

Software Requirements Specifications for Quarters

James Anthony (1135089)

Wenqiang Chen(1155437)

Carolyn Chong (1139105)

Kevin Ly (1144604)

October 21, 2015

Contents

1	General Information	4
1.1	Purpose	4
1.2	Scope	4
1.3	Acronyms, Abbreviations, and Symbols	4
1.4	Overview of Document	4
2	Plan	5
2.1	Software Description	5
2.2	Test Team	5
2.3	Milestones	5
2.3.1	Location	5
2.3.2	Dates and Deadlines	5
2.4	Budget	6
3	Software Specification	6
3.1	Functional Requirements	6
3.2	Nonfunctional Requirements	6
4	Evaluation	6
4.1	Methods and Constraints	6
4.1.1	Methodology	6
4.1.2	Extent of Testing	7
4.1.3	Test Tools	7
4.1.4	Testing Constraints	7
4.2	Data Evaluation	7
4.2.1	Testing Criteria	7
4.2.2	Testing Data Reduction	7
5	System Test Description	7
5.1	Faulty Input	7
5.1.1	Input	7
5.1.2	Preparation and Procedure	10
5.2	Unrecommended Input	10
5.2.1	Input	10
5.2.2	Preparation and Procedure	12
5.3	Closed Form Solution for Latent Heating	12
5.3.1	Means of Control	12
5.3.2	Input	12
5.3.3	Expected Output	12
5.3.4	Preparation and Procedure	12
5.4	Alternative Algorithm	13

5.4.1	Means of Control	13
5.4.2	Input	13
5.4.3	Expected Output	13
5.4.4	Preparation and Procedure	13
5.5	Comparison to Original Implementation	13
5.5.1	Means of Control	13
5.5.2	Input	13
5.5.3	Expected Output	14
5.5.4	Procedure	14
5.5.5	Preparation	14
6	Temperature Modules	14
6.1	Module Information	14
6.1.1	Module Inputs	14
6.1.2	Module Outputs	15
6.1.3	Related Modules	15
6.2	Test Data	15
6.2.1	Inputs	15
6.2.2	Expected Outputs	15
7	Energy Modules	16
7.1	Module Information	16
7.1.1	Module Inputs	16
7.1.2	Module Outputs	16
7.2	Test Data	16
7.2.1	Inputs	16
7.2.2	Expected Outputs	17
8	Event Modules	17
8.1	Module Information	17
8.1.1	Module Inputs	17
8.1.2	Module Outputs	17
8.1.3	Related Modules	18
8.2	Test Data	18
8.2.1	Inputs	18
8.2.2	Expected Outputs	18
9	Appendix	19

Revision History

Date	Comments
October 21, 2015	Created first draft.

Template

This document makes use of the XXX Template for all of its organization.

1 Introductions

The following section provides an overview of the Verification and Validation (V&V) Plan for a solar water heating systems incorporating phase change material simulator. This section explains the purpose of this document, the scope of the system, common definitions, acronyms and abbreviations that are used in the document, and an overview of the following sections

1.1 Objectives

The main purpose of this document is to describe the verification and validation process that will be used to test a simulation for solar water heating systems incorporating PCM. This document is intended to be used as a reference for all future testing and will be used to increase confidence in the software implementation.

This document will be used as a starting point for the verification and validation report. The test cases presented within this document will be executed and the output will be analyzed to determine if the software is implemented correctly.

1.2 Scope

1.3 Constraints

symbol	description
QA	Quality assurance
SRS	Software requirements specification
V&V	Verification and validation
V&VP	Verification and validation plan
V&VR	Verification and validation report
PCM	Phase change material
SWHS	Solar Water Heating System
ϵ	10^{-2}

1.4 Acronyms and Definitions

The following sections provide more detail about the V&V of a solar water heating simulator. Information about the testing process is provided, and the software specifications that were

discussed in the SRS document are stated. The evaluation process that will be followed during testing is outlined, and test cases for both the system testing and unit testing are provided

2 Types of Tests

This section provides a description of the software that is being tested, the team that will perform the testing, the milestones for the testing phase, and the budget allocated to the testing.

2.1 Functional Testing

The software being tested is a simulator for a SWHS incorporating PCM. Given the physical parameters of the system, including dimensions, properties of the water and PCM, and relevant physical constants, the simulator calculates the changes in temperature and energy of the water and PCM over time.

2.2 Test Team

The team that will execute the test cases, write and review the V&VR consists of:

- Maya Grab
- Dr. Spencer Smith
- Thulasi Jegatheesan

2.3 Milestones

2.3.1 Location

The location where the testing will be performed is Hamilton Ontario. The institution that will be performing the testing is McMaster University.

2.3.2 Dates and Deadlines

Test Case:

The creation of the test cases for both system testing and unit testing is scheduled to begin June 1st 2015. The deadline for the creation of the test cases is June 15th 2015.

Test Case Implementation:

Implementing code for the automation of the unit testing is scheduled to begin June 15th 2015. The implementation period is expected to last approximately two weeks and has a deadline of June 30th 2015.

Verification and Validation Report:

The writing of the V&VR is scheduled to begin July 1st 2015 and end on July 15th 2015.

2.4 Budget

The budget for the testing of this system is being funded by McMaster University and NSERC.

3 Software Specification

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.

3.1 Functional Requirements

- Input the physical constants, properties and initial temperatures of water and PCM, and dimensions of the tank
- Verify that the inputs satisfy the required physical constraints
- Compute the calculated values required to solve the governing differential equations
- Calculate the temperatures and energy of water and PCM over time.

3.2 Nonfunctional Requirements

Priority nonfunctional requirements are correctness, understandability, reliability, and maintainability.

4 Evaluation

This section first presents the methods and constraints that are to be used during the evaluation process. This is followed by how the data obtained by the testing will be evaluated, which includes: how the data will be recorded, how to move from one test to the next, and how to determine if the test was successful.

4.1 Methods and Constraints

4.1.1 Methodology

The testing of the SWHS will be fully automated with the exception one testing method, where a change to the ODE solver algorithm is required.

4.1.2 Extent of Testing

The extent of testing that will be employed is extensive. The unit test cases below provide complete code coverage and will increase confidence in the verification of the software. The system test cases increase confidence in the validation of the system.

4.1.3 Test Tools

A unit testing framework will be used to implement the unit test cases and run them automatically.

The following equation will be implemented in a script in order to compare the outputs of different implementation:

$$\Delta_{\text{relative}} = \frac{\text{True} - \text{Calculated}}{\text{True}}$$

4.1.4 Testing Constraints

There are currently no anticipated limitations on the testing.

4.2 Data Evaluation

4.2.1 Testing Criteria

Test criteria are divided into two categories:

1. **Numerical:** testing results will be compared to expected results within an allowable margin of error ϵ .
2. **Error catching:** testing will ensure that faulty and unrecommended inputs are caught in exceptions and warnings.

4.2.2 Testing Data Reduction

The results of the test data will be evaluated on a PASS/FAIL basis. If the actual results match the expected results the test will be considered a PASS, otherwise the test is considered a FAIL.

5 System Test Description

5.1 Faulty Input

5.1.1 Input

The input will be based on the Data Constraints on Input table provided in the appendix of this document (borrowed from Input Variables table in the SRS document). Each test

will correspond to one entry from the physical constraints column, altering a specific input variable to a non-permissible value. The list of inputs is in order with the entries in the table, though note there are several cases tested for each constraint described.

Table 1: Faulty Input Cases

No.	Input	Expected Outcome	MsgID
01	$L = -2$	error: Tank length must be > 0	input:L
02	$L = 0$	error: Tank length must be > 0	input:L
03	$D = -2$	error: Tank diameter must be > 0	input:diam
04	$D = 0$	error: Tank diameter must be > 0	input:diam
05	$V_P = -0.05$	error: PCM volume must be > 0	input:Vp
06	$V_P = 0$	error: PCM volume must be > 0	input:Vp
07	$L = 0.5$ $D = 0.5$ $V_P = 0.5$	error: Tank volume must be $>$ PCM volume	input:VpVt
08	$V_P = 0.199974938771605$ (tank volume) $A_P = 2.208137511613965$ (tank surface area)	error: Tank volume must be $>$ PCM volume	input:VpVt
09	$A_P = -1.5$	error: PCM area must be > 0	input:Ap
10	$A_P = 0$	error: PCM area must be > 0	input:Ap
11	$\rho_P = -1000$	error: ρ_P must be > 0	input:rho_p
12	$\rho_P = 0$	error: ρ_P must be > 0	input:rho_p
13	$T_{\text{melt}}^P = -10$	error: T_{melt} must be > 0 and $< T_C$	input:Tmelt

14	$T_{\text{melt}}^P = 0$	error: T_{melt} must be > 0 and $< T_C$	input:Tmelt
15	$T_{\text{melt}}^P = 45$ $T_C = 40$	error: T_{melt} must be > 0 and $< T_C$	input:Tmelt
16	$C_P^S = -1000$	error: C_{PS} must be > 0	input:C_ps
17	$C_P^S = 0$	error: C_{PS} must be > 0	input:C_ps
18	$C_P^L = -1000$	error: C_{PL} must be > 0	input:C_pl
19	$C_P^L = 0$	error: C_{PL} must be > 0	input:C_pl
20	$H_f = -200000$	error: Hf must be > 0	input:Hf
21	$H_f = 0$	error: Hf must be > 0	input:Hf
22	$A_C = -0.12$	error: Ac must be > 0	input:Ac
23	$A_C = 0$	error: Ac must be > 0	input:Ac
24	$T_C = -50$	error: T_{melt} must be > 0 and $< T_C$	input:Tmelt
25	$T_C = 0$	error: T_{melt} must be > 0 and $< T_C$	input:Tmelt
26	$T_C = 100$	error: Tc must be > 0 and < 100	input:Tc
27	$T_C = 110$	error: Tc must be > 0 and < 100	input:Tc
28	$\rho_W = -1000$	error: ρ_W must be > 0	input:rho_w
29	$\rho_W = 0$	error: ρ_W must be > 0	input:rho_w
30	$C_W = -4000$	error: C_W must be > 0	input:C_w
31	$C_W = 0$	error: C_W must be > 0	input:C_w
32	$h_C = -1000$	error: hc must be > 0	input:hc
33	$h_C = 0$	error: hc must be > 0	input:hc
34	$h_P = -1000$	error: hp must be > 0	input:hp
35	$h_P = 0$	error: hp must be > 0	input:hp

36	$T_{\text{init}} = -5$	error: Tinit must be > 0 and < 100	input:Tinit
37	$T_{\text{init}} = 0$	error: Tinit must be > 0 and < 100	input:Tinit
38	$T_{\text{init}} = 100$	error: Tc must be > Tinit	input:TcTinit
39	$T_{\text{init}} = 110$	error: Tc must be > Tinit	input:TcTinit
40	$T_{\text{init}} = 45$	error: Tinit must be < 0 Tmelt	input:TinitTmelt
41	$T_{\text{init}} = 50$	error: Tc must be > Tinit	input:TcTinit
42	$T_{\text{init}} = 60$	error: Tc must be > Tinit	input:TcTinit
43	$t_{\text{final}} = 0$	error: tfinal must be > 0	input:tfinal
44	$t_{\text{final}} = -50000$	error: tfinal must be > 0	input:tfinal

5.1.2 Preparation and Procedure

Input files are to be generated for each test case and be stored in an appropriate directory. The automated test should include a setup procedure that would add the directory to the Matlab path. Each test case runs main.m on an input file and checks for the appropriate MsgId.

5.2 Unrecommended Input

5.2.1 Input

The input will be based on the Data Constraints on Input table provided in the appendix of this document (borrowed from Input Variables table in the SRS document). Each test will correspond to one entry from the physical constraints column, altering a specific input variable to a non-advisable value. The list of inputs is in order with the entries in the table, though note there are several cases tested for each constraint described.

Table 2: Unrecommended Input Cases

No.	Input	Expected Outcome	MsgID
01	$L = 0.01$ $V_P = 0.001$	It is recommended that $0.1 \leq L \leq 50$	inputwarn:L
02	$L = 55$	It is recommended that $0.1 \leq L \leq 50$	inputwarn:L
03	$L = 30$	It is recommended that $0.002 \leq D/L \leq 200$	inputwarn:diam
04	$D = 400$	It is recommended that $0.002 \leq D/L \leq 200$	inputwarn:diam
05	$L = 15$ $D = 4.12$ $V_P = 0.0006$	It is recommended that Vp be $\geq 0.0001\%$ of Vt	inputwarn:VpVt
06	$A_P = 110$	It is recommended that $V_p \leq A_p \leq 2 * V_p / 0.001$ inputwarn:VpAp	
07	$\rho_P = 450$	It is recommended that $500 < \rho_p < 20000$	inputwarn:rho_p
08	$\rho_P = 20005$	It is recommended that $500 < \rho_p < 20000$	inputwarn:rho_p
09	$C_P^S = 90$	It is recommended that $100 < C_{ps} < 4000$	inputwarn:C_ps
10	$C_P^S = 5000$	It is recommended that $100 < C_{ps} < 4000$	intwarn:C_ps
11	$C_P^L = 90$	It is recommended that $100 < C_{pl} < 5000$	intwarn:C_lp
12	$C_P^L = 5005$	It is recommended that $100 < C_{pl} < 5000$	intwarn:C_pl
13	$H_f = \min$		
14	$H_f = \max$		
15	$A_C = 0.7$	It is recommended that $A_c \leq \pi * D / 2$	intwarn:Ac
16	$\rho_W = 900$	It is recommended that $950 < \rho_w \leq 1000$	intwarn:rho_w
17	$\rho_W = 1010$	It is recommended that $950 < \rho_w \leq 1000$	intwarn:rho_w

18	$C_W = 4160$	It is recommended that $4170 < C_w < 4210$	intwarn:C_w
19	$C_W = 4220$	It is recommended that $4170 < C_w < 4210$	intwarn:C_w
20	$t_{\text{final}} = 86500$	It is recommended that $0 < t_{\text{final}} < 86400$	intwarn:tfinal

5.2.2 Preparation and Procedure

Input files are to be generated for each test case and be stored in an appropriate directory. The automated test should include a setup procedure that would add the directory to the Matlab path. Each test case runs main.m on an input file and checks for the appropriate MsgId.

5.3 Closed Form Solution for Latent Heating

5.3.1 Means of Control

A closed form solution for the temperature of water is derived using Maple18 for a case of latent heating. The solution vector generated over time will be compared to the latent heating stage from the output generated by Standard Input Variables.

5.3.2 Input

The file will be based on the Standard input table and its output. Since the closed form solution is only correct for latent heating, the initial temperature will be set at the melting temperature of the PCM, and the time span will be the amount of time required for full melting.

5.3.3 Expected Output

The delta between the standard out vector and the closed form solution must be $< \epsilon$

5.3.4 Preparation and Procedure

The closed form solution found:

$$T_P(t) = T_{\text{init}}$$

$$T_W(t) = \frac{(T_{\text{init}} - T_C)e^{-\frac{(\eta+1)t}{\tau_W}} + T_{\text{init}}\eta + T_C}{\eta + 1}$$

5.4 Alternative Algorithm

5.4.1 Means of Control

The program will be manually compared to an implementation substituting the ode45 algorithm for ode23.

5.4.2 Input

The input will be the standard working input, run once through the ode45 implementation and once through the ode23 implementation.

5.4.3 Expected Output

The delta between the output vectors must be $< \epsilon$.

5.4.4 Preparation and Procedure

The ode45 function will be manually substituted with ode23 algorithm in main.m in order to compare the implementations. Both functions will be run on the input file and results will be compared.

5.5 Comparison to Original Implementation

5.5.1 Means of Control

Valid output from the current Matlab implementation will be compared to output from the original Fortran implementation.

5.5.2 Input

The standard input file for Matlab and the corresponding file in Fortran format will be ran through their respective implementations.

The standard input file will be modified to five different variations, as described in the table below.

Table 3: Faulty Input Cases

No.	Purpose	Δ Input
01		Standard
02	Set coil temperature just above melting temperature of PCM.	$T_C = 44.21^\circ C$ $T_{\text{melt}} = 44.2^\circ C$

03	Set t at exactly the initial melting time according to No.1 (Fortran implementation).	$t = 20570\text{s}$
04	Set t just above the initial melting time.	$t = 20580\text{s}$

5.5.3 Expected Output

The delta between the output vectors should be $< \epsilon$ where $\epsilon = 10^{-2}$.

5.5.4 Procedure

5.5.5 Preparation

6 Temperature Modules

6.1 Module Information

This testing suite will test the three temperature modules handling the governing differential equations for temperature of water and temperature of PCM.

- Temperature1.m handles the case where $T_P < T_{\text{melt}}$.
- Temperature2.m handles the case where $T_P = T_{\text{melt}}$.
- Temperature3.m handles the case where $T_P > T_{\text{melt}}$.

6.1.1 Module Inputs

The Temperature modules take as input:

- a time t ;
- a temperature vector T where $T(1) = T_W$ and $T(2) = T_P$;
- an input parameters structures containing the typical simulator parameters

6.1.2 Module Outputs

The Temperature modules output a vector dTdt where:

- $dTdt(1) = \frac{dT_W}{dt}$
- $dTdt(2) = \frac{dT_P}{dt}$
- $dTdt(3) = \frac{dQ_P}{dt}$ (temperature2.m only)

6.1.3 Related Modules

The module loadparams.m was used in testing in order to load the input parameters structure into each function.

6.2 Test Data

6.2.1 Inputs

Table 4: Temperature Tests Input

Test	Input
temperature 1	$t = 100$ $T = [40.7, 40.5]$ params
temperature 2a	$t = 3000$ $T = [44.2, 44.2]$ params
temperature 2b	$t = 4000$ $T = [45, 44.2]$ params
temperature 3	$t = 25000$ $T = [47, 46.5]$ params

6.2.2 Expected Outputs

Table 5: Temperature Tests Expected Outputs

Test	Input
temperature 1	[0.00139536, 0.002708315]
temperature 2a	[0.001108642, 0, 0]

temperature 2b	$[-0.000573435, 0, 960]$
temperature 3	$[-0.00038229, 0.005249596]$

7 Energy Modules

7.1 Module Information

This testing suite will test the three energy modules handling the governing equations for energy of water and PCM.

- energy1.m handles the case where $T_P < T_{\text{melt}}$.
- energy2.m handles the case where $T_P = T_{\text{melt}}$.
- energy3.m handles the case where $T_P > T_{\text{melt}}$.

7.1.1 Module Inputs

The Energy modules take as input:

- a temperature matrix T where $T(:, 1) = T_W$, $T(:, 2) = T_P$ and (energy2 only) $T(:, 3) = Q_P$
- an input parameters structures containing the typical simulator parameters

7.1.2 Module Outputs

The Energy modules output two vectors: E_W and E_P .

7.2 Test Data

7.2.1 Inputs

Table 6: Energy Tests Input

Test	Input
energy 1	$T = [40 : 44; 40 : 44]$ params
energy 2	$T = [44.2 : 0.1 : 44.6; 44.2, 44.2, 44.2, 44.2, 44.2; 372000 : 51000 : 576000]$ params
energy 3	$T = [45 : 49; 45 : 49]$ params

7.2.2 Expected Outputs

Table 7: Energy Tests Output

Test	E_W	E_P
energy 1	0	0
	627795.0938	88616
	1255590.188	177232
	1883385.281	265848
	2511180.375	354464
energy 2	2636739.394	744187.2
	2699518.903	795187.2
	2762298.413	846187.2
	2825077.922	897187.2
	2887857.431	948187.2
energy 3	3138975.469	11117682.8
	3766770.563	11231977.3
	4394565.657	11346271.8
	5022360.75	11460566.3
	5650155.844	11574860.8

8 Event Modules

8.1 Module Information

This testing suite will test the two Event modules, which handle the switch between the cases $T_P < T_{\text{melt}}$, $T_P = T_{\text{melt}}$ and $T_P > T_{\text{melt}}$.

8.1.1 Module Inputs

The Event modules take as input:

- a time t ;
- a temperature vector T where $T(1) = T_W$, $T(2) = T_P$ and (event 2 only) $T(3) = Q_P$;
- an input parameters structures containing the typical simulator parameters

8.1.2 Module Outputs

The Event modules output three values: [value, isterminal, direction].

8.1.3 Related Modules

The module loadparams.m was used in testing in order to load the input parameters structure into each function.

8.2 Test Data

8.2.1 Inputs

Table 8: Event Tests Input

Test	Input
event 1a	$t = 100$ $T = [41, 40.9]$ params
event 1b	$t = 3000$ $T = [44.2, 44.2]$ params
event 2a	$t = 3000$ $T = [44.2, 44.2, 0]$ params
event 2b	$t = 4000$ $T = [45, 44.2, 600000]$ params
event 2c	$t = 20570$ $T = [44.7, 44.2, 10654060]$ params

8.2.2 Expected Outputs

Table 9: Event Tests Expected Output

Test	Output
event 1a	[3.3, 1, 0]
event 1b	[0, 1, 0]
event 2a	[-1, 1, 0]
event 2b	[-0.943683441, 1, 0]
event 2c	[0, 1, 0]

9 Appendix

Table 10: Standard Input Variables

Var	Typical Value
L	1.5 m
D	0.412 m
V_P	0.05 m ³
A_P	1.2 m ²
ρ_P	1007 kg/m ³
T_{melt}^P	44.2 °C
C_P^S	1760 J/(kg °C)
C_P^L	2270 J/(kg °C)
H_f	211600 J/kg
A_C	0.12 m ²
T_C	50 °C
ρ_W	1000 kg/m ³
C_W	4186 J/(kg °C)
h_C	1000 W/(m ² °C)
h_P	1000 W/(m ² °C)
T_{init}	40 °C
t_{final}	50000 s
AbsTol	10 ⁻¹⁰
RelTol	10 ⁻¹⁰

Table 12: Data Constraints on Input

Var	Physical Constraints
L	$L > 0$
D	$D > 0$
V_P	$V_P > 0$ (*) $V_P < \pi(D/2)^2 L$
A_P	$A_P > 0$ (*)
ρ_P	$\rho_P > 0$
T_{melt}^P	$0 < T_{\text{melt}}^P < T_C$
C_P^S	$C_P^S > 0$
C_P^L	$C_P^L > 0$
H_f	$H_f > 0$
A_C	$A_C > 0$ (*)
T_C	$T_C > 0$ (+)
ρ_W	$\rho_W > 0$
C_W	$C_W > 0$
h_C	$h_C > 0$
h_P	$h_P > 0$
T_{init}	$0 < T_{\text{init}} < T_{\text{melt}}$ (+)
t_{final}	$t_{\text{final}} > 0$

(*) These quantities cannot be equal to zero, or there will be a divide by zero in the model.

(+) These quantities cannot be zero, or there would be freezing (A??).