System Test Report: Lattice Boltzmann Solver

Peter Michalski

December 17, 2019

1 Revision History

Date	Version	Notes
Dec. 16	1.0	Initial Document

2 Symbols, Abbreviations and Acronyms

Please see Section 2.2 and Section 2.3 of the Commonality Analysis (Michalski (a)).

Contents

List of Tables

List of Figures

This document reports the results of the tests found in the System VnV Plan (Michalski (c)).

3 Functional Requirements Evaluation

Functional requirements are evaluated using system tests of id1A to id17 of the System VnV Plan (Michalski (c)). Tests id11 to id17 are not covered in this report as they deal with a problem that is not implemented in the first stage of implementation of Lattice Boltzmann Solver. The results of tests id1A to id10 can be found in Section 6. Traceability of the tests of this document to functional requirements is noted in Section 9.

4 Nonfunctional Requirements Evaluation

4.1 Maintainability

This test will be conducted in January 2020.

4.2 Performance

System test id19 (performance-test-id19) found in Section 5.2.3 of the System VnV Plan (Michalski (c) compares the running time of each of the two problems, Von Karman Vortex Street and Poiseuille Flow, against the psuedo-oracle pyLBM using pyCharm IDE. The Poiseuille Flow problem will be tested in the second implementation of Lattice Boltzmann Solver. The test result for Von Karman Vortex Street is found below:

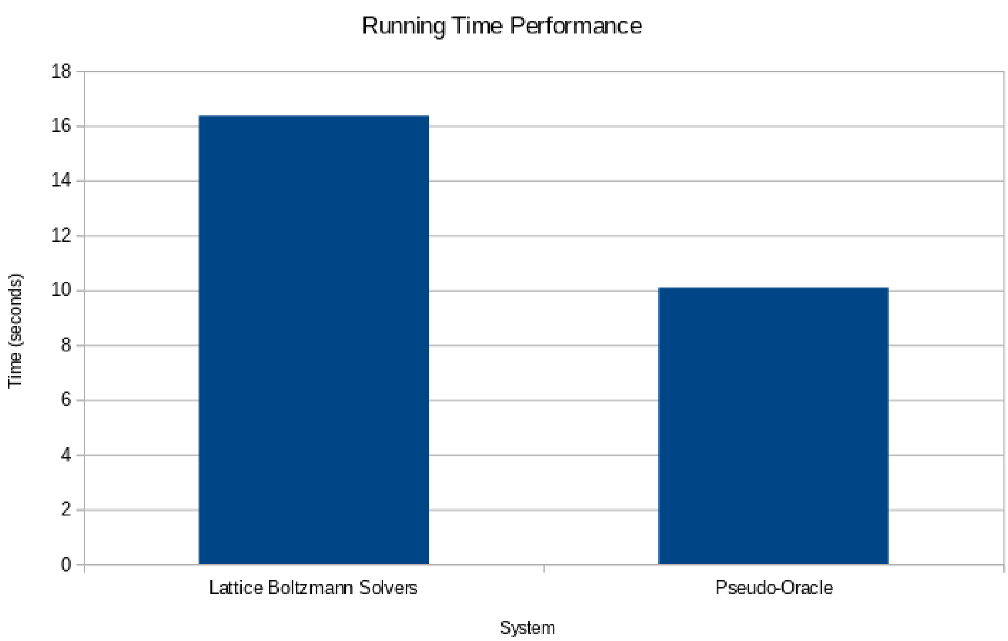

Figure 1: Running Time Performance of Lattice Boltzmann Solver vs Pseudo-Oracle

As we can see from Figure 1, the computational time of Lattice Boltzmann Solver is considerably longer (63%) than that of the pseudo-oracle. This can be attributed to the increased overhead of Lattice Boltzmann Solver, which is designed for scalability with an increased number of libraries solving an increasing number of problems.

4.3 Usability

This test will be conducted in January 2020.

5 Comparison to Existing Implementation

The first stage of implementation will not incorporate the Poiseuille Flow problem. Thus, Section 5.1.3 of the System Vnv Plan (Michalski (c)) is not reflected in this document.

6 Unit Testing

6.1 Input

6.1.1 input-reading-id1A

System test id1A found in Section 5.1.1 of the System VnV Plan (Michalski (c) verifies if the system correctly reads input from the input.txt file.

Output of test: "{'Library': 1.0, 'Problem': 1.0, 'ReynoldsNumber': 500.0, 'Density': 1.0, 'BulkViscosity': 0.001, 'Size': 1.0, 'Velocity': 1.0, 'SpatialStep': 64.0, 'Dimensions': 2.0, 'VelocityDirections': 9.0, 'DomainLength': 2.0, 'DomainWidth': 1.0}"

This output matches the input of input.txt, verifying that the system is reading the input file correctly.

6.1.2 input-reading-id1B

System test id1B found in Section 5.1.1 of the System VnV Plan (Michalski (c) verifies if the system correctly identifies a situation where the input.txt file is improperly formatted.

Output of test: An error message of “Could not read the input file.” was printed to log_file.log.

This output verified that the system handled the situation correctly.

6.1.3 input-bounds-id2A

System test id2A found in Section 2 of the System VnV Plan (Michalski (c) verifies if the system correctly identifies a situation where the input file has values that are above the allowable upper limit.

Output of test: The following error messages of printed to log_file.log:
“The input.txt file parameter ReynoldsNumber is out of bounds. Please see the User Guide.”
“The input.txt file parameter Density is out of bounds. Please see the User Guide.”
“The input.txt file parameter BulkViscosity is out of bounds. Please see the User Guide.”
“The input.txt file parameter ShearViscosity is out of bounds. Please see the User Guide.”
“The input.txt file parameter Time is out of bounds. Please see the User Guide.”

This output confirms that the module correctly identifies a situation where an input file has values that are above the allowable upper limit.

6.1.4 input-bounds-id2B

System test id2B found in Section 2 of the System VnV Plan (Michalski (c) verifies if the system correctly identifies a situation where the input file has values that are below the allowable lower limit.

Output of test: The following error messages of printed to log_file.log:
“The input.txt file parameter ReynoldsNumber is out of bounds. Please see the User Guide.”
“The input.txt file parameter Density is out of bounds. Please see the User Guide.”

"The input.txt file parameter BulkViscosity is out of bounds. Please see the User Guide."
"The input.txt file parameter ShearViscosity is out of bounds. Please see the User Guide."
"The input.txt file parameter Time is out of bounds. Please see the User Guide."

This output confirms that the module correctly identifies a situation where an input file has values that are below the allowable lower limit.

6.2 Von Karman Vortex Street

6.2.1 tutorial-test-id3

System test id3 found in Section 5.1.2 of the System VnV Plan (Michalski (c) verifies if the output vorticity vector values differ between Lattice Boltzmann Solver and the pseudo-oracle under mid-range input conditions.

Output of test found in file id3-result.txt

Compared to the expected output the vorticity vector values were identical, confirming that the output vorticity vector values between Lattice Boltzmann Solver and the pseudo-oracle, under mid-range input conditions, are the same.

6.2.2 Reynolds-rel-error-test-id4

System test id4 found in Section 5.1.2 of the System VnV Plan (Michalski (c) verifies if the output vorticity vector values differ between Lattice Boltzmann Solver and the pseudo-oracle under various Reynold Number values

Output of tests are found in files id4_1_result.txt, id4_2_result.txt, id4_3_result.txt, id4_4_result.txt, id4_5_result.txt

Compared to the expected output the vorticity vector values were identical, confirming that the output vorticity vector values between Lattice Boltzmann Solver and the pseudo-oracle, under various Reynold Number values, are the same.

6.2.3 laminar-test-id5

System test id5 found in Section 5.1.2 of the System VnV Plan (Michalski (c) verifies if the output vorticity vector values differ between Lattice Boltzmann Solver and the pseudo-oracle under a condition of laminar flow.

Output of test found in file id5-result.txt

Compared to the expected output the vorticity vector values were identical, confirming that the output vorticity vector values between Lattice Boltzmann Solver and the pseudo-oracle, under a condition of laminar flow, are the same.

6.2.4 turbulent-test-id6

System test id6 found in Section 5.1.2 of the System VnV Plan (Michalski (c) verifies if the output vorticity vector values differ between Lattice Boltzmann Solver and the pseudo-oracle under a condition of turbulent flow.

Output of test found in file id6-result.txt

Compared to the expected output the vorticity vector values were identical, confirming that the output vorticity vector values between Lattice Boltzmann Solver and the pseudo-oracle, under a condition of turbulent flow, are the same.

6.2.5 low-density-test-id7

System test id7 found in Section 5.1.2 of the System VnV Plan (Michalski (c) verifies if the output vorticity vector values differ between Lattice Boltzmann Solver and the pseudo-oracle under a condition of low density.

Output of test found in file id7-result.txt

Compared to the expected output the vorticity vector values were identical, confirming that the output vorticity vector values between Lattice Boltzmann Solver and the pseudo-oracle, under a condition of low density, are the same.

6.2.6 high-density-test-id8

System test id8 found in Section 5.1.2 of the System VnV Plan (Michalski (c) verifies if the output vorticity vector values differ between Lattice Boltzmann Solver and the pseudo-oracle under a condition of high density.

Output of test found in file id8-result.txt

Compared to the expected output the vorticity vector values were identical, confirming that the output vorticity vector values between Lattice Boltzmann Solver and the pseudo-oracle, under a condition of high density, are the same.

6.2.7 low-bulk-viscosity-test-id9

System test id9 found in Section 5.1.2 of the System VnV Plan (Michalski (c) verifies if the output vorticity vector values differ between Lattice Boltzmann Solver and the pseudo-oracle under a condition of low bulk viscosity.

Output of test found in file id9-result.txt

Compared to the expected output the vorticity vector values were identical, confirming that the output vorticity vector values between Lattice Boltzmann Solver and the pseudo-oracle, under a condition of low bulk viscosity, are the same.

6.2.8 high-bulk-viscosity-test-id10

System test id10 found in Section 5.1.2 of the System VnV Plan (Michalski (c) verifies if the output vorticity vector values differ between Lattice Boltzmann Solver and the pseudo-oracle under a condition of high bulk viscosity.

Output of test found in file id10-result.txt

Compared to the expected output the vorticity vector values were identical, confirming that the output vorticity vector values between Lattice Boltzmann Solver and the pseudo-oracle, under a condition of high bulk viscosity, are the same.

6.3 Poiseuille Flow

These tests will be conducted in January 2020 after the second stage of implementation.

7 Changes Due to Testing

No changes will be necessary.

8 Automated Testing

The System VnV Plan (Michalski (c)) specifies which unit tests were to be automated. Time constraints have resulted in manual testing of the tests reported in this document.

9 Trace to Requirements

A complete description of requirements is found in the CA (Michalski (a)). A traceability of system tests to functional requirements can be found in Table 3 in Section 5.3 of the System VnV Plan (Michalski (c)). A traceability of system tests to NFRs can be found in Table 4 of the same section.

10 Trace to Modules

A complete description of modules is found in the MG (Michalski (b)).

Cases / Modules	1	2	3	4	5	6	7	8	9	10	11	12	13
id1A	✓	✓	✓									✓	
id1B	✓	✓	✓										
id2A	✓	✓	✓	✓								✓	✓
id2B	✓	✓	✓	✓								✓	✓
id3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
id4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
id5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
id6	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
id7	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
id8	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
id9	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
id10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
id11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	N/A	✓	✓
id12	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	N/A	✓	✓
id13	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	N/A	✓	✓
id14	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	N/A	✓	✓
id15	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	N/A	✓	✓
id16	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	N/A	✓	✓
id17	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	N/A	✓	✓
id18	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
id19	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
id20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Traceability Matrix Showing the Connections Between Test Cases and Modules

11 Code Coverage Metrics

Module coverage is guaranteed for those modules that are implemented and not outsourced to external libraries. Module coverage is outline in Table 1 of Section 10.

References

Peter Michalski. Lattice Boltzmann Solvers - CA, a. URL https://github.com/peter-michalski/LatticeBoltzmannSolvers/blob/master/docs/SRS/CA.pdf.

Peter Michalski. Module Guide for Lattice Boltzmann Solvers , b. URL https://github.com/peter-michalski/LatticeBoltzmannSolvers/blob/master/docs/Design/MG/MG.pdf.

Peter Michalski. System Verification and Validation Plan for Lattice Boltzmann Solver, c. URL https://github.com/peter-michalski/LatticeBoltzmannSolvers/blob/master/docs/VnVPlan/SystemVnVPlan/SystemVnVPlan.pdf.