Lattice Boltzmann Solvers

Peter Michalski

November 1, 2019

1 Revision History

Date	Version	Notes
October 7,2019	1.0	Initial Document
October 20,2019	1.1	Issues Fixed + 10.4 Added

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
s	time	second
F	force	newton
cm	length	centimetre
g	mass	gram
Pa	pressure	pascal

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
l	m	characteristic length of the system
t	s	time
f	N/A	distribution function
Ω	N/A	collision operator
f^{eq}	N/A	equilibrium distribution function

k	N/A	velocity direction
ρ	$\frac{g}{cm^3}$	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 (linkages) of lattice model
$\mathbb{R}$	N/A	real numbers
$\mathbb{N}$	N/A	natural numbers
c_k	NA	unit vector along the lattice streaming direction
c_s	$\frac{m}{s}$	speed of sound
Re	NA	Reynolds number

2.3 Abbreviations and Acronyms

symbol	description
1D	1-Dimensional
2D	2-Dimensional
3D	3-Dimensional
A	Assumption
AHP	Analytic Hierarchy Process
CA	Commonality Analysis
DD	Data Definition
GS	Goal Statement
LBM	Lattice Boltzmann Methods
LBS	Lattice Boltzmann Solvers
LC	Likely Change
MG	Module Guide
MIS	Module Interface Specification
MPI	Message Passing Interface
MTBF	Mean Time Between Failures
NFR	Non-Functional Requirement
OTS	Off The Shelf
R	Requirement
T	Theoretical Model
VnV	Verification and Validation

Contents

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 [8]). 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 subsections 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 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 1D, 2D, and 3D computational models, and will output the data into memory and render it on a screen in up to 3D imaging. The output results can also be rendered to a file.

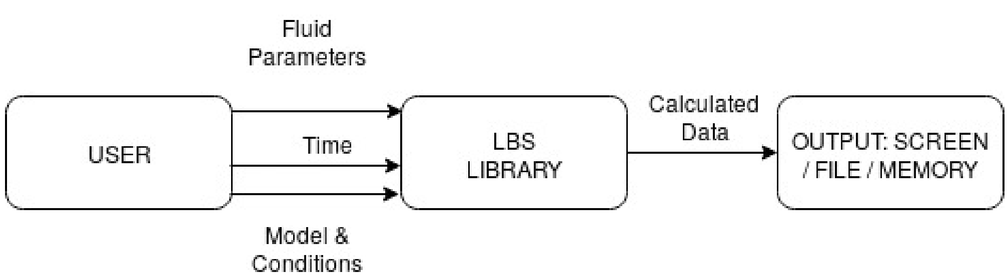

Figure 1: LBS Family Scope

3.3 Characteristics of Intended Reader

The intended reader of this document should have an undergraduate understanding of software requirements specifications and software design principles. Ideally, they will also have at least an undergraduate Level 3 physics understanding, including of lattice structures and fluid dynamics.

3.4 Organization of Document

This document is organized following a template for a CA for scientific computing software proposed by Smith [22]. 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 (for example, A 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, an 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 process is streaming, where the particles move along the lattice via links. The second process is collision, where energy and momentum is transferred among particles that collide [5]. There are many standardized lattice models - individual solvers within the family might 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 [11]).
- 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 [11]).
- 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 [11]).
- Portability: The ease with which a system or component can e transferred from one hardware or software environment to another (IEEE Std 610.12-1990 [11]).
- 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 [11]).
- 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 [11]).
- 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 [11]).
- Scalability: The ability of the system to cope with increasing numbers of users without reducing overall QoS that is delivered to any user (Sommerville [23]).
- Understandability: The ease of understanding the software system (Uchida and Shima [24]).
- 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 [11]).

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	$\mathrm{e} = \frac{dr}{dt}$
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 [17]
Ref. By	T1 T3 T4 A7

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 [3]
Ref. By	T4 DD3 A1 A6

Number	DD3
Label	**Relaxation Rate Towards Equilibrium**
Symbol	τ
SI Units	NA
Equation	$\tau = \frac{12\eta\Delta t}{\Delta \mathrm{x}^2} + \frac{1}{2}$
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 [7]
Ref. By	T1 A1 A5 A6

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 [3]
Ref. By	DD2 A1

Number	DD5
Label	**Fluid Density**
Symbol	ρ
SI Units	$\frac{g}{cm^3}$
Equation	$\rho = \frac{g}{cm^3}$
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 T4 A1 A6

5.4 Goal Statements

Given the lattice model, boundary conditions, simulation time, fluid characteristics, and forces, the goal statements are:

G1 - Location: Predict the location of all fluid particles in the lattice over time.

G2 - Velocity: Predict the velocity of all fluid particles within the lattice over time.

G3 - Fluid Pressure: Predict the pressure of all 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	$f(\mathrm{x} + \mathrm{e}dt, \mathrm{e} + \frac{F}{kg}dt, t + dt)d\mathrm{x}d\mathrm{e} - f(\mathrm{x}, \mathrm{e}, t)d\mathrm{x}d\mathrm{e} = \Omega(f)d\mathrm{x}d\mathrm{e}$
Description	This equation determines the statistical description of a group of particles. The left part of the equation, $f(\mathrm{x} + \mathrm{e}dt, \mathrm{e} + \frac{F}{kg}dt, t + dt)d\mathrm{x}d\mathrm{e}$, represents the distribution function result after an external force F is applied. The middle function, $f(\mathrm{x}, \mathrm{e}, t)d\mathrm{x}d\mathrm{e}$, 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. The variable x represents the vector of the particles within the lattice, e is velocity $\frac{\mathrm{m}}{\mathrm{s}}$, t is time (s), F is force (N), kg is mass. This equation can be further developed for specific instances.
Source	Bolton [7] Mohamad [17]
Ref. By	A5 A6

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.
Source	Bolton [7] Mohamad [17]
Ref. By	T1 A1 A5 A6

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	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. ρ is the fluid density ($\frac{\mathrm{g}}{\mathrm{cm}^2}$). w is the weighting coefficient for the lattice model as the fluid flows through a lattice structure. The weighting coefficients are standard. k is the discretized velocity direction, referring to the directions of the chosen lattice model. $\mathrm{c_k}$ 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. $\mathrm{c_s}$ is the speed of sound, a constant. This equation can be further developed for specific instances.
Source	Bolton [7] Mohamad [17]
Ref. By	T2 A1 A2 A3 A4 A5 A6 A7

Number	T4
Label	**Reynolds Number**
Equation	$Re = \frac{pl\mathrm{e}}{\eta}$
Description	the ration of inertial resistance to viscous resistance for a flowing fluid. ρ is the density of the fluid, e is the velocity, l is the characteristic length of the system, and η is the viscosity
Sources	rey [2]
Ref. By	A6

6 Variabilities

6.1 Assumptions

A1: One or more fluids can be modeled. The fluid parameters are defined in DD2 and DD5. They are used in DD3, and in DD4.

A2: The fluid may 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. The position vector of DD3 is influenced by this.

A6: The user will enter a subset of fluid parameter inputs prior to running the simulation. See Table 2. These inputs are used in DD2, DD3, DD4 and DD5.

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

A8: The average error in outputs between OTS solutions is 3%.

Goal	**Model**	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**
G1	T1					✓	✓		✓
G1	T2	✓				✓	✓		✓
G1	T3	✓	✓	✓	✓	✓	✓	✓	✓
G1	T4	✓	✓	✓	✓	✓	✓		✓
G2	T1					✓	✓		✓
G2	T2	✓				✓	✓		✓
G2	T3	✓	✓	✓	✓	✓	✓	✓	✓
G2	T4	✓	✓	✓	✓	✓	✓		✓
G3	T1					✓	✓		✓
G3	T2	✓				✓	✓		✓
G3	T3	✓				✓	✓	✓	✓
G3	T4			✓	✓	✓	✓		✓

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
destination for output	Set of {file, screen, memory}	compile
boundary parameters	Set of {deflective, non deflective}	compile
model choice	Set of {1D, 2D, 3D}	compile
# of velocity directions	Set of {2, 3, 5, 9, 13, 15, 19, 27}	compile
velocity (e)	Set of $\mathbb{R}$	run
time (t)	Positive $\mathbb{R}$	run
macroscopic velocity (u)	Set of $\mathbb{R}$	run
fluid density (ρ)	Set of positive $\mathbb{R}$	run
position vector (x)	Vector of Set of $\mathbb{R}$	run
viscosity (η)	Set of $\mathbb{R} \geq 0$	run
relaxation rate (τ)	Set of $\mathbb{R}$	run
velocity gradient (γ)	Set of $\mathbb{R}$	run
force (F)	Set of $\mathbb{R} \geq 0$	run
cross-sectional area (A)	Set of $\mathbb{R} \geq 0$	run

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)	Set of {ParMETIS library, PT_Scotch library, block-wide decomposition, domain decomposition, spinoidal decomposition}	compile
coefficient weights	Set of $\{0 < \mathbb{R} < 1\}; \sum \mathbb{R} = 1$	compile
input check	boolean (false if input satisfies input assumptions)	compile
encoding of output	Set of {binary, text}	compile
exception check	boolean (false if no exception condition raised)	compile

Table 3: Calculation Variabilities

6.3 Output

Variability	Parameter of Variation	Binding Time
graphical model	Set of $\{2D, 3D\}; \sum \mathbb{R} = 1$	compile
fluid characteristics	Set of {wall pressure, flow velocity, fluid location, pressure gradient}	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 system shall read a set of input fluid parameters, listed in Table 2, as per A6. These parameters will be used in calculations for T1, T2, and T3.

R2: The system shall allow the user to select from a set of model and velocity direction parameters, listed in Table 2, 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 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 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 in a manner consistent with the decisions made regarding variabilities.

7.2 Non-Functional Requirements

The following are non-functional requirements (NFRs), and their metrics, for a family of LBS. They are defined in Section 5.2:

1. Correctness: The allowable error of the output results, per module, will be less than the average error taken from the results of a sample of OTS solutions.
2. Maintainability: The system shall be documented with a Commonality Analysis (CA), Verification and Validation (VnV) plan, MG (Module Guide), MIS (Module Interface Specification), and User Guide.
3. Performance: The system shall be able to run modules faster than the pseudo-oracle pyLBM.
4. Portability: The system shall be able to run on macOS, Windows, and Linux operating systems.
5. Reliability: The mean time between failures (MTBF) will be longer than the average MTBF of a sample of the OTS solutions.
6. Reusability: Individual modules of the system can be removed and reused in other systems.
7. Robustness: The system will not crash when a user provides invalid input.
8. Scalability: The system must be able to support additional computational models.
9. Understandability: New users must easily understand which LBM models are available in ProgName.
10. Usability: Users will find the system easy to use.

The NFRs have been compared using a pairwise technique based on the Analytic Hierarchy Process (AHP), a decision making method. Each NFR has been compared to every other individual NFR and a professional judgment has been made by the author of this document as to which NFR of the pair has been prioritized in the OTS LBM solutions found in Section 10.2. The judgment is based on a test of the OTS solutions and a review of their documentation.

The results of the comparison 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**	$\sum$
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. The solvers follow the assumptions of A3, A4, A5.

LC2: Wall pressure is not an output variability that is often needed. This may be removed from a family of LBS. See Table 10. The solvers follow the assumptions of A2, A3, A5.

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	T4	DD1	DD2	DD3	DD4	DD5
T1		✓			✓		✓		
T2	✓		✓						
T3			✓		✓				✓
T4					✓	✓			✓
DD1	✓		✓	✓					
DD2				✓			✓	✓	
DD3	✓					✓			
DD4						✓			
DD5			✓	✓					

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

	A1	A2	A3	A4	A5	A6	A7	A8
T1					✓	✓		✓
T2	✓			✓	✓			✓
T3	✓	✓	✓	✓	✓	✓	✓	✓
T4						✓		✓
DD1						✓		✓
DD2	✓					✓		✓
DD3	✓				✓	✓		✓
DD4	✓					✓		✓
DD5	✓					✓		✓
LC1			✓	✓	✓			
LC2		✓	✓		✓			
LC3								
LC4								
LC5								

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

References

[1] Density. URL https://physics.info/density.

[2] Flow regimes. URL https://physics.info/turbulence.

[3] Viscosity. URL https://physics.info/viscosity.

[4] Aursjo. Lattice boltzmann simulations. URL folk.uio.no/olavau/LBsim.

[5] 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.

[6] 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.

[7] Shreedharan Bolton, Schwartz. Lattice boltzmann methods. URL https://personal.ems.psu.edu/~fkd/courses/EGEE520/2017Deliverables/LBM_2017.pdf.

[8] Shiyi Chen and Gary D Doolen. Lattice boltzmann method for fluid flows. *Annual review of fluid mechanics*, 30(1):329–364, 1998.

[9] Fernandez. Multiphase lattice boltzmann suite user guide, 2014. URL https://github.com/carlosrosales/mplabs/blob/master/docs/mp-labs.pdf.

[10] Gouarin Graille. pylbm documentation release 0.3.2, 2018. URL https://buildmedia.readthedocs.org/media/pdf/pylbm/develop/pylbm.pdf.

[11] IEEE Std 610.12-1990. Ieee standard glossary of software engineering terminology. Standard, IEEE, 1991.

[12] Michal Januszewski and Marcin Kostur. Sailfish: A flexible multi-gpu implementation of the lattice boltzmann method. *Computer Physics Communications*, 185(9):2350–2368, 2014.

[13] Li Liu, Lu. Lattice boltzmann method, . URL https://personal.ems.psu.edu/~fkd/courses/EGEE520/2018Deliverables/lbm.pdf.

[14] Li Liu, Lu. The lattice boltzmann method, . URL https://www.shu.ac.uk/research/specialisms/materials-and-engineering-research-institute/what-we-do/expertise/materials-modelling-and-complex-flows/the-lattice-boltzmann-method.

[15] Pedro Lopez. *Thermal Lattice Boltzmann Simulation for Rarefied Flow in Microchannels*. PhD thesis, 2014.

[16] 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.

[17] AA Mohamad. *Lattice Boltzmann Method*, volume 70. Springer, 2011.

[18] 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.

[19] 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.

[20] Florian Schornbaum and Ulrich Rüde. Massively parallel algorithms for the lattice boltzmann method on nonuniform grids. *SIAM Journal on Scientific Computing*, 38 (2):C96–C126, 2016.

[21] MA Seaton and W Smith. Dl meso user manual, 2016.

[22] 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.

[23] Ian Sommerville. *Software Engineering 9*. Pearson Education, 2011.

[24] 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.

MIN_INSTANCES: 2

AVERAGE_ERROR: 3% : Invoked by A8

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[16]	≥0	≥0	3D	15	≥0	≥0	prompt
MUPHY[6]	≥0	≥0	3D	19	≥0	≥0	file
Walberla[20]	≥0	≥0	2D/3D	19	≥0	≥0	file
DL_Meso[21]	≥0	≥0	2D/3D	9,15,19,27	≥0	≥0	file
LB3D[19]	≥0	≥0	3D	19	≥0	≥0	file
Sailfish[12]	≥0	≥0	2D/3D	9,13,15, 19,27	≥0		
mplabs[9]	≥0	≥0	2D/3D	9,19	≥0		file
LBSIM[4]	≥0		2D/3D	6,19	≥0		
pyLBM[10]	≥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[16]	D3Q15i	ParMETIS library	MPI
MUPHY[6]	D3Q19+	PT_Scotch library	MPI
Walberla[20]	D2Q9, D3Q19	block-wide decomposition	MPI
DL_Meso[21]	D2Q9, D3Q15, D3Q19, D3Q27	domain decomposition	MPI
LB3D[19]	D3Q19	spinodal decomposition	MPI
Sailfish[12]	D2Q9, D3Q13, D3Q15, D3Q19, D3Q27	spinoidal decomposition	MPI
mplabs[9]	D2Q9, D3Q19		MPI
LBSIM[4]	D2Q6, D3Q19	spinoidal decomposition	
pylbm[10]	D1Q2, D1Q3, D1Q5, D2Q9, D2Q13, D2Q15, D3Q15, D3Q19		MPI

Table 9: OTS LBS Computational Parameters

solver	wall pressure	flow velocity	graphical model
hemeLB[16]	≥0	≥0	2D/3D
MUPHY[6]			2D/3D
Walberla[20]		≥0	2D/3D
DL_Meso[21]		≥0	2D/3D
LB3D[19]		≥0	2D/3D
Sailfish[12]		≥0	2D
mplabs[9]		≥0	2D/3D
LBSIM[4]			2D/3D
pyLBM[10]			2D/3D

Table 10: OTS LBS Output Parameters

10.3 Coefficient Weights for Equilibrium Distribution Function

lattice model	coefficient weights (w_i)
D1Q2[]	
D1Q3[13]	4/6, i = 0; 1/6, i=1,2
D1Q5[14]	1/2,i = 0; 1/6, i = 1,2; 1/12, i = 3,4
D2Q9[18]	4/9,i = 0; 1/9, i = 1,2,3,4; 1/36, i = 5,6,7,8
D2Q13[15]	3/8,i = 0; 1/12, i = 1,2,3,4; 1/16, i = 5,6,7,8; 1/96, i = 9-12
D2Q15[]	
D3Q15[18]	2/9, i = 0; 1/9, i = 1,2,...,6; 1/72, i = 7,8,...,14
D3Q19[18]	2/9, i = 0; 1/18, i = 1,2,...,6; 1/36, i = 7,8,...,18
D3Q27[18]	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

10.4 First Stage of Implementation

- The first stage of implementation will focus on a library of services using the OTS family member pyLBM.

variability	value	binding time
Computational Model	D2Q9	compile
Re	0.0001 - 50000	run
ρ	0.0708 - 13.6	run
Bulk viscosity	0.0001 - 20000	run
Shear viscosity	0.001 - 20000	run
t	$\mathbb{N} > 0$	run
Pressure gradient	$\mathbb{R}$	run

Table 12: First Stage Implementation Values