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, also referred to as linkages

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
PS	Physical System Description
R	Requirement
T	Theoretical Model

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 [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. Various off the shelf Lattice Boltzmann Solvers (LBS) solutions available today 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 my 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 [17]. It follows a standard pattern of presenting a general system description, commonalities of the members of the software family, variabilities of the members of the software family, and the requirements for the 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 6. Tables of off the shelf 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 variate between the specific 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.

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 DσQκ, 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:

also see slide 35?

-

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	$\frac{\mathrm{m}}{\mathrm{s}}$
Equation	$\mathrm{e} = \frac{d\mathrm{r}}{dt}$
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 [12]
Ref. By	T1 T3

Number	DD2
Label	**Viscosity**
Symbol	η
SI Units	$\mathrm{Pa} - \mathrm{s}$
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	$\tau = \frac{12\eta\Delta t}{\Delta \mathrm{x}^2} + \frac{1}{2}$
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 [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 probabilities of fluid particles in the lattice over time.

G_model: Model the location of fluid particles within the lattice over time.

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

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

G_wallPressure: Model the pressure exerted on the walls of the boundary 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	$f(\mathrm{x}+\mathrm{e}dt, \mathrm{e}+\frac{\mathrm{F}}{\mathrm{m}}dt, t+dt)d\mathrm{x}d\mathrm{e} - f(\mathrm{x}, \mathrm{e}, t)d\mathrm{x}de = \Omega(f)d\mathrm{x}de$
Description	This equation determines the statistical description of a group of particles. The left part of the equation, $f(\mathrm{x}+\mathrm{e}dt, \mathrm{e}+\frac{\mathrm{F}}{\mathrm{kg}}dt, t+dt)d\mathrm{x}de$, represents the distribution function result after an external force F is applied. The middle function, $f(\mathrm{x}, \mathrm{e}, t)d\mathrm{x}de$, 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 length (kg). This equation can be further developed for specific instances.
Source	https://personal.ems.psu.edu/~fkd/courses/EGEE520/2017Deliverables/LBM_2017.pdf Mohamad [12]
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.
Source	https://personal.ems.psu.edu/~fkd/courses/EGEE520/2017Deliverables/LBM_2017.pdf Mohamad [12]
Ref. By	T1

Number	T3
Label	**Equilibrium Distribution Function**
Equation	$f_k^{eq} = pw_k[1 + \frac{2\vec{c}\,\vec{u} - \vec{u}\,\vec{u}}{2c_s^2} + \frac{(\vec{c}\,\vec{u})^2}{2c_s^4}] + O(u^2)$
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. p is the fluid density $(\frac{\mathrm{g}}{\mathrm{cm}^2})$. w is the weighting coefficient for the lattice model. k is the discretized velocity direction, referring to the directions of the chosen lattice model. e is the velocity $(\frac{\mathrm{m}}{\mathrm{s}})$. u is the macroscopic velocity of the fluid, which is a vector field of velocity at a specific position and time. This equation can be further developed for specific instances.
Source	https://personal.ems.psu.edu/~fkd/courses/EGEE520/2017Deliverables/LBM_2017.pdf [12] [12] Mohamad [12]
Ref. By	T2

6 Variabilities

state the binding time for each of the variabilities

input variabilities? what are their associated parameters of variation? see table lecture 5 slide 39/40

Variability	Parameter of Variation
"Border" Shape	Set of defined 2D, defined 3D, undefined
"Border" Parameters	Set of deflective, non deflective
Fluid Parameters	Set of e, t, u, p, x, η, τ, γ, F, A
Model Choice	Set of 1D, 2D, 3D
Velocity Directions	Set of 2, 3, 5, 9, 13, 15, 19, 27
Input Methods	Set of file, type

Table 1: Input Variabilities

[The variabilities are summarized in the following subsections. They may each be summarized separately, like in ?], or in a table, as in ?]. —TPLT]

[For each variability, a description should be given, along with the parameters of variation and the binding time. The parameters of variation give the type that defines possible values. The binding time is when the variability is set. The possible values are specification time (scope time), build time and run time. —TPLT]

6.1 Assumptions

also see slide 47

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: There is a visual presentation of the predicted fluid flow available in all libraries.

A5: Data representing the predicted fluid flow can be passed externally of the system through memory or a file.

A6: Weight coefficients are standard for each lattice model. See the Appendix section for weight coefficients for a subset of lattice models.

table with goals and theoretical models on y axis and assumptions on x axis?

A7: [Short description of each assumption. Each assumption should have a meaningful label. Use cross-references to identify the appropriate traceability to T, GD, DD etc., using commands like dref, ddref etc. —TPLT]

table with goals and theoretical models on y axis and assumptions on x axis

[Input assumptions will be appropriate for many problems. Some input will have simplifying constraints, and other inputs will not. —TPLT]

6.2 Calculation

Variability	Parameter of Variation

Table 2: Calculation Variabilities

[The calculation variabilities should be as abstract as possible. If there are variabilities that are related to imposed design decisions, the system constraints section should be referenced for the relevant constraint. Design constraint related variabilities should be listed separately. —TPLT]

[Variabilities related to data structure choices would go in this section. However, these variabilities are related to design, so they should be separated from the more abstract variabilities. —TPLT]

[Algorithmic variations would go here as well, but as for data structures, they should be separated from the more abstract variabilities. —TPLT]

6.3 Output

Variability	Parameter of Variation

Table 3: 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

[Since the CA will often be applied to a library, the functionality will not be a single use case. Therefore, this section should summarize the family of potential requirements. A good way to provide an overview of the functional requirements would be to provide multiple use cases on how the library will be employed. —TPLT]

R1: [Requirements for the inputs that are supplied by the user. This information has to be explicit. —TPLT]

R2: [It isn't always required, but often echoing the inputs as part of the output is a good idea. —TPLT]

R3: [Calculation related requirements. —TPLT]

R4: [Verification related requirements. —TPLT]

R5: [Output related requirements. —TPLT]

7.2 Nonfunctional Requirements

rather than absolute quantification of nfrs, use relative comparison between other program family members

specify requirements in big O notation

NFR -¿ AHP

relative comparison between programs is a validateable requirement

focus on a posteriori description rather than a priori specification

identify benchmark test problems

test cases built starting from assumed solutions

see slide 42

ahp each design against each nfr see slide 42

[To allow the Non-Functional Requirements (NFRs) to vary between family members, try to parameterize them. The value of the parameter is than a variability. —TPLT]

[An important variability between family members it the relative importance of the NFRs. **?**] shows how pairwise comparisons can be used to rank the importance of NFRs. —TPLT]

[List your nonfunctional requirements. You may consider using a fit criterion to make them verifiable. —TPLT]

Correctness - satisfies reqs spec Reliability - usually does what it is intnded to do Robustness - behaves reasnably during exceptional situations Performance (low computer resource usage): evaluated using empirical measurement, analysis of an analytic model, or analysis of a simulation model Usability - ease of usage, user interface,installability? maintainability - ease of modification; corrective, adaptive, perfective Reusability - create a new product? standardized? instantiable components? Portability - run in different environments? hardware platform, operating system, supporting software, user base add more understandability - ease with which the reqs, design, implementation, documentation, etc can be understood -¿ impacts verifiability, maintainability, resusablity

Define these in the document before using them.

8 Likely Changes

LC1: [If there is a ranking of variabilities, or combinations of variabilities, that are more likely, this information can be included here. —TPLT]

9 Traceability Matrices and Graphs

[You will have to add tables. —TPLT]

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

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

10 Appendix

[Your report may require an appendix. For instance, this is a good point to show the values of the symbolic parameters introduced in the report. —TPLT]

10.1 Symbolic Parameters

[The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance. —TPLT]
[Advice on using the template:

- Assumptions have to be invoked somewhere
- "Referenced by" implies that there is an explicit reference
- Think of traceability matrix, list of assumption invocations and list of reference by fields as automatically generatable
- If you say the format of the output (plot, table etc), then your requirement could be more abstract
- For families the notion of binding time should be introduced
- Think of families as a library, not as a single program

—TPLT]

10.2 Off The Shelf Solutions

OTS1

The following table lists some Lattice Boltzmann Solvers, along with an incomplete list of some input parameters. Cells that are blank represent an unknown value. Assumptions have been stated in the assumptions section.

10.2.1

The following table lists some Lattice Boltzmann Solvers, along with an incomplete list of some computational parameters. Cells that are blank represent an unknown value. Assumptions have been stated in the assumptions section.

The following table lists some Lattice Boltzmann Solvers, along with an incomplete list of some output parameters.

Cells that are blank represent an unknown value. Assumptions have been stated in the assumptions section.

10.3 Coefficient Weights for Equilibrium Distribution Function

Blank coefficients are unknown to the author at the time of writing.

solver	velocity	density	model	velocity directions	time	viscosity	input method
hemeLB[11]	≥0	≥0	3D	15	≥0	≥0	prompt
MUPHY[5]	≥0	≥0	3D	19	≥0	≥0	file
Walberla[15]	≥0	≥0	2D/3D	19	≥0	≥0	file
DL_Meso[16]	≥0	≥0	2D/3D	9,15,19,27	≥0	≥0	file
LB3D[14]	≥0	≥0	3D	19	≥0	≥0	file
Sailfish[10]	≥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 4: LBS Inputs

solver	computational model	decomposition technique	parallel interface
hemeLB[11]	D3Q15i	ParMETIS library	MPI
MUPHY[5]	D3Q19+	PT_Scotch library	MPI
Walberla[15]	D2Q9, D3Q19	block-wide decomposition	MPI
DL_Meso[16]	D2Q9, D3Q15, D3Q19, D3Q27	domain decomposition	MPI
LB3D[14]	D3Q19	spinodal decomposition	MPI
Sailfish[10]	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 5: LBS Computational Parameters

solver	wall pressure	flow velocity	graphical model
hemeLB[11]	≥0	≥0	2D/3D
MUPHY[5]			2D/3D
Walberla[15]		≥0	2D/3D
DL_Meso[16]		≥0	2D/3D
LB3D[14]		≥0	2D/3D
Sailfish[10]		≥0	2D
mplabs[8]		≥0	2D/3D
LBSIM[3]			2D/3D
pylbm[9]			2D/3D

Table 6: LBS Output Parameters

lattice model	coefficient weights (w_i)
D1Q2[]	
D1Q3[]	4/6, i = 0; 1/6, i=1,2
D1Q5[]	
D2Q9[13]	4/9,i = 0; 1/9, i = 1,2,3,4; 1/36, i = 5,6,7,8
D2Q13[]	
D2Q15[]	
D3Q15[13]	2/9, i = 0; 1/9, i = 1,2,...,6; 1/72, i = 7,8,...,14
D3Q19[13]	2/9, i = 0; 1/18, i = 1,2,...,6; 1/36, i = 7,8,...,18
D3Q27[13]	8/27, i = 0; 2/27, i = 1,2,...,6; 1/54, i = 7,8,...,18; 1/216. i = 19,20,...,26

Table 7: Lattice Model Coefficient Weights