System Verification and Validation Plan for Lattice Boltzmann Solver

Peter Michalski

December 1, 2019

1 Revision History

Date	Version	Notes
Oct. 28, 2019	1.0	Initial Document
Nov. 27, 2019	2.0	Initial Issues Resolved

Contents

List of Tables

List of Figures

2 Symbols, Abbreviations and Acronyms

2.1 Abbreviations and Acronyms

symbol	description
1D	One Dimensional
2D	Two Dimensional
A	Assumption
C	Pseudo-Oracle Control Value
CA	Commonality Analysis
I	Individual Parameter Input Values
IDE	Integrated Development Environment
LBM	Lattice Boltzmann Method
LBS	Lattice Boltzmann Solvers
MG	Module Guide
MIS	Module Interface Specification
MTBF	Mean Time Between Failures
NAN	Not A Number
NFR	Non-Functional Requirement
O	Program Output
OTS	Off The Shelf
P	Program
R	Functional Requirement
SRS	Software Requirement Specification
V	Set of All Inputs
VnV	Verification and Validation

2.2 Symbols

symbol	description
D	Signifies the Dimension Component of Lattice Model
D_l	Length of the Domain
D_w	Width of the Domain
$\mathrm{e_{fac}}$	Velocity of Scheme
$\mathrm{e_{max}}$	Max Velocity in the Middle of Channel
$\mathrm{e_{sch}}$	Velocity Slowdown Factor
η_b	Bulk Viscosity
η_s	Shear Viscosity
$\mathbb{N}$	Natural Numbers
Q	Signifies Number of Velocity Directions of Lattice Model
R	Radius of Cylinder
Re	Reynolds Number
ρ	Density
S	Spatial Step - Number of Spatial Subsections
t	Time
X_{max}	Highest X-Axis Interval of Boundary
X_{min}	Lowest X-Axis Interval of Boundary
Y_{max}	Highest Y-Axis Interval of Boundary
Y_{min}	Lowest Y-Axis Interval of Boundary

This document covers the system Verification and Validation (VnV) of Lattice Boltzmann Solver. Functional and non-functional requirements, as found in Section 3.2 and retrieved from the Commonality Analysis (CA) (Michalski [3]), will be tested. The document will outline the general information of the system in Section 3, which will be followed by a testing plan in Section 4. Finally, individual tests will be described in Section 5.

3 General Information

This section of the document will begin with a summary of Lattice Boltzmann Solver, objectives and a listing of relevant documents.

3.1 Summary

The software being tested is Lattice Boltzmann Solver. This software reads inputs from a file, and calculates outputs using Lattice Boltzmann Methods (LBM). The software outputs the results to the screen, to a file, and/or to memory.
Lattice Boltzmann Solver can model many fluid dynamics problems, including Poiseuille Flow and Von Karman Vortex Street. Both of these problems will be used in our functional test cases, as the results of Lattice Boltzmann Solver can be compared to the results of the pseudo-oracle pyLBM (pyl [1]).

3.2 Objectives

The intended objective of this plan is to verify that functional requirements and non-functional requirements (NFR), as found in Section 7.1 and Section 7.2 of the CA titled Lattice Boltzmann Solvers (Michalski [3]), have been met.

3.3 Relevant Documentation

The CA of Lattice Boltzmann Solver can be found in (Michalski [3]).The link to Section 1 navigates to the document, beginning with the revision history. The MG can be found in (Michalski [4]). The link to Section 1 navigates to the document, beginning with the revision history. The MIS can be found in

(Michalski [5]). The link to Section 1 navigates to the document, beginning with the revision history.

4 Plan

4.1 Verification and Validation Team

This VnV plan will be conducted by Peter Michalski and several classmates, including the primary reviewer Ao Dong, and secondary reviewers Sasha Soraine, Bo Cao, and Zhi Zhang.

4.2 CA Verification Plan

The CA of Lattice Boltzmann Solver shall be verified in the following ways:

1. Feedback: Classmates, including all primary and secondary reviewers listed above, shall provide feedback on GitHub. They shall read the document and provide insight on how to improve the documents and Lattice Boltzmann Solver.

2. Initial Review: The document shall be manually reviewed by the author using the SRS checklist upon its initial creation, as found in the CAS741 GitLab repository (Smith [8]).

3. Second Review: The document shall be manually reviewed by the author using the SRS checklist after VnV completion, as found in the CAS741 GitLab repository (Smith [8]).

4. Final Review: The document shall be manually reviewed by the author using the SRS checklist after MG and MIS development, as found in the CAS741 repository (Smith [8]).

4.3 Design Verification Plan

The design shall be verified by ensuring that functional requirements and NFRs are tested, as listed in Section 3.2.
The system functional requirements shall be tested first, as outlined in 5.1. This will be followed by reviewing the NFRs as outlined in 5.2.

4.4 Implementation Verification Plan

The implementation shall be verified in the following ways:

1. Code Walkthrough - Author: Module unit code shall be inspected for functional errors by the author immediately after MIS document version 1.0 is developed. A rubber duck debugging method will be followed. Any defects shall be immediately fixed. This plan is described in Section 5.1 of the document Unit Verification and Validation Plan for Lattice Boltzmann Solver (Michalski [6]).

2. Code Walkthrough - Peer: Module unit code shall be inspected for functional errors by Ao Dong after MIS document version 1.0 is submitted on GitHub. Defects shall be noted in GitHub using issue creation. This plan is described in Section 5.1 of the document Unit Verification and Validation Plan for Lattice Boltzmann Solver (Michalski [6]).

3. Dynamic Unit Tests - Author: Dynamic unit test will be carried out, as listed in Section 5.1 of the document Unit Verification and Validation Plan for Lattice Boltzmann Solver (Michalski [6]).

4. System Tests: System tests will be carried out as listed in Section 5. The author shall conduct the tests of the functional requirements. Tests of the NFRs shall be conducted by those listed in the outline of each test.

4.5 Software Validation Plan

Lattice Boltzmann Solver does not have a validation step. Validation is related to comparing the outputs of models to experimental values. Lattice Boltzmann Solver is simply concerned that its solution is correct mathematically.

5 System Test Description

5.1 Tests for Functional Requirements

The subsections below are designed to cover several of the functional requirements of the system. Subsection 5.1.1 will cover R1 and R3. Subsections 5.1.2 and 5.1.3 will cover R1, R2, R3, R4, R5, R6, and R7, and will test out the input vales at their allowable edges.

5.1.1 Input

Input Reading

1. input-reading-id1A

   Control: Program (P) reading of input values (V) from a file.

   Initial State: input.txt file is in the Input directory.

   Input: A comma delimited text file, with inputs marked in the manner outlined in the User Guide (Michalski [7]).

   The input values for this test will be:
   Library: 1
   Problem: 1
   Re: 500
   t: 75
   ρ: 1.0
   η_b: 1.e-3
   $X_{min} = 0.$
   $X_{max} = 3.$
   $Y_{min} = 0.$
   $Y_{max} = 1.$
   $R = 0.05$
   $\mathrm{e_{sch}} = 1.$
   $\mathrm{e_{fac}} = 20$
   $\mathrm{e_{max}} = 0.1$
   $S = 64$
   $\mathrm{D} = 2$
   $\mathrm{Q} = 9$
   $D_l = 2$

$D_w = 1$

Output: All inputs printed to the screen as output (O).
Outputs printed to the screen:
Library: 1
Problem: 1
Re: 500
t: 75
ρ: 1.0
η_b: 1.e-3
$X_{min} = 0.$
$X_{max} = 3.$
$Y_{min} = 0.$
$Y_{max} = 1.$
$R = 0.05$
$\mathrm{e_{sch}} = 1.$
$\mathrm{e_{fac}} = 20$
$\mathrm{e_{max}} = 0.1$
$S = 64$
D = 2
Q = 9
$D_l = 2$
$D_w = 1$

Test Case Derivation: P(V) is correct if P:V -> O and O = V

The result required for the test to pass is the successful printing of all input values to the screen. This test satisfies R1.

How test will be performed:

(a) Outside of the system, the input parameter values will be written to a comma delimited text file titled input.txt, as outlined in the User Guide.

(b) The file will be placed into the Input directory, under the home directory of the project.

(c) Lattice Boltzmann Solver will be run.

(d) If successful, The system will output the input parameters to the screen.

2. input-reading-id1B

   Control: Program (P) reading of input values (V) from a file.

   Initial State: input.txt file is in the Input directory.

   Input: A comma delimited text file with the following input: ccQseHfFspBLAYZjCGZtJYMAdeAc

   Output: An error message of "cannot read file" should be shown.

   Test Case Derivation: P(V) is correct if P:V -> Error Message

   The result required for the test to pass is the successful printing of the error message. This test satisfies R1.

   How test will be performed:

   (a) Outside of the system, the input parameter value will be written to a comma delimited text file titled input.txt, as outlined in the User Guide.

   (b) The file will be placed into the Input directory, under the home directory of the project.

   (c) Lattice Boltzmann Solver will be run.

   (d) If successful, the system will output an error message to the screen.

Input Bounds This test will be composed of two parts:

1. input-bounds-id2A

2. input-bounds-id2B

Control: Program (P) testing of high and low input values (V) read from a file.

Initial State: input.txt file is in the Input directory.

Input: A comma delimited file, with inputs marked in the manner outlined in the User Guide (Michalski [7]).
The input values for this test will be:

Output: A descriptive failure message printed to the screen as seen below, or on the command line depending on how the system is run. In the below figure, X will contain all parameter names that are out of bounds. In both tests that will include Re, t, ρ, η_b, and η_s.

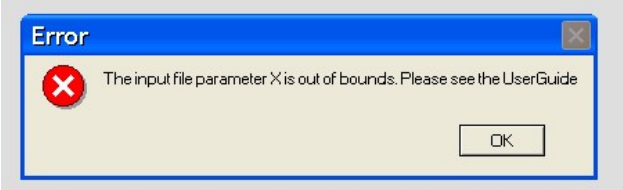

Figure 1: Input out of Bounds

Test Case Derivation: I (individual inputs) are an element of V. P(V) is incorrect if P:V and at least one I is out of bounds as per Table 5: Input Data Bounds. This test satisfies R3.

How test will be performed:

(a) Outside of the system, the input values will be written to a comma delimited text file titled input.txt, as outlined in the User Guide (Michalski [7]).

(b) The file will be placed into the Input directory, under the home directory of the project.

Input	id2A	id2B
Library	1	1
Problem	1	1
Re	50000	-500000
t	0	0
ρ	15	-15
η_b	100	-100
η_s	30000	-30000
X_{min}	0.	0.
X_{max}	3.	3.
Y_{min}	0.	0.
Y_{max}	1.	1.
R	0.05	0.05
$\mathrm{e_{sch}}$	1.	1.
$\mathrm{e_{fac}}$	20	20
$\mathrm{e_{max}}$	0.1	0.1
S	64	64
D	2	2
Q	9	9
D_l	2	2
D_w	1	1

Table 1: Input Bounds Test Inputs

(c) Lattice Boltzmann Solver will be run.

(d) If the test is successful, the system will output a descriptive error message, as seen above, to the screen.

5.1.2 Von Karman Vortex Street

Dynamic Testing - Automated:

A Von Karman vortex street is a fluid dynamics problem involving a repeating pattern of vortices caused by flow separation around a physical obstacle (von [2]).

These dynamic tests will compare Lattice Boltzmann Solver output with output from the pseudo-oracle myLBM (pyl [1]). The pseudo-oracle will have the same inputs as Lattice Boltzmann Solver for each test. The difference in output between Lattice Boltzmann Solver and the pseudo-oracle will be compared and noted.

1. tutorial-test-id3

   Control: Program (P) module execution using inputs (V), and expecting output (O) to match a control value (C).

   Initial State: input.txt file is in the Input directory.

   Input: A comma delimited file, with inputs marked in the manner outlined in the User Guide (Michalski [7]).
   The input values for this test will be:
   Library: 1
   Problem: 1
   Re: 500
   t: 75
   ρ: 1.0
   η_b: 1.e-3
   $X_{min} = 0.$
   $X_{max} = 3.$
   $Y_{min} = 0.$
   $Y_{max} = 1.$
   $R = 0.05$
   $\mathrm{e_{sch}} = 1.$
   $\mathrm{e_{fac}} = 20$
   $S = 64$

D = 2
Q = 9

Output: Vorticity vector values printed to the screen.

Test Case Derivation: This case is a comparison with the pseudo-oracle pyLBM. The output vector values (C) of this test for pyLBM can be found in the file id3output.txt located in the OracleOutput folder. The output values of Lattice Boltzmann Solver will be compared to the pseudo-oracle output values.

How test will be performed:

(a) The Von Karman Vortex Street module shall be modified by the author to print the vorticity vector as output.

(b) Outside of the system, the input parameter values will be written to a comma delimited text file titled input.txt, as outlined in the User Guide.

(c) The file will be placed into the Input directory, under the home directory of the project.

(d) The module for Von Karman Vortex Street will be selected to run.

(e) Upon completion of the module, the output values of the vorticity vector will be compared to the vorticity vector values from pyLBM - comparison will be done per cell. Comparisons can be done manually using Excel, or through a script, using the equation for relative error found in Section 6.3.

2. Reynolds-Rel-Error-test-id4

Control: Program (P) module execution using inputs (V) with varying *Re* parameter (I), and observing the variance of the output (O) when compared to a control value (C).

Initial State: input.txt file is in the Input directory.

Input: A comma delimited file, with inputs marked in the manner outlined in the User Guide (Michalski [7]).

The input values for each of the iterations of this test will be:

Input	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
Library	1	1	1	1	1
Problem	1	1	1	1	1
Re	500	1500	2500	5000	7500
t	75	75	75	75	75
ρ	1.0	1.0	1.0	1.0	1.0
η_b	1.e-3	1.e-3	1.e-3	1.e-3	1.e-3
X_{min}	0.	0.	0.	0.	0.
X_{max}	3.	3.	3.	3.	3.
Y_{min}	0.	0.	0.	0.	0.
Y_{max}	1.	1.	1.	1.	1.
R	0.05	0.05	0.05	0.05	0.05
$\mathrm{e_{sch}}$	1.	1.	1.	1.	1.
$\mathrm{e_{fac}}$	20	20	20	20	20
S	64	64	64	64	64
D	2	2	2	2	2
Q	9	9	9	9	9

Table 2: Reynolds Test Inputs

Output: In each iteration we will have vorticity vector values printed to the screen.

Test Case Derivation: This case is a comparison with the pseudo-oracle pyLBM. The C values of this test for pyLBM can be found in the files id4_1_output.txt, id4_2_output.txt, id4_3_output.txt, id4_4_output.txt, and id4_5_output.txt located in the OracleOutput folder. The files hold the C values for iterations 1 to 5 respectively. Each iteration will compare its output to its respective C counterpart. The intent is to discover how changes in Re affect correctness. This may be helpful in improving

correctness of the implementation.

How test will be performed:

(a) The Von Karman Vortex Street module shall be modified by the author to print the vorticity vector as output.

(b) Outside of the system, the input parameter values will be written to a comma delimited text file titled input.txt, as outlined in the User Guide.

(c) The file will be placed into the Input directory, under the home directory of the project.

(d) The module for Von Karman Vortex Street will be selected to run.

(e) Upon completion of the module, the output values of the vorticity vector will be compared to the vorticity vector values from pyLBM - comparison will be done per cell. Comparisons can be done manually using Excel, or through a script.

(f) Steps (a) - (e) will be repeated for each test iteration.

(g) The output value of each iteration will be compared to the pseudo-oracle value.

3. laminar-test-id5

Control: Program (P) module execution using inputs (V), and expecting output (O) to match a control value (C).

Initial State: input.txt file is in the Input directory.

A comma delimited file, with inputs marked in the manner outlined in the User Guide (Michalski [7]).
The input values for this test will be:
Library: 1
Problem: 1
Re: 0.0001
t: 75
ρ: 1.0

η_b: 1.e-3
$X_{min} = 0.$
$X_{max} = 3.$
$Y_{min} = 0.$
$Y_{max} = 1.$
$R = 0.05$
$\mathrm{e_{sch}} = 1.$
$\mathrm{e_{fac}} = 20$
$S = 64$
D = 2
Q = 9

Output: Vorticity vector values printed to the screen.

Test Case Derivation: This case is a comparison with the pseudo-oracle pyLBM. The C values of this test for pyLBM can be found in the file id5output.txt located in the OracleOutput folder. The output values of Lattice Boltzmann Solver will be compared to the pseudo-oracle output values.

This test covers a very small Reynolds number representing laminar flow.

How test will be performed:

(a) The Von Karman Vortex Street module shall be modified by the author to print the vorticity vector as output.

(b) Outside of the system, the input parameter values will be written to a comma delimited text file titled input.txt, as outlined in the User Guide.

(c) The file will be placed into the Input directory, under the home directory of the project.

(d) The module for Von Karman Vortex Street will be selected to run.

(e) Upon completion of the module, the output values of the vorticity vector will be compared to the vorticity vector values from pyLBM

- comparison will be done per cell. Comparisons can be done manually using Excel, or through a script, using the equation for relative error found in Section 6.3.

4. turbulent-test-id6

   Control: Program (P) module execution using inputs (V), and expecting output (O) to match a control value (C).

   Initial State: input.txt file is in the Input directory.

   A comma delimited file, with inputs marked in the manner outlined in the User Guide (Michalski [7]).
   The input values for this test will be:
   Library: 1
   Problem: 1
   Re: 50000
   t: 75
   ρ: 1.0
   η_b: 1.e-3
   $X_{min} = 0.$
   $X_{max} = 3.$
   $Y_{min} = 0.$
   $Y_{max} = 1.$
   $R = 0.05$
   $\mathrm{e_{sch}} = 1.$
   $\mathrm{e_{fac}} = 20$
   $S = 64$
   $\mathrm{D} = 2$
   $\mathrm{Q} = 9$

   Output: Vorticity vector values printed to the screen.

   Test Case Derivation: This case is a comparison with the pseudo-oracle pyLBM. The C values of this test for pyLBM can be found in the file id6output.txt located in the OracleOutput folder. The output values of

Lattice Boltzmann Solver will be compared to the pseudo-oracle output values.

This test covers a very large Reynolds number representing very turbulent flow.

How test will be performed:

(a) The Von Karman Vortex Street module shall be modified by the author to print the vorticity vector as output.

(b) Outside of the system, the input parameter values will be written to a comma delimited text file titled input.txt, as outlined in the User Guide.

(c) The file will be placed into the Input directory, under the home directory of the project.

(d) The module for Von Karman Vortex Street will be selected to run.

(e) Upon completion of the module, the output values of the vorticity vector will be compared to the vorticity vector values from pyLBM - comparison will be done per cell. Comparisons can be done manually using Excel, or through a script, using the equation for relative error found in Section 6.3.

5. low-density-test-id7

Control: Program (P) module execution using inputs (V), and expecting output (O) to match a control value (C).

Initial State: input.txt file is in the Input directory.

A comma delimited file, with inputs marked in the manner outlined in the User Guide (Michalski [7]).
The input values for this test will be:
Library: 1
Problem: 1
Re: 500
t: 75

ρ: 7.08e-2
η_b: 1.e-3
$X_{min} = 0.$
$X_{max} = 3.$
$Y_{min} = 0.$
$Y_{max} = 1.$
$R = 0.05$
$\mathrm{e_{sch}} = 1.$
$\mathrm{e_{fac}} = 20$
$S = 64$
D = 2
Q = 9

Output: Vorticity vector values printed to the screen.

Test Case Derivation: This case is a comparison with the pseudo-oracle pyLBM. The C values of this test for pyLBM can be found in the file id7output.txt located in the OracleOutput folder. The output values of Lattice Boltzmann Solver will be compared to the pseudo-oracle output values.

This test covers the low bound for density using a density of liquid hydrogen.

How test will be performed:

(a) The Von Karman Vortex Street module shall be modified by the author to print the vorticity vector as output.

(b) Outside of the system, the input parameter values will be written to a comma delimited text file titled input.txt, as outlined in the User Guide.

(c) The file will be placed into the Input directory, under the home directory of the project.

(d) The module for Von Karman Vortex Street will be selected to run.

(e) Upon completion of the module, the output values of the vorticity vector will be compared to the vorticity vector values from pyLBM

- comparison will be done per cell. Comparisons can be done manually using Excel, or through a script, using the equation for relative error found in Section 6.3.

6. high-density-test-id8

   Control: Program (P) module execution using inputs (V), and expecting output (O) to match a control value (C).

   Initial State: input.txt file is in the Input directory.

   A comma delimited file, with inputs marked in the manner outlined in the User Guide (Michalski [7]).
   The input values for this test will be:
   Library: 1
   Problem: 1
   Re: 500
   t: 75
   ρ: 13.6
   η_b: 1.e-3
   $X_{min} = 0.$
   $X_{max} = 3.$
   $Y_{min} = 0.$
   $Y_{max} = 1.$
   $R = 0.05$
   $\mathrm{e_{sch}} = 1.$
   $\mathrm{e_{fac}} = 20$
   $S = 64$
   $\mathrm{D} = 2$
   $\mathrm{Q} = 9$

   Output: Vorticity vector values printed to the screen.

   Test Case Derivation: This case is a comparison with the pseudo-oracle pyLBM. The C values of this test for pyLBM can be found in the file id8output.txt located in the OracleOutput folder. The output values of

Lattice Boltzmann Solver will be compared to the pseudo-oracle output values.

This test covers the high bound for density using a density close to that of mercury.

How test will be performed:

(a) The Von Karman Vortex Street module shall be modified by the author to print the vorticity vector as output.

(b) Outside of the system, the input parameter values will be written to a comma delimited text file titled input.txt, as outlined in the User Guide.

(c) The file will be placed into the Input directory, under the home directory of the project.

(d) The module for Von Karman Vortex Street will be selected to run.

(e) Upon completion of the module, the output values of the vorticity vector will be compared to the vorticity vector values from pyLBM - comparison will be done per cell. Comparisons can be done manually using Excel, or through a script, using the equation for relative error found in Section 6.3.

7. low-bulk-viscosity-test-id9

Control: Program (P) module execution using inputs (V), and expecting output (O) to match a control value (C).

Initial State: input.txt file is in the Input directory.

A comma delimited file, with inputs marked in the manner outlined in the User Guide (Michalski [7]).
The input values for this test will be:
Library: 1
Problem: 1
Re: 500

t: 75
ρ: 1.0
η_b: 1.e-4
$X_{min} = 0.$
$X_{max} = 3.$
$Y_{min} = 0.$
$Y_{max} = 1.$
$R = 0.05$
$\mathrm{e_{sch}} = 1.$
$\mathrm{e_{fac}} = 20$
$S = 64$
D = 2
Q = 9

Output: Vorticity vector values printed to the screen.

Test Case Derivation: This case is a comparison with the pseudo-oracle pyLBM. The C values of this test for pyLBM can be found in the file id9output.txt located in the OracleOutput folder. The output values of Lattice Boltzmann Solver will be compared to the pseudo-oracle output values.
This test covers the low bound for bulk viscosity.

How test will be performed:

(a) The Von Karman Vortex Street module shall be modified by the author to print the vorticity vector as output.

(b) Outside of the system, the input parameter values will be written to a comma delimited text file titled input.txt, as outlined in the User Guide.

(c) The file will be placed into the Input directory, under the home directory of the project.

(d) The module for Von Karman Vortex Street will be selected to run.

(e) Upon completion of the module, the output values of the vorticity vector will be compared to the vorticity vector values from pyLBM - comparison will be done per cell. Comparisons can be done

manually using Excel, or through a script, using the equation for relative error found in Section 6.3.

8. high-bulk-viscosity-test-id10

   Control: Program (P) module execution using inputs (V), and expecting output (O) to match a control value (C).

   Initial State: input.txt file is in the Input directory.

   A comma delimited file, with inputs marked in the manner outlined in the User Guide (Michalski [7]).
   The input values for this test will be:
   Library: 1
   Problem: 1
   Re: 500
   t: 75
   ρ: 1.0
   η_b: 1.e-2
   $X_{min} = 0.$
   $X_{max} = 3.$
   $Y_{min} = 0.$
   $Y_{max} = 1.$
   $R = 0.05$
   $\mathrm{e_{sch}} = 1.$
   $\mathrm{e_{fac}} = 20$
   $S = 64$
   $\mathrm{D} = 2$
   $\mathrm{Q} = 9$

   Output: Vorticity vector values printed to the screen.

   Test Case Derivation: This case is a comparison with the pseudo-oracle pyLBM. The C values of this test for pyLBM can be found in the file id10output.txt located in the OracleOutput folder. The output values of Lattice Boltzmann Solver will be compared to the pseudo-oracle out-

put values.

This test covers the high bound for bulk viscosity.

How test will be performed:

(a) The Von Karman Vortex Street module shall be modified by the author to print the vorticity vector as output.

(b) Outside of the system, the input parameter values will be written to a comma delimited text file titled input.txt, as outlined in the User Guide.

(c) The file will be placed into the Input directory, under the home directory of the project.

(d) The module for Von Karman Vortex Street will be selected to run.

(e) Upon completion of the module, the output values of the vorticity vector will be compared to the vorticity vector values from pyLBM - comparison will be done per cell. Comparisons can be done manually using Excel or through a script, using the equation for relative error found in Section 6.3.

5.1.3 Poiseuille Flow

Dynamic Testing - Automated:

Poiseuille flow is the movement of a fluid through a circular tube.

These dynamic tests will compare Lattice Boltzmann Solver output with output from the pseudo-oracle myLBM (pyl [1]). The input oracle will have the same inputs as Lattice Boltzmann Solver for each test.

1. tutorial-test-id11

   Control: Program (P) module execution using inputs (V), and expecting output (O) to match a control value (C).

   Initial State: input.txt file is in the Input directory.

   A comma delimited file, with inputs marked in the manner outlined in the User Guide (Michalski [7]).
   The input values for this test will be:
   Library: 1
   Problem: 2
   t: 50
   ρ: 1
   η_b: 1.e-2
   η_s: 1.e-2
   $S = 16$
   $D_l = 2$
   $D_w = 1$
   $\mathrm{e_{sch}} = 1.$
   $\mathrm{e_{max}} = 0.1$
   $X_{min} = 0.$

   Output: Pressure gradient value printed to the screen.

   Test Case Derivation: This case is a comparison with the pseudo-oracle pyLBM, which has a module for Poiseuille Flow. The output pressure

gradient value of this test is to be compared to the C value from the pseudo-oracle, -7.074e-03.

How test will be performed:

(a) Outside of the system, the input parameter values will be written to a comma delimited text file titled input.txt, as outlined in the User Guide.

(b) The file will be placed into the Input directory, under the home directory of the project.

(c) The module for Poiseuille Flow will be selected to run.

(d) Upon completion of the module, the pressure gradient output value will be compared to the above output value from the pseudo-oracle.

2. low-density-test-id12

Control: Program (P) module execution using inputs (V), and expecting output (O) to match a control value (C).

Initial State: input.txt file is in the Input directory.

A comma delimited file, with inputs marked in the manner outlined in the User Guide (Michalski [7]).
The input values for this test will be:
Library: 1
Problem: 2
t: 50
ρ: 0.0708
η_b: 1.e-2
η_s: 1.e-2
$S = 16$
$D_l = 2$
$D_w = 1$
$\mathrm{e_{sch}} = 1.$
$\mathrm{e_{max}} = 0.1$

$X_{min} = 0$.

Output: Pressure gradient value printed to the screen.

Test Case Derivation: This case is a comparison with the pseudo-oracle pyLBM, which has a module for Poiseuille Flow. The output pressure gradient value of this test is to be compared to the C value from the pseudo-oracle, -4.103e-03.

How test will be performed:

(a) Outside of the system, the input parameter values will be written to a comma delimited text file titled input.txt, as outlined in the User Guide.

(b) The file will be placed into the Input directory, under the home directory of the project.

(c) The module for Poiseuille Flow will be selected to run.

(d) Upon completion of the module, the pressure gradient output value will be compared to the above output value from the pseudo-oracle.

3. high-density-test-id13

Control: Program (P) module execution using inputs (V), and expecting output (O) to match a control value (C).

Initial State: input.txt file is in the Input directory.

A comma delimited file, with inputs marked in the manner outlined in the User Guide (Michalski [7]).
The input values for this test will be:
Library: 1
Problem: 2
t: 50
ρ: 13.6

η_b: 1.e-2
η_s: 1.e-2
$S = 16$
$D_l = 2$
$D_w = 1$
$\mathrm{e_{sch}} = 1.$
$\mathrm{e_{max}} = 0.1$
$X_{min} = 0.$

Output: Pressure gradient value printed to the screen.

Test Case Derivation: This case is a comparison with the pseudo-oracle pyLBM, which has a module for Poiseuille Flow. The output pressure gradient value of this test is to be compared to the C value from the pseudo-oracle, -1.522e-01.

How test will be performed:

(a) Outside of the system, the input parameter values will be written to a comma delimited text file titled input.txt, as outlined in the User Guide.

(b) The file will be placed into the Input directory, under the home directory of the project.

(c) The module for Poiseuille Flow will be selected to run.

(d) Upon completion of the module, the pressure gradient output value will be compared to the above output value from the pseudo-oracle.

4. low-bulk-viscosity-test-id14

Control: Program (P) module execution using inputs (V), and expecting output (O) to match a control value (C).

Initial State: input.txt file is in the Input directory.

A comma delimited file, with inputs marked in the manner outlined in the User Guide (Michalski [7]).
The input values for this test will be:
Library: 1
Problem: 2
t: 50
ρ: 1
η_b: 1.e-4
η_s: 1.e-2
$S = 16$
$D_l = 2$
$D_w = 1$
$\mathrm{e_{sch}} = 1.$
$\mathrm{e_{max}} = 0.1$
$X_{min} = 0.$

Output: Pressure gradient value printed to the screen.

Test Case Derivation: This case is a comparison with the pseudo-oracle pyLBM, which has a module for Poiseuille Flow. The output pressure gradient value of this test is to be compared to the C output from the pseudo-oracle, -1.462e-01.

How test will be performed:

(a) Outside of the system, the input parameter values will be written to a comma delimited text file titled input.txt, as outlined in the User Guide.

(b) The file will be placed into the Input directory, under the home directory of the project.

(c) The module for Poiseuille Flow will be selected to run.

(d) Upon completion of the module, the pressure gradient output value will be compared to the above output value from the pseudo-oracle.

5. high-bulk-viscosity-test-id15

Control: Program (P) module execution using inputs (V), and expecting output (O) to match a control value (C).

Initial State: input.txt file is in the Input directory.

A comma delimited file, with inputs marked in the manner outlined in the User Guide (Michalski [7]).
The input values for this test will be:
Library: 1
Problem: 2
t: 50
ρ: 1
η_b: 20000
η_s: 1.e-2
$S = 16$
$D_l = 2$
$D_w = 1$
$\mathrm{e_{sch}} = 1.$
$\mathrm{e_{max}} = 0.1$
$X_{min} = 0.$

Output: Pressure gradient value printed to the screen.

Test Case Derivation: This case is a comparison with the pseudo-oracle pyLBM, which has a module for Poiseuille Flow. The output pressure gradient value of this test is to be compared to the C value from the pseudo-oracle, -3.262e-03.

How test will be performed:

(a) Outside of the system, the input parameter values will be written to a comma delimited text file titled input.txt, as outlined in the User Guide.

(b) The file will be placed into the Input directory, under the home directory of the project.

(c) The module for Poiseuille Flow will be selected to run.

(d) Upon completion of the module, the pressure gradient output value will be compared to the above output value from the pseudo-oracle.

6. low-shear-viscosity-test-id16

Control: Program (P) module execution using inputs (V), and expecting output (O) to match a control value (C).

Initial State: input.txt file is in the Input directory.

A comma delimited file, with inputs marked in the manner outlined in the User Guide (Michalski [7]).
The input values for this test will be:
Library: 1
Problem: 2
t: 50
ρ: 1
η_b: 1.e-2
η_s: 1.e-3
$S = 16$
$D_l = 2$
$D_w = 1$
$\mathrm{e_{sch}} = 1.$
$\mathrm{e_{max}} = 0.1$
$X_{min} = 0.$

Output: Pressure gradient value printed to the screen.

Test Case Derivation: This case is a comparison with the pseudo-oracle pyLBM, which has a module for Poiseuille Flow. The output pressure gradient value of this test is to be compared to the C value from the pseudo-oracle, -1.604e-03.

How test will be performed:

(a) Outside of the system, the input parameter values will be written to a comma delimited text file titled input.txt, as outlined in the User Guide.

(b) The file will be placed into the Input directory, under the home directory of the project.

(c) The module for Poiseuille Flow will be selected to run.

(d) Upon completion of the module, the pressure gradient output value will be compared to the above output value from the pseudo-oracle.

7. high-shear-viscosity-test-id17

Control: Program (P) module execution using inputs (V), and expecting output (O) to match a control value (C).

Initial State: input.txt file is in the Input directory.

A comma delimited file, with inputs marked in the manner outlined in the User Guide (Michalski [7]).
The input values for this test will be:
Library: 1
Problem: 2
t: 50
ρ: 1
η_b: 1.e-2
η_s: 20000
$S = 16$
$D_l = 2$
$D_w = 1$
$\mathrm{e_{sch}} = 1.$
$\mathrm{e_{max}} = 0.1$
$X_{min} = 0.$

Output: Pressure gradient value printed to the screen.

Test Case Derivation: This case is a comparison with the pseudo-oracle pyLBM, which has a module for Poiseuille Flow. The output pressure gradient value of this test is to be compared to the C value from the pseudo-oracle, -6.868e-01.

How test will be performed:

(a) Outside of the system, the input parameter values will be written to a comma delimited text file titled input.txt, as outlined in the User Guide.

(b) The file will be placed into the Input directory, under the home directory of the project.

(c) The module for Poiseuille Flow will be selected to run.

(d) Upon completion of the module, the pressure gradient output value will be compared to the above output value from the pseudo-oracle.

5.2 Tests for Nonfunctional Requirements

5.2.1 Correctness

System correctness is tested via the functional tests of Section 5.1.

5.2.2 Maintainability

Document Walkthrough

1. maintainability-test-id18

   Type: Maintainability Walkthrough

   Initial State: Maintainability of the repository has not been tested.

   Input/Condition: A production version of Lattice Boltzmann Solver has been released.

   Output/Result: A graded report describing the maintainability of the repository.

   Test Case Derivation: This test will determine if the system documentation allows for easy maintainability by checking how much documentation is available, as well as reviewing its clarity and traceability. A score of higher than 3 in each of the tested categories will indicate that the system documentation is maintainable.

   How test will be performed:

   (a) Ao Dong shall check the repository for the following documentation: SRS/CA, VnV Plan, MG, MIS, User Guide.

   (b) Ao Dong shall mark 1 point for each of the above documents.

   (c) Ao Dong shall read through each of the above documents and provide a grade between 1 and 5 for clarity of the writing. A score of 1 represents a document that is hard to understand, and a score of 5 represents a document that is easy to understand. The user shall divide the sum of the scores for all of the reports by 5.

(d) Ao Dong shall read through each of the above documents and provide a grade between 1 and 5 for traceability within the document. A score of 1 represents no links within the report, and a score of 5 represents many links between sections of the report. The user shall then divide the sum of the scores for all of the reports by 5.

(e) Ao Dong will generate a brief report with the above three values, from steps (c) to (e).

5.2.3 Performance

Running Time

1. performance-test-id19

   Type: Running Time - Automated Test

   Initial State: Input file is in the appropriate directory. In the first test the file contains the required parameter values for Von Karman Vortex Street and in the second test it contains the required parameter values for Poiseuille Flow.

   Input/Condition of first test: A comma delimited file with the following input parameters and values:
   Library: 1
   Problem: 1
   Re: 500
   t: 200
   ρ: 1.0
   η_b: 1.e-3
   $X_{min} = 0.$
   $X_{max} = 3.$
   $Y_{min} = 0.$
   $Y_{max} = 1.$
   $R = 0.05$
   $\mathrm{e_{sch}} = 1.$
   $\mathrm{e_{fac}} = 20$

$S = 64$
D = 2
Q = 9

Input/Condition of second test: A comma delimited file with the following input parameters and values:
Library: 1
Problem: 2
Re: 500
t: 200
ρ: 1.0
η_b: 1.e-3
$X_{min} = 0.$
$X_{max} = 3.$
$Y_{min} = 0.$
$Y_{max} = 1.$
$R = 0.05$
$\mathrm{e_{sch}} = 1.$
$\mathrm{e_{fac}} = 20$
$S = 64$
D = 2
Q = 9

Output/Result: A timed comparison of running each of the two system modules, Von Karman Vortex Street and Poiseuille Flow, against the psuedo-oracle pyLBM using pyCharm IDE. The percentage difference in running time between Lattice Boltzmann Solver and the pseudo-oracle will noted and analyzed.

How test will be performed:

(a) Outside of the system, the input parameter values will be written, by the author, to a comma delimited text file titled input.txt, as outlined in the User Guide.

(b) The file will be placed into the Input directory, under the home directory of the project.

(c) The system shall start a timer when the Poiseuille Flow module is selected to run.

(d) Upon completion of the module the author shall note the module running time, which will be printed by the system.

(e) The author shall repeat steps (a) to (d) for the Von Karman Vortex Street module.

(f) The author shall then run each of the problems, Poiseuille Flow and Von Karman Vortex Street, using the pseudo-oracle pyLBM using pyCharm IDE, making sure to capture the running time of each problem.

(g) The author shall compare the running time of each of the two system modules, Von Karman Vortex Street and Poiseuille Flow, against the pseudo-oracle.

(h) The system environment will be kept as constant as reasonably possible in order to prevent timing errors. Identical background programs will be running between tests of the modules and the pseudo-oracle.

5.2.4 Portability

Portability is tested by performing the functional tests of Section 5.1.2 and Section 5.1.3 on three operating systems (virtual machines can be used for this): macOS, Windows, Linux. Test output will be compared and recorded.

5.2.5 Robustness

System robustness is tested through the Input Bounds test in Section 5.1.

5.2.6 Usability

Usability Survey

1. usability-test-id20

   Type: Usability Survey

Initial State: The Lattice Boltzmann Solver repository, the below input parameters, and an instruction to run the Von Karman Vortex Street module are provided to three anonymous users via GitHub.

Input/Condition: A usability survey with the questions listed in Section 6.1 and a comma delimited file with the following input parameters and values:
Library: 1
Problem: 1
Re: 500
t: 75
ρ: 1.0
η_b: 1.e-3
$X_{min} = 0.$
$X_{max} = 3.$
$Y_{min} = 0.$
$Y_{max} = 1.$
$R = 0.05$
$\mathrm{e_{sch}} = 1.$
$\mathrm{e_{fac}} = 20$
$S = 64$
$\mathrm{D} = 2$
$\mathrm{Q} = 9$

Output/Result: Survey results.

Test Case Derivation: This case is a check of the usability of Lattice Boltzmann Solver. Respondents will be asked to rank their experience of running a module. A final average grade of 3 will indicate that the users found the system to have average usability. The higher the score, the better the perception of usability.

How test will be performed:

(a) Each participant will be instructed to run a Von Karman Vortex Street simulation using the provided input parameters.

(b) The participants will attempt to run the simulation.

(c) Upon completion of the attempt, the participants will be asked to answer the survey questions, found in SubSection 6.1.

(d) The questionnaire results will be tallied and averaged.

5.3 Traceability Between Test Cases and Requirements

	R1	R2	R3	R4	R5	R6	R7
id1A	✓						
id1B	✓						
id2A			✓				
id2B			✓				
id3	✓	✓	✓	✓	✓	✓	✓
id4	✓	✓	✓	✓	✓	✓	✓
id5	✓	✓	✓	✓	✓	✓	✓
id6	✓	✓	✓	✓	✓	✓	✓
id7	✓	✓	✓	✓	✓	✓	✓
id8	✓	✓	✓	✓	✓	✓	✓
id9	✓	✓	✓	✓	✓	✓	✓
id10	✓	✓	✓	✓	✓	✓	✓
id11	✓	✓	✓	✓	✓	✓	✓
id12	✓	✓	✓	✓	✓	✓	✓
id13	✓	✓	✓	✓	✓	✓	✓
id14	✓	✓	✓	✓	✓	✓	✓
id15	✓	✓	✓	✓	✓	✓	✓
id16	✓	✓	✓	✓	✓	✓	✓
id17	✓	✓	✓	✓	✓	✓	✓
id18							
id19	✓	✓	✓	✓	✓	✓	✓
id20	✓	✓	✓	✓	✓	✓	✓

Table 3: Traceability Matrix Showing the Connections Between Test Cases and Functional Requirements

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

Table 4: Traceability Matrix Showing the Connections Between Test Cases and NFRs

References

[1] pyLBM. URL https://github.com/pylbm/pylbm.

[2] Von Karman vortex street. URL https://pylbm.readthedocs.io/en/latest/notebooks/07_Von_Karman_vortex_street.html.

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

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

[5] Peter Michalski. Module Interface Specification for Lattice Boltzmann Solvers, . URL https://github.com/peter-michalski/LatticeBoltzmannSolvers/blob/master/docs/Design/MIS/MIS.pdf.

[6] Peter Michalski. Lattice Boltzmann Solvers - Unit VnV, . URL https://github.com/peter-michalski/LatticeBoltzmannSolvers/blob/master/docs/VnVPlan/UnitVnVPlan/UnitVnVPlan.pdf.

[7] Peter Michalski. Lattice Boltzmann Solvers - User Guide, . URL https://github.com/peter-michalski/LatticeBoltzmannSolvers/tree/master/docs/UserGuide.

[8] Smith. SRS Checklist. URL https://gitlab.cas.mcmaster.ca/smiths/cas741/blob/master/BlankProjectTemplate/docs/SRS/SRS-Checklist.pdf.

6 Appendix

6.1 Usability Survey Questions

Using the following rubric please rate the five statements found below:

1 - strongly disagree
2 - somewhat disagree
3 - neither agree nor disagree
4 - somewhat agree
5 - strongly agree

1. The formatting of the input file was easy to understand.
2. The location to place the input file was easy to find.
3. Navigating to the correct module was straightforward.
4. The User Guide of this product explained the modules well.
5. I would recommend this product.

6.2 Input Data

variability	value
Re	0.0001 - 50000
ρ	0.0708 - 13.6
η_b	0.0001 - 0.01
η_s	0.001 - 20000
t	$\mathbb{N}$
X_{min}	0.
X_{max}	3.
Y_{min}	0.
Y_{max}	1.
R	0.05
$\mathrm{e_{sch}}$	1.
$\mathrm{e_{fac}}$	20
$\mathrm{e_{max}}$	0.1
S	64 or 16 (module specific)
D	2
Q	9
D_l	2
D_w	1

Table 5: Input Data Bounds

The upper and lower bounds of Re, ρ, η_b and η_s are derived from the allowable input range of the pseudo-oracle pyLBM. Input values outside of these bounds are known to sometimes return NAN module results. t can be any $\mathbb{N}$. Fixed value variables of tests in this testing report are: X_{min}, X_{max}, Y_{min}, Y_{max}, R, D_l, D_w, $\mathrm{e_{sch}}$, $\mathrm{e_{fac}}$, $\mathrm{e_{max}}$, S (module specific), D, and Q.

6.3 Mathematical Equation for Error

For the purpose of testing, the equation for error is:

$$\% \text{ relative error} = \frac{|LatticeBoltzmannSolver\ output\ -\ pseudo{-}oracle\ output|}{pseudo{-}oracle\ output}$$