

Lattice Boltzmann Solvers

Peter Michalski

October 6, 2019

1 Revision History

Date	Version	Notes
October 7,2019	1.0	Initial Document

2 Reference Material

This section records information for easy reference.

2.1 Table of Units

Throughout this document SI (Système International d’Unités) is employed as the unit system. In addition to the basic units, several derived units are used as described below. For each unit, the symbol is given followed by a description of the unit and the SI name.

symbol	unit	SI
m	length	metre
kg	mass	kilogram
t	time	second
F	force	newton
cm	length	centimetre
g	mass	gram

2.2 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	unit	description
e	$\frac{m}{s}$	velocity
η	$Pa - s$	viscosity
A	m^2	cross-sectional area
γ	$\frac{1}{s}$	velocity gradient
τ	N/A	relaxation rate
x	N/A	position vector
f	N/A	distribution function
Ω	N/A	collision operator
f^{eq}	N/A	equilibrium distribution function
k	N/A	velocity direction
p	$\frac{g}{cm^3}$	fluid density
w	N/A	weight coefficient (implementation specific)

u	$\frac{m}{s}$	macroscopic velocity of fluid
D	N/A	signifies the dimension component of lattice model
Q	N/A	signifies number of velocity directions of lattice model
σ	N/A	variable number of dimensions in the lattice model
κ	N/A	variable number of velocity directions of lattice model (linkages)
\mathbb{R}	N/A	real numbers
c_k	NA	unit vector along the lattice streaming direction
c_s	$\frac{m}{s}$	speed of sound

2.3 Abbreviations and Acronyms

symbol	description
1D	1-Dimensional
2D	2-Dimensional
3D	3-Dimensional
A	Assumption
CA	Commonality Analysis
DD	Data Definition
GS	Goal Statement
LBM	Lattice Boltzmann Methods
LBS	Lattice Boltzmann Solvers
LC	Likely Change
MPI	Message Passing Interface
OTS	Off The Shelf (LBS Solutions)
R	Requirement
T	Theoretical Model

Contents

1	Revision History	i
2	Reference Material	ii
2.1	Table of Units	ii
2.2	Table of Symbols	ii
2.3	Abbreviations and Acronyms	iii
3	Introduction	2
3.1	Purpose of Document	2
3.2	Scope of the Family	2
3.3	Characteristics of Intended Reader	2
3.4	Organization of Document	2
4	General System Description	3
4.1	Potential System Contexts	3
4.2	Potential User Characteristics	3
4.3	Potential System Constraints	3
5	Commonalities	4
5.1	Background Overview	4
5.2	Terminology and Definitions	4
5.3	Data Definitions	5
5.4	Goal Statements	7
5.5	Theoretical Models	8
6	Variabilities	11
6.1	Assumptions	11
6.2	Calculation	12
6.3	Output	13
7	Requirements	14
7.1	Family of Functional Requirements	14
7.2	Nonfunctional Requirements	15
8	Likely Changes	16
9	Traceability Matrices and Graphs	17
10	Appendix	20
10.1	Symbolic Parameters	20
10.2	Off The Shelf Solutions	20
10.3	Coefficient Weights for Equilibrium Distribution Function	22

3 Introduction

This document provides a Commonality Analysis (CA) for a family of Lattice Boltzmann Solvers (LBS), which provide services based on Lattice Boltzmann Methods (LBM). LBM are a family of fluid dynamics algorithms for simulating single-phase and multiphase fluid flows, often incorporating additional physical complexities Chen and Doolen [7]. They consider the behaviours of a collection of particles as a single unit at the mesoscopic scale. These methods predict the positional probability of a collection of particles moving through a lattice structure. Off the shelf (OTS) Lattice Boltzmann Solvers (LBS) allow for a range of fluid and physical model input parameters, computational parameters, and output parameters as outlined in Section 10.2. The following subsection of this introduction will outline the purpose of this document, a general scope of the family of LBS, the characteristics of the intended reader, and finally an outline of the rest of this document.

3.1 Purpose of Document

The purpose of this document is to provide general information on the currently available LBS solutions, including their commonalities and variabilities, as well as a baseline understanding of the model and structure of abstract LBM. The information provided here will be used in the development of the design of a solution providing services of a family of LBS.

3.2 Scope of the Family

The family of LBS will model one or more fluids as they pass through a boundary, modeled by a lattice. Fluids with any properties can be modeled, however only those properties that are accepted as inputs by the LBS will affect the model results. The calculation of the LBM distribution function will use up to 3D computational models, and will output the data into memory and render it in up to 3D imaging.

3.3 Characteristics of Intended Reader

The intended reader of this document should have an undergraduate understanding of software requirements and specifications as well as software design principles. Ideally, the user will be knowledgeable of commonality analysis,

3.4 Organization of Document

This document is organized along a template for a CA for scientific computing software proposed by Smith [18]. It follows a standard pattern of presenting a general system description, commonalities, variabilities, and the requirements for a family of LBS. The goal statements of the family of LBS, found in Section 5.4, are refined to the theoretical models in Section 5.5. Variabilities within the family are found in Section 6. Tables of OTS solution commonalities and variabilities are found in Section 10.2.

4 General System Description

This section identifies the interfaces between the system and its environment, describes the potential user characteristics and lists the potential system constraints.

4.1 Potential System Contexts

- User Responsibilities:
 - The user must provide the system with correctly formatted physical model parameters.
 - The user must select the desired mathematical model for the computation.
 - The user must select the desired format of output for the model.
- Lattice Boltzmann Solvers Responsibilities:
 - Detect data type mismatch, such as a negative number instead of a positive number for a parameter, such as A that cannot accept negative values.
 - Initialize the correct data types and data structures for the model.
 - Perform the calculations to predict the distribution of fluid particles over time.
 - Store the distribution function output data.
 - Store calculated fluid parameters over time.
 - Visually model the results of the distribution function.
 - Store the calculation results in a file and/or in memory.
 - Detect errors during parameter input, model calculation, or model output; store the errors in a file and show the error to the user.
 - Recover from error states, such as those that develop from division by zero or a buffer overflow.

4.2 Potential User Characteristics

The end user of Lattice Boltzmann Solvers should ideally have an understanding of undergraduate Level 1 physics and fluid dynamics. The ideal end user characteristics may differ between the members of the family of solvers. For example, a user of HemeLB, a off the shelf LBM solution for simulating blood flow, would ideally have an understanding of phlebology.

4.3 Potential System Constraints

The parallel nature of LBS prefers operating and hardware systems that can handle concurrency and large amounts of data. Modern operating systems and computer hardware platforms are suggested. Memory should be scaled to the requirements of the desired LBS library, and decomposition technique.

5 Commonalities

5.1 Background Overview

As LBS model fluid dynamics within a boundary using a predefined lattice structure, the methods rely on a two step calculation process. The first processes is known as streaming, where the particles move along the lattice via links, and the second process is collision, where energy and momentum is transferred among particles that collide [4]. In the LBS solutions, the particles are mapped using a lattice structure. The lattice structure can be a 1D, 2D, or 3D model with varying velocity directions. The notation is $D\sigma Q\kappa$, where σ represents the number of dimensions and κ represents the number of velocity directions. There are many standardized lattice models; individual solvers within the family may only use a subset of them. The LBM uses the initial parameters of the fluid to find the probability of where along the lattice linkages a group of particles are most likely to travel. It then moves the particles into the next node, and transfers the energy and momentum if a collision occurs. Then the process repeats for the duration of the modeling instance.

5.2 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:

- **Correctness** : The degree to which a system or component is free from faults in its specification, design, and implementation IEEE Std 610.12-1990 [10].
- **Maintainability**: The ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment IEEE Std 610.12-1990 [10].
- **Performance**: The degree to which a system or component accomplishes its designated functions within given constraints, such as speed, accuracy, or memory usage IEEE Std 610.12-1990 [10].
- **Portability**: The ease with which a system or component can be transferred from one hardware or software environment to another IEEE Std 610.12-1990 [10].
- **Reliability**: The ability of a system or component to perform its required functions under stated conditions for a specified period of time IEEE Std 610.12-1990 [10].
- **Reusability**: The degree to which a software module or other work product can be used in more than one computer program or software system IEEE Std 610.12-1990 [10].
- **Robustness**: The degree to which a system or component can function correctly in the presence of invalid inputs or stressful environmental conditions IEEE Std 610.12-1990 [10].

- Scalability: The ability of the system to cope with increasing numbers of users without reducing overall QoS that is delivered to any user Sommerville [19].
- Understandability: The ease of understanding the software system Uchida and Shima [20].
- Usability: The ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component IEEE Std 610.12-1990 [10].

5.3 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given.

Number	DD1
Label	Velocity
Symbol	e
SI Units	$\frac{m}{s}$
Equation	$e = \frac{dr}{dt}$
Description	Velocity is the distance that an object moves relative to time. r is the the distance (m) in change for our change in time t of units (s).
Sources	Mohamad [13]
Ref. By	T1 T3

Number	DD2
Label	Viscosity
Symbol	η
SI Units	Pa – s
Equation	$\eta = \frac{F/A}{\gamma}$
Description	Viscosity is the measure of resistance to deformation. F is the applied force (N), A is the cross-sectional area (m^2), and γ is the velocity gradient.
Sources	vis [2]
Ref. By	DD3

Number	DD3
Label	Relaxation Rate Towards Equilibrium
Symbol	τ
SI Units	NA
Equation	$\tau = \frac{12\eta\Delta t}{\Delta x^2} + \frac{1}{2}$
Description	The relaxation rate defines how quickly the particles recover to equilibrium state. Adjusting this method in the implementation allows for the simulation of complex physical phenomena, specifically concerning the fluid media. η is the viscosity of the fluid, t is the time interval (s), and x is the position vector.
Sources	Bolton [6]
Ref. By	T1

Number	DD4
Label	Velocity Gradient
Symbol	γ
SI Units	$\frac{1}{s}$
Equation	$\gamma = \frac{de}{dz}$
Description	Velocity gradient is the difference in velocity between adjacent fluids. de represents the difference in velocities of the fluids and dz is the distance of the two velocities.
Sources	vis [2]
Ref. By	DD2

Number	DD5
Label	Fluid Density
Symbol	p
SI Units	$\frac{g}{cm^3}$
Equation	$p = \frac{g}{cm^3}$
Description	Density is the ratio of mass to volume of a material. g is the mass and cm^3 is the volume.
Sources	den [1]
Ref. By	T3

5.4 Goal Statements

Given the boundary conditions, lattice model, weighting coefficient of the lattice, simulation time, fluid particle mass, and initial conditions for the momentum, density and position of the fluid particles, as well as any applied external force, the goal statements are:

G_prob: Predict the location of fluid particles in the lattice over time.

G_velocity: Predict the velocity of fluid particles within the lattice over time.

G_fluidPressure: Predict the pressure of fluid particles within the lattice over time.

5.5 Theoretical Models

This section focuses on the general equations and laws that Lattice Boltzmann Solvers are based on.

Number	T1
Label	Boltzmann Transport Equation
Equation	$f(x + edt, e + \frac{F}{m}dt, t + dt)dxde - f(x, e, t)dxde = \Omega(f)dxde$
Description	<p>This equation determines the statistical description of a group of particles. The left part of the equation, $f(x + edt, e + \frac{F}{m}dt, t + dt)dxde$, represents the distribution function result after an external force F is applied. The middle function, $f(x, e, t)dxde$, represents the distribution function result before the external force is applied. The distribution function f represents the probability that a set of particles will be at a specific location of the lattice at a specified time. The right hand side of the equation represents the collision operator, Ω.</p> <p>The variable x represents the vector of the particles within the lattice, e is velocity $\frac{m}{s}$, t is time (s), F is force (N), kg is length (kg). This equation can be further developed for specific instances. The user shall select the desired model as stated in A5, and input the desired starting parameters as stated in A6.</p>
Source	Bolton [6] Mohamad [13]
Ref. By	

Number	T2
Label	Bhatnagar-Gross-Krook Collision Operator
Equation	$\Omega = \frac{\Delta t}{\tau}(f^{eq}(r, t) - f(r, t))$
Description	The above equation is a mathematical operator that preserves continuity for a discretized model. τ is the relaxation rate towards equilibrium and should be in the range of 0.5 - 2.0. It is related to viscosity as outlined in DD. f^{eq} is the equilibrium particle probability distribution function. f is the particle probability distribution function. This equation can be further developed for specific instances. Several fluids in the instance model can be modeled by this equation, as stated in A1. The user shall select the desired model as stated in A5, and input the desired starting parameters as stated in A6.
Source	Bolton [6] Mohamad [13]
Ref. By	T1

Number	T3
Label	Equilibrium Distribution Function
Equation	$f_k^{eq} = pw_k[1 + \frac{2\vec{c}_k\vec{u} - \vec{u}\vec{u}}{2c_s^2} + \frac{(\vec{c}_k\vec{u})^2}{2c_s^4}] + O(u^2)$
Description	<p>The above equation captures the probability distribution of the particles. Adjusting this method in the implementation allows for the simulation of complex physical phenomena, including geometry of the boundary. p is the fluid density ($\frac{g}{cm^2}$). w is the weighting coefficient for the lattice model as the fluid flows through a lattice structure A3. The weighting coefficients are standard, as per A4. k is the discretized velocity direction, referring to the directions of the chosen lattice model. c_k is the unit vector along the lattice streaming direction. u is the macroscopic velocity of the fluid, which is a vector field of velocity at a specific position and time. c_s is the speed of sound, a constant, as stated in A7. This equation can be further developed for specific instances. Several fluids in the instance model can be modeled by this equation, as stated in A1. Several fluids in the instance model can be modeled by this equation, as stated in A2, can be reflected in the velocity direction weights. The user shall select the desired model as stated in A5, and input the desired starting parameters as stated in A6.</p>
Source	Bolton [6] Mohamad [13]
Ref. By	T2

6 Variabilities

6.1 Assumptions

A1: One or more fluids can be modeled.

A2: The fluid can, but does not need to, flow through an object with boundary conditions.

A3: The fluid flows through space via a lattice structure, moving between lattice nodes via linkages (Q).

A4: Weight coefficients are standard for each lattice model. See the Table 11.

A5: The user will select the desired model prior to running the simulation.

A6: The user will enter a subset of fluid parameter inputs prior to running the simulation. See Table 2.

A7: The speed of sound is constant. It's value can be found in Section 10.1. The constant is referenced in model T3.

Goal	Model	A1	A2	A3	A4	A5	A6	A7
G1	T1					✓	✓	
G1	T2	✓				✓	✓	
G1	T3	✓	✓	✓	✓	✓	✓	✓
G2	T1					✓	✓	
G2	T2	✓				✓	✓	
G2	T3	✓	✓	✓	✓	✓	✓	✓
G3	T1					✓	✓	
G3	T2	✓				✓	✓	
G3	T3	✓				✓	✓	✓

Table 1: Assumption Relationship to Goals and Models

6.2 Calculation

Variability	Parameter of Variation	Binding Time
boundary shape	Set of {defined 2D, defined 3D, undefined}	compile
boundary parameters	Set of {deflective, non deflective}	compile
fluid parameters	Set of $\{e, t, u, p, x, \eta, \tau, \gamma, F, A\}$	compile
model choice	Set of {1D, 2D, 3D}	compile
velocity directions	Set of {2, 3, 5, 9, 13, 15, 19, 27}	scope
velocity (e)	Set of R	scope
time (t)	Set of R	scope
macroscopic velocity (u)	Set of R	scope
fluid density (p)	Set of positive R	scope
position vector (x)	Vector of Set of R	scope
viscosity (η)	Set of $R \geq 0$	scope
relaxation rate (τ)	Set of R	scope
velocity gradient (γ)	Set of R	scope
force (F)	Set of $R \geq 0$	scope
cross-sectional area (A)	Set of $R \geq 0$	scope

Table 2: Input Variabilities

Variability	Parameter of Variation	Binding Time
computational model (see Section 4.3)	D1Q2, D1Q3, D1Q5, D2Q9, D2Q13, D2Q15, D3Q15, D3Q15i, D3Q19, D3Q19+, D3Q27	compile
decomposition technique (see Section 4.3)	{ParMETIS library, PT_Scotch library, block-wide decompo- sition, domain decomposition, spinoidal decomposition}	compile
coefficient weights	Set of $\{0 < R < 1\}$; $\sum \mathbb{R} = 1$	compile
input check	boolean (false if input satisfies in- put assumptions)	compile
exception check	boolean (false if no exception con- dition raised)	compile

Table 3: Calculation Variabilities

6.3 Output

Variability	Parameter of Variation	Binding Time
graphical model	Set of $\{2D, 3D\}$; $\sum \mathbb{R} = 1$	run
destination for output	Set of $\{\text{file, screen, memory}\} = 1$	run
fluid characteristics	Set of $\{\text{wall pressure, flow velocity, fluid location}\}$	run
encoding of output	Set of $\{\text{binary, text}\}$	run

Table 4: Output Variabilities

7 Requirements

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.

7.1 Family of Functional Requirements

- R1: The user shall input a set of fluid parameters, listed in Table 2, into the system, as per A6. These parameters will be used in calculations for T1, T2, and T3.
- R2: The user shall select from a set of model and velocity direction parameters, listed in Table 2, into the system, as per A5. These models will be reflected in the calculations of T1.
- R3: The system shall verify that the inputs fall within the allowable parameters of variation, see Table 2.
- R4: The system shall instantiate required data types and structures for the selected model.
- R5: The system shall import from memory the relevant coefficient weights for the selected model, as per A4. The weighting values can be found in Table 11. These models will be reflected in the calculations of T1.
- R6: The system shall calculate and store in memory the predicted fluid parameters, iterating through streaming and collision processes over the desired model time.
- R7: The system shall output the results of the calculations to the screen, to a file, and/or to memory.

7.2 Nonfunctional Requirements

The following are non-functional requirements for the family of LBS. They are defined in Section 5.2:

1. Correctness
2. Maintainability
3. Performance
4. Portability
5. Reliability
6. Reusability
7. Robustness
8. Scalability
9. Understandability
10. Usability

The requirements have been compared using a pairwise process, the results of which are listed in Table 5 below. The comparison took into account the available documentation of current off the shelf LBS solutions, listed in Table 8, Table 9, and Table 10.

NFR/ NFR	1	2	3	4	5	6	7	8	9	10	Σ
1	-	1	1	1	1	1	1	1	1	1	9
2	0	-	0	1	0	1	1	1	0	0	4
3	0	1	-	1	0	1	0	1	1	1	6
4	0	0	0	-	0	0	0	0	0	0	0
5	0	1	1	1	-	1	1	1	1	1	8
6	0	0	0	1	0	-	0	0	0	0	1
7	0	0	1	1	0	1	-	1	1	0	5
8	0	0	0	1	0	1	0	-	0	0	2
9	0	1	0	1	0	1	0	1	-	0	4
10	0	1	0	1	0	1	1	1	1	-	6

Table 5: Pairwise Comparison of NFR

The following is a list of NFR by importance as found in the Table 5 pairwise comparison:

1. Correctness
2. Reliability
3. Performance and Usability
4. Robustness
5. Maintainability and Understandability
6. Scalability
7. Reusability
8. Portability

8 Likely Changes

LC1: A family of LBS solvers will have 2D and 3D output. 1D output is not a common variability. See Table 8.

LC2: Wall pressure is not an output variability that is often needed. This may be removed from a family of LBS. See Table 10.

LC3: Spinoidal decomposition is most common among LBS family members and should be the standard for a library implementation. See Table 9.

LC4: MPI is the standard parallel interface for LBS and should be the standard for a library implementation. See Table 9.

LC5: LBS generally read input parameters from a file and this should be the standard for a library implementation. See Table 8.

9 Traceability Matrices and Graphs

	T1	T2	T3	DD1	DD2	DD3	DD4	DD5
T1								
T2								
T3								
DD1								
DD2								
DD3								
DD4								
DD5								

Table 6: Traceability Matrix Showing the Connections Between Items of Different Sections

	A1	A2	A3	A4	A5	A6	DD7
T1							
T2							
T3							
DD1							
DD2							
DD3							
DD4							
DD5							
DD1							
DD2							
DD3							
DD4							
DD5							

Table 7: Traceability Matrix Showing the Connections Between Assumptions and Other Items

References

- [1] Density. URL <https://physics.info/density>.
- [2] Viscosity. URL <https://physics.info/viscosity>.
- [3] Aursjo. Lattice boltzmann simulations. URL folk.uio.no/olavau/LBsim.
- [4] Yuanxun Bill Bao and Justin Meskas. Lattice boltzmann method for fluid simulations. *Department of Mathematics, Courant Institute of Mathematical Sciences, New York University*, page 44, 2011.
- [5] Massimo Bernaschi, Simone Melchionna, Sauro Succi, Maria Fyta, Efthimios Kaxiras, and Joy K Sircar. Muphy: A parallel multi physics/scale code for high performance bio-fluidic simulations. *Computer Physics Communications*, 180(9):1495–1502, 2009.
- [6] Shreedharan Bolton, Schwartz. Lattice boltzmann methods. URL https://personal.ems.psu.edu/~fkd/courses/EGEE520/2017Deliverables/LBM_2017.pdf.
- [7] Shiyi Chen and Gary D Doolen. Lattice boltzmann method for fluid flows. *Annual review of fluid mechanics*, 30(1):329–364, 1998.
- [8] Fernandez. Multiphase lattice boltzmann suite user guide, 2014. URL <https://github.com/carlosrosales/mplabs/blob/master/docs/mp-labs.pdf>.
- [9] Gouarin Graille. pylbm documentation release 0.3.2, 2018. URL <https://buildmedia.readthedocs.org/media/pdf/pylbm/develop/pylbm.pdf>.
- [10] IEEE Std 610.12-1990. Ieee standard glossary of software engineering terminology. Standard, IEEE, 1991.
- [11] Michal Januszewski and Marcin Kostur. Sailfish: A flexible multi-gpu implementation of the lattice boltzmann method. *Computer Physics Communications*, 185(9):2350–2368, 2014.
- [12] Marco D Mazzeo and Peter V Coveney. Hemelb: A high performance parallel lattice-boltzmann code for large scale fluid flow in complex geometries. *Computer Physics Communications*, 178(12):894–914, 2008.
- [13] AA Mohamad. *Lattice Boltzmann Method*, volume 70. Springer, 2011.
- [14] D Arumuga Perumal and Anoop K Dass. A review on the development of lattice boltzmann computation of macro fluid flows and heat transfer. *Alexandria Engineering Journal*, 54(4):955–971, 2015.

- [15] Sebastian Schmieschek, Lev Shamardin, Stefan Frijters, Timm Krüger, Ulf D Schiller, Jens Harting, and Peter V Coveney. Lb3d: A parallel implementation of the lattice-boltzmann method for simulation of interacting amphiphilic fluids. *Computer Physics Communications*, 217:149–161, 2017.
- [16] Florian Schornbaum and Ulrich Rude. Massively parallel algorithms for the lattice boltzmann method on nonuniform grids. *SIAM Journal on Scientific Computing*, 38(2):C96–C126, 2016.
- [17] MA Seaton and W Smith. Dl meso user manual, 2016.
- [18] Spencer Smith. Systematic development of requirements documentation for general purpose scientific computing software. In *14th IEEE International Requirements Engineering Conference (RE’06)*, pages 209–218. IEEE, 2006.
- [19] Ian Sommerville. *Software Engineering 9*. Pearson Education, 2011.
- [20] Shinji Uchida and Kazuyuki Shima. An experiment of evaluating software understandability. *Journal of Systemic, Cybernetics and Informatics*, 2:7–11, 2005.

10 Appendix

10.1 Symbolic Parameters

Cons_SpeedSound: The speed of sound c_s is equal to $343 \frac{m}{s}$. The binding time for this constant is during compile time. This constant is referenced by T3, and is invoked by A7.

10.2 Off The Shelf Solutions

The following tables list some off the shelf Lattice Boltzmann Solvers, along with input parameters, computational parameters, and output parameters.

solver	velocity	density	model	velocity directions	time	viscosity	input method
hemeLB[12]	≥ 0	≥ 0	3D	15	≥ 0	≥ 0	prompt
MUPHY[5]	≥ 0	≥ 0	3D	19	≥ 0	≥ 0	file
Walberla[16]	≥ 0	≥ 0	2D/3D	19	≥ 0	≥ 0	file
DL_Meso[17]	≥ 0	≥ 0	2D/3D	9,15,19,27	≥ 0	≥ 0	file
LB3D[15]	≥ 0	≥ 0	3D	19	≥ 0	≥ 0	file
Sailfish[11]	≥ 0	≥ 0	2D/3D	9,13,15, 19,27	≥ 0		
mplabs[8]	≥ 0	≥ 0	2D/3D	9,19	≥ 0		file
LBSIM[3]	≥ 0		2D/3D	6,19	≥ 0		
pylbm[9]	≥ 0	≥ 0	1D,2D,3D	2,3,5,9, 13,15,19	≥ 0	≥ 0	file

Table 8: OTS LBS Inputs

solver	computational model	decomposition technique	parallel interface
hemeLB[12]	D3Q15i	ParMETIS library	MPI
MUPHY[5]	D3Q19+	PT_Scotch library	MPI
Walberla[16]	D2Q9, D3Q19	block-wide decomposition	MPI
DL_Meso[17]	D2Q9, D3Q15, D3Q19, D3Q27	domain decomposition	MPI
LB3D[15]	D3Q19	spinodal decomposition	MPI
Sailfish[11]	D2Q9, D3Q13, D3Q15, D3Q19, D3Q27	spinoidal decomposition	MPI
mplabs[8]	D2Q9, D3Q19		MPI
LBSIM[3]	D2Q6, D3Q19	spinoidal decomposition	
pylbm[9]	D1Q2, D1Q3, D1Q5, D2Q9, D2Q13, D2Q15, D3Q15, D3Q19		MPI

Table 9: OTS LBS Computational Parameters

solver	wall pressure	flow velocity	graphical model
hemeLB[12]	≥ 0	≥ 0	2D/3D
MUPHY[5]			2D/3D
Walberla[16]		≥ 0	2D/3D
DL_Meso[17]		≥ 0	2D/3D
LB3D[15]		≥ 0	2D/3D
Sailfish[11]		≥ 0	2D
mplabs[8]		≥ 0	2D/3D
LBSIM[3]			2D/3D
pylbm[9]			2D/3D

Table 10: OTS LBS Output Parameters

10.3 Coefficient Weights for Equilibrium Distribution Function

lattice model	coefficient weights (w_i)
D1Q2[]	
D1Q3[]	$4/6, i = 0; 1/6, i=1,2$
D1Q5[]	
D2Q9[14]	$4/9, i = 0; 1/9, i = 1,2,3,4; 1/36, i = 5,6,7,8$
D2Q13[]	
D2Q15[]	
D3Q15[14]	$2/9, i = 0; 1/9, i = 1,2,...,6; 1/72, i = 7,8,...,14$
D3Q19[14]	$2/9, i = 0; 1/18, i = 1,2,...,6; 1/36, i = 7,8,...,18$
D3Q27[14]	$8/27, i = 0; 2/27, i = 1,2,...,6;$ $1/54, i = 7,8,...,18; 1/216, i =$ $19,20,...,26$

Table 11: Lattice Model Coefficient Weights