SpecSearch: System Verification and Validation Test Report

Robert E. White

December 9, 2018

1 Revision History

Date	Version	Notes
2018-12-07	1.0	First draft of SystVnV test report.
2018-12-09	1.1	Creation of final draft for final documentation.

2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test
R	Requirement
NFR	Non-functional Requirement

Refer to the SRS Symbols, Abbreviations and Acronyms for a more complete list (White (2018)) https://github.com/whitere123/CAS741_REW.

Contents

List of Tables

List of Figures

This document will briefly summarize the tests outlined in the System Verification and Validation plan, explain the automated testing set-up, summarize the results of the tests and list the changes made in response to the tests. After reading this document one should be able to determine whether or not SpecSearch satisfied its functional and nonfunctional requirements with respect to the system tests. They should also be able to trace a test to a particular module and a module to a particular requirement.

3 Functional Requirements Evaluation

The three functional requirements for SpecSearch are Rin, Rfind and Rplt. More details on these requirements can be found in the System Requirements Specification, White (2018). There are six tests in the System Verification and Validation plan (https://github.com/whitere123/CAS741_REW) that cover functional requirements: test-Rin1NonNumeric, test-Rin1kBounds, test-Rin1NBounds, test-Rin1PBounds, test-Rfind-Rin and test-Rplt. The first five aforementioned tests are automated tests that check for exceptions and count the spectrum size. Test-Rplt is a visual analysis of the six spectral plots by my thesis supervisor. A summary of this test is presented at the end of this section.
A binary decision will be used to evaluate the tests. Each of the automated tests loop through a list of input and return a pass or fail message for each input. These automated tests and a summary of their results are presented in 8. A test will be considered a failure if it returns at least one fail message. In this case the code will have to be modified until non of the tests return a fail message. The evaluation of test-Rplt is dependant on the standards of my supervisor. The code will have to be modified if he does not consider the plots to be adequate enough for his research. The evaluation of test-Rplt is given in the table below. A list of passing input (inputs that do not throw exceptions) was given to my supervisor. He ran the software with these input and gave a check mark if the numerical plot was of the correct form (ie appropriate symmetry and clusting of points). A final column full of check marks will indicate a passing test for test-Rplt.

Input ID	k	N	P	Result
I1	0.6	100	2	✓
I2	0.1	120	2	✓
I3	0.9	500	2	✓
I4	0.88	550	2	✓
I5	0.99	200	2	✓
I6	0.65	700	2	✓
I7	0.4	100	2	✓
I8	0.8	400	4	✓
I9	0.9	500	4	✓
I10	0.2	700	4	✓
I11	0.3	200	4	✓
I13	0.8	100	4	✓
I14	0.89	150	4	✓
I15	0.69	500	4	✓
I16	0.55	300	2	✓
I17	0.9	400	4	✓

Table 1: test-Rplt

4 Nonfunctional Requirements Evaluation

The two Nonfunctional requirements for SpecSearch are NFR1, which is related to maintainability and manageability, and NFR2, which is related to accuracy. More details on these requirements can be found in the System Requirements Specification, White (2018). Although not explicitly mentioned in the SRS, usabilty will also be tested. It is related to maintainability as my supervisor should be able to use the code after we have completed the project.

The three Nonfunctional requirement tests are outlined in the System Verification and Validation plan (https://github.com/whitere123/CAS741_REW). They are test-NFR1, test-NFR2 and test-UserPerformance. Usability will recieve a mark out of 15. Each question is weighted according to importance by my supervisor. A question will not recieve marks if I was unable to answer it or if the response was below satisfactory. A failing grade on the usability survey is less than or equal to 11.

test-NFR2 was a code inspection that checked for maintainability and manageability. My supervisor inspected the code to see that each equation was implemented successfully and to check if he was able understand the module structure. Upon inspection my supervisor

deemed the code adequate, maintainable and manageable.

4.1 Usability

The following usability table was proposed in the System Verification and Validation Plan. I have filled out my responses under each question in the following list. Questions with poor responses were addressed in 7. Due to time constraints and other commitments I was the only person to fill out this survey.

- How long did it take before you could run the software? How many attempts at running SpecSeach did it take before you understood how to properly use it and interpret the output? (1 marks)
  This question is not applicable to me as I was the person who created the software. The response would be biased.

- Was this program useful for your research and were you able to interpret the results? (4 marks)
  Yes, this program was helpful for my research. The spectral plot helped me to foreshadow the next stage of my research and understand the context of the spectrum.
  Yes, I was able to interpret the results. The range of the axes was appropriate and matlab has a convienient interface for analyzing figures. In particular it allows you to click on particular points and display their values.

- What aspects of this software do you feel need improvement? (1 mark)
  The system performs adequately with respect to the functional requirements and within reasonable time.

- How does this program compare with other software that finds this particular spectrum? (2 mark)
  I could not find another software package that performs the same task as SpecSearch.

- Was it clear how and where to input the variables? (3 marks)
  Yes, the command line prompt gave very clear instructions.

- Were the plots clear? (4 marks)
  Yes, the choice of markers was appropriate and it was easy to differentiate between the different plots.

Based on the previously described metric, usability gets $\frac{12}{15}$. This test is related to testUserPerformance.

4.2 Checklist

The following checklist was proposed in the System Verification and Validation Plan. This checklist is related to accuracy and usability. The evaluation metric is simple: the total

number of checks divided by 5, the total number of elements in the checklist. A pass is a perfect score.

- Non of the results surprised you. ✓
- Were you able to identify which numerical algorithm and wave solution each plot represented? ✓
- Were all of the plots legible? ✓
- Was the output useful for your research? ✓
- Was it clear how to input the variables? ✓

The evaluation for this test is 1.

4.3 Inspection by Supervisor

The code inspection by my supervisor is amongst the most simple and important nonfunctional test. The metric is binary. Either my supervisor says that the code satisfies the nonfunctional requirements or doesn't. The code passed the inspection by my supervisor.

5 Comparison to Existing Implementation

My supervisor and I could not find any software that performs the same tasks as SpecSearch.

6 Unit Testing

This section is not appropriate for the System Verification and Validation Plan. For more details about unit testing please refer to the Unit VnV Plan and Unit VnV Report at (https://github.com/whitere123/CAS741_REW).

7 Changes Due to Testing

All of the tests passed. No changes were made to the software package.

8 Automated Testing

Each test involved cycling through a list of intial data and comparing the outputed result with the expected output.The following tables summarize the results for test-Rin1NonNumeric, test-Rin1kBounds, test-Rin1NBounds, test-Rin1PBounds and test-Rfind-Rin. Details about the Input, expected result and test result are given in the tables. A check mark indicates that

the test returned the expected result. Descriptions of each test can be found in the SystVn-VPlan at https://github.com/whitere123/CAS741_REW. A test is considered a failure if at least one expected value does not match the output for one intial data configuration (one of the rows).

Input ID	k	N	P	Expected Result	Test Result
I1	X	X	X	Exception	✓
I2	X	30	2	Exception	✓
I3	X	X	2	Exception	✓
I4	X	100	X	Exception	✓
I5	0.5	X	2	Exception	✓
I6	0.8	X	X	Exception	✓
I7	X	X	4	Exception	✓
I8	X	1000	X	Exception	✓
I9	0.99	X	X	Exception	✓
I10	0.6	250	X	Exception	✓

Table 2: test-Rin1NonNumeric

Input ID	k	N	P	Expected Result	Test Result
I1	1	100	2	Error	✓
I2	0	100	4	Error	✓
I3	-10	50	2	Error	✓
I4	-55	200	4	Error	✓
I5	2	150	2	Error	✓
I6	1.0001	100	2	Error	✓
I7	-0.001	200	4	Error	✓
I8	1.01	200	2	Error	✓
I9	-0.01	100	4	Error	✓
I10	0.7	-10	2	Error	✓
I11	0.9	20	4	Error	✓
I12	0.9	5	2	Error	✓
I13	0.7	0	4	Error	✓
I14	0.5	-1	2	Error	✓
I15	0.9	100	0	Error	✓
I16	0.9	500	-2	Error	✓
I17	0.7	150	-4	Error	✓
I18	0.5	200	100	Error	✓
I19	0.7	0.5	100	Error	✓

Table 3: test-RinkBounds, test-RinNBounds,test-Rin1PBounds

Input ID	k	N	P	Expected Result	Test result
I1	0.6	100	2	4N	✓
I2	0.1	120	2	4N	✓
I3	0.9	500	2	4N	✓
I4	0.88	550	2	4N	✓
I5	0.99	200	2	4N	✓
I6	0.65	700	2	4N	✓
I7	0.4	100	2	4N	✓
I8	0.8	400	4	4N	✓
I9	0.9	500	4	4N	✓
I10	0.2	700	4	4N	✓
I11	0.3	200	4	4N	✓
I13	0.8	100	4	4N	✓
I14	0.89	150	4	4N	✓
I15	0.69	500	4	4N	✓
I16	0.55	300	2	4N	✓
I17	0.9	400	4	4N	✓

Table 4: test-Rin-Rfind

9 Trace to Requirements

	Rin	Rfind	Rplt	NFR1	NFR2
test-Rin1NonNumeric	X				
test-Rin1kBounds	X				
test-Rin1NBounds	X				
test-Rin1PBounds	X				
test-RFind-Rin	X	X			
test-Rplt			X		
test-NFR1				X	
test-NFR2					X
test-UserPerformance					X

Table 5: Traceability Between Test

10 Trace to Modules

Test Case	Modules
test-Rin1NonNumeric	M1, M2, M13
test-Rin1kBounds	M1, M2, M13
test-Rin1NBounds	M1, M2, M13
test-Rin1PBounds	M1, M2, M13
test-RFind-Rin	M1, M2, M13, M4, M6, M7, M8, M9, M10,M11,M14
Rfind	M4, M6, M7, M8, M9, M10,M11, M13, M14
test-Rplt	M3, M11, M13
test-NFR1	M1-M14
test-NFR2	M1-M14
test-UserPerformance	M1-M14

Table 6: Trace Between Test Cases and Modules

11 Code Coverage Metrics

No code covering metrics were used in the creation of SpecSearch.

References

Robert White. *System Requirements Specification*. 2018.