\,T_W}{d\,t} = h_C A_C \,(T_C - T_W)$$

Dividing (3) by $m_W C_W$, we obtain:

$$\frac{d\,T_W}{d\,t} = \frac{h_C A_C}{m_W C_W}\,(T_C - T_W)$$

Setting $\tau_W = m_W \; C_W \; / \; h_C \; A_C$, Equation (4) can be written in its final form as:

$$\frac{d\,T_W}{d\,t} = \frac{1}{\tau_W}\,(T_C - T_W)$$

Refname	**IM:heatEInWtr**
Label	Heat energy in the water
Input	T_{init}, m_W, C_W, m_W
Output	E_W
Input Constraints	
Output Constraints	$$0 < t < t_{final}$$
Equation	$$E_W\left(t\right) = C_W m_W \left(T_W\left(t\right) - T_{init}\right)$$
Description	E_W is the change in heat energy in the water (J) t is the time (s) C_W is the specific heat capacity of water ($\frac{\text{J}}{(\text{kg}^\circ\text{C})}$) m_W is the mass of water (kg) T_W is the temperature of the water ($^\circ$C) T_{init} is the initial temperature ($^\circ$C)
Notes	The above equation is derived using TM: sensHtE. E_W is the change in thermal energy of the liquid water relative to the energy at the initial temperature (T_{init}) (J). C_W is the specific heat capacity of liquid water ($\frac{\text{J}}{(\text{kg}^\circ\text{C})}$) and m_W is the mass of the water (kg). The change in temperature is the difference between the temperature at time t (s), T_W and the initial temperature, T_{init} ($^\circ$C). This equation applies as long as $0 < T_W < 100^\circ$C (A: Water-Always-Liquid, A: Atmospheric-Pressure-Tank).
Source	[2]
RefBy	FR: Calculate-Change-Heat_Energy-Water-Over-Time.

4.2.6 Data Constraints

Table:InDataConstraints and Table:OutDataConstraints show the data constraints on the input and output variables, respectively. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. 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 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 column for software constraints restricts the range of inputs to reasonable values.

Var	Physical Constraints	Software Constraints	Typical Value	Uncert.
A_C	$A_C > 0$	$A_C \leq {A_C}^{max}$	$120.0 \cdot 10^{-3}$ m^2	10%
C_W	$C_W > 0$	${C_W}^{min} < C_W < {C_W}^{max}$	$4.186 \cdot 10^3$ $\frac{\text{J}}{(\text{kg}^\circ\text{C})}$	10%
D	$D > 0$	–	$412.0 \cdot 10^{-3}$ m	10%
h_C	$h_C > 0$	${h_C}^{min} \leq h_C \leq {h_C}^{max}$	$1.0 \cdot 10^3$ $\frac{\text{W}}{(\text{m}^2{}^\circ\text{C})}$	10%
L	$L > 0$	$L_{min} \leq L \leq L_{max}$	1.5 m	10%
T_C	$0 < T_C < 100$	–	50.0 °C	10%
t_{final}	$t_{final} > 0$	$t_{final} < {t_{final}}^{max}$	$50.0 \cdot 10^3$ s	10%
T_{init}	$0 < T_{init} < 100$	–	40.0 °C	10%
ρ_W	$\rho_W > 0$	${\rho_W}^{min} < \rho_W \leq {\rho_W}^{max}$	$1.0 \cdot 10^3$ $\frac{\text{kg}}{\text{m}^3}$	10%

Table 4: Input Data Constraints

Var	Physical Constraints
T_W	$T_{init} \leq T_W \leq T_C$
E_W	$E_W \geq 0$

Table 5: Output Data Constraints

4.2.7 Properties of a Correct Solution

FIXME.

5 Requirements

This section provides the functional requirements, the tasks and behaviours that the software is expected to complete, and the non-functional requirements, the qualities that the software is expected to exhibit.

5.1 Functional Requirements

This section provides the functional requirements, the tasks and behaviours that the software is expected to complete.

-Initial-Quantities: Input the following quantities described in Table:Input-Variable-Requirements, which define the tank parameters, material properties and initial conditions.

Find-Mass: Use the inputs in FR: Input-Initial-Quantities to find the mass needed for IM: eBalanceOnWtr, using $m_W = V_W \rho_W = \pi \left(\frac{D}{2}\right)^2 L \rho_W$, where V_W is the volume of water.

sical_Constraints: Verify that the inputs satisfy the required physical constraints shown in Table:InDataConstraints.

Derived-Quantities: Output the input quantities and derived quantities in the following list: the quantities from FR: Input-Initial-Quantities, the mass from FR: Find-Mass, and τ_W (from IM: eBalanceOnWtr).

-Water-Over-Time: Calculate and output the temperature of the water ($T_W(t)$) over the simulation time (from IM: eBalanceOnWtr).

-Water-Over-Time: Calculate and output the change in heat energy in the water ($E_W(t)$) over the simulation time (from IM: heatEInWtr).

Symbol	Unit	Description
L	m	length of tank
D	m	diameter of tank
A_C	m^2	heating coil surface area
T_C	$^\circ\mathrm{C}$	temperature of the heating coil
ρ_W	$\frac{\mathrm{kg}}{\mathrm{m}^3}$	density of water
C_W	$\frac{\mathrm{J}}{(\mathrm{kg}^\circ\mathrm{C})}$	specific heat capacity of water
h_C	$\frac{\mathrm{W}}{(\mathrm{m}^{2\circ}\mathrm{C})}$	convective heat transfer coefficient between coil and water
T_{init}	$^\circ\mathrm{C}$	initial temperature
t_{final}	s	final time
A_{tol}	–	absolute tolerance
R_{tol}	–	relative tolerance
C_{tol}	–	relative tolerance for conservation of energy

Table 6: Input Variable Requirements

5.2 Non-Functional Requirements

This section provides the non-functional requirements, the qualities that the software is expected to exhibit.

Correct: The outputs of the code have the properties described in Section: Properties of a Correct Solution.

Verifiable: The code is tested with complete verification and validation plan.

Understandable: The code is modularized with complete module guide and module interface specification.

Reusable: The code is modularized.

Maintainable: The traceability between requirements, assumptions, theoretical models, general definitions, data definitions, instance models, likely changes, unlikely changes, and modules is completely recorded in traceability matrices in the SRS and module guide.

6 Likely Changes

Variable-Over-Day: A: Temp-Heating-Coil-Constant-over-Time - The temperature of the heating coil will change over the course of the day, depending on the energy received from the sun.

iable-Over-Length: A: Temp-Heating-Coil-Constant-over-Length - The temperature of the heating coil will actually change along its length as the water within it cools.

Discharging-Tank: A: Charging-Tank-No-Temp-Discharge - The model currently only accounts for charging of the tank. That is, increasing the temperature of the water to match the temperature of the coil. A more complete model would also account for discharging of the tank.

Tank-Lose-Heat: A: Perfect-Insulation-Tank - Any real tank cannot be perfectly insulated and will lose heat.

7 Unlikely Changes

Vater-Fixed-States: A: Water-Always-Liquid - It is unlikely for the change of water from liquid to a solid, or from liquid to gas to be considered.

l-Heat-Generation: A: No-Internal-Heat-Generation-By-Water - Is used for the derivations of IM: eBalanceOnWtr.

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 row of that component that are marked with an "X" should be modified as well. Table:Tracey shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table:TraceyRI shows the dependencies of instance models, requirements, and data constraints on each other. Table:TraceyRIs shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions.

	FR: Check-Input-with-Physical_Constraints	FR: Inp
Table:InDataConstraints	X	
Table:Input-Variable-Requirements		X
A: Atmospheric-Pressure-Tank		
A: Charging-Tank-No-Temp-Discharge		
A: Constant-Water-Temp-Across-Tank		
A: Density-Water-Constant-over-Volume		
A: Heat-Transfer-Coeffs-Constant		
A: Newton-Law-Convective-Cooling-Coil-Water		
A: No-Internal-Heat-Generation-By-Water		
A: Perfect-Insulation-Tank		
A: Specific-Heat-Energy-Constant-over-Volume		
A: Thermal-Energy-Only		
A: Temp-Heating-Coil-Constant-over-Length		
A: Temp-Heating-Coil-Constant-over-Time		
A: Water-Always-Liquid		
TM: consThermE		
IM: eBalanceOnWtr		
FR: Find-Mass		
IM: heatEInWtr		
DD: ht_flux_C		
FR: Input-Initial-Quantities		
GD: rocTempSimp		
TM: sensHtE		

	T1 (TM: consThermE)	T2 (TM: sensHtE)	GD1 (GD: nwtnCooling)	G
T1 (TM: consThermE)				
T2 (TM: sensHtE)				

	T1 (TM: consThermE)	T2 (TM: sensHtE)	GD1 (GD: nwtnCooling)	G
GD1 (GD: nwtnCooling)	X			
GD2 (GD: rocTempSimp)			X	
DD1 (DD: ht_flux_C)				X
IM1 (IM: eBalanceOnWtr)				

Table 8: Traceability Matrix Showing the Con
Models

	IM1 (IM: eBalanceOnWtr)	IM2 (IM: hea
IM1 (IM: eBalanceOnWtr)		
IM2 (IM: heatEInWtr)		
R1 (FR: Input-Initial-Quantities)		
R2 (FR: Find-Mass)	X	
R3 (FR: Check-Input-with-Physical_Constraints)		
R4 (FR: Output-Input-Derived-Quantities)	X	
R5 (FR: Calculate-Temperature-Water-Over-Time)	X	
R6 (FR: Calculate-Change-Heat_Energy-Water-Over-Time)		X

	A1 (A: Thermal-Energy-Only)	A2 (A: Heat-Tran
T1 (TM: consThermE)	X	
T2 (TM: sensHtE)		X
GD1 (GD: nwtnCooling)		
GD2 (GD: rocTempSimp)		
DD1 (DD: ht_flux_C)		
IM1 (IM: eBalanceOnWtr)		
IM2 (IM: heatEInWtr)		
LC1 (LC: Temperature-Coil-Variable-Over-Day)		
LC2 (LC: Temperature-Coil-Variable-Over-Length)		
LC3 (LC: Discharging-Tank)		

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. Fig:TraceyA shows the dependencies of theoretical models, general definitions, data definitions, instance models, likely changes, and assumptions on each other. Fig:TraceyR shows the dependencies of instance models, requirements, and data constraints on each other.

9 Values of Auxiliary Constants

This section contains the standard values that are used for calculations in SWHS.

Symbol	Description	Value	Unit
L_{min}	minimum length of tank	$100.0 \cdot 10^{-3}$	m
L_{max}	maximum length of tank	50	m
${\rho_W}^{min}$	minimum density of water	950	$\frac{\text{kg}}{\text{m}^3}$
${\rho_W}^{max}$	maximum density of water	1000	$\frac{\text{kg}}{\text{m}^3}$
${C_W}^{min}$	minimum specific heat capacity of water	4170	$\frac{\text{J}}{(\text{kg}^\circ\text{C})}$
${C_W}^{max}$	maximum specific heat capacity of water	4210	$\frac{\text{J}}{(\text{kg}^\circ\text{C})}$
${h_C}^{min}$	minimum convective heat transfer coefficient between coil and water	10	$\frac{\text{W}}{(\text{m}^2{}^\circ\text{C})}$
${h_C}^{max}$	maximum convective heat transfer coefficient between coil and water	10000	$\frac{\text{W}}{(\text{m}^2{}^\circ\text{C})}$
${t_{final}}^{max}$	maximum final time	86400	s

Table 11: Auxiliary Constants

10 References

[1] F. P. Incropera et al. *Fundamentals of Heat and Mass Transfer*. 6th ed. Hoboken, New Jersey: John Wiley and Sons, 2007.

[2] Nirmitha Koothoor. “A document drive approach to certifying scientific computing software”. MA thesis. Hamilton, ON, Canada: McMaster University, 2013.

[3] Marilyn Lightstone. *Derivation of tank/pcm model*. From Marilyn Lightstone’s Personal Notes. 2012.

[4] David L. Parnas and P. C. Clements. “A rational design process: How and why to fake it”. In: *IEEE Transactions on Software Engineering* 12.2 (Feb. 1986), ”251–257”.

[5] W. Spencer Smith and Lei Lai. “A new requirements template for scientific computing”. In: *Proceedings of the First International Workshop on Situational Requirements Engineering Processes - Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05*. Ed. by PJ Agerfalk, N. Kraiem, and J. Ralyte. In conjunction with 13th IEEE International Requirements Engineering Conference, Paris, France, 2005, "107–121".