Lattice Boltzmann Solvers

Peter Michalski

October 5, 2019

[The CA template is related to the SRS template. Many of the sections are in common. The notes and advice for the SRS template are not reproduced here. Please have a look at the SRS template for advice. —TPLT]

[This CA template is based on **?**]. An example for a family of material models is given in **?**]. This example is for a physics based family. Often the families will be based on generic numerical techniques, rather than physics. —TPLT]

[A good mindset for thinking about the families is often to think of the family as providing a library of services, as opposed to a single executable. The library of services can be used to build an application that uses a subset of the services, which is like providing the smaller library as a single family member. —TPLT]

[In CAS 741, you will not have to implement the entire family. We will decide on a reasonable subset of the family for implementation. —TPLT]

1 Revision History

Date	**Version**	**Notes**
Date 1	1.0	Notes
Date 2	1.1	Notes

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

[Only include the units that your CA actually uses. If there are no units for your problem, like for a general purpose library, you should still include the heading, with the content "not applicable" (or similar). —TPLT]

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	$\frac{1}{s}$	shear rate
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	$\frac{g}{cm^2}$	fluid density
w	N/A	weight coefficient (implementation specific)
u	$\frac{m}{s}$	macroscopic velocity of fluid
D	N/A	dimension component of lattice model
Q	N/A	number of velocity directions of lattice model, also referred to as linkages

[Use your problems actual symbols. The si package is a good idea to use for units. —TPLT] [For the case of a generic numerical library, units will likely not be included. For instance, a linear ODE solver will not know the units of its coefficients. —TPLT]

2.3 Abbreviations and Acronyms

symbol	description
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
Lattice Boltzmann Solvers	[put your famram name here —TPLT]
T	Theoretical Model

[Add any other abbreviations or acronyms that you add. —TPLT] [Only include abbreviations and acronyms that are actually used. —TPLT]

Contents

3 Introduction

Lattice Boltzmann Methods are a family of fluid dynamics algorithms for simulating single-phase and multiphase fluid flows, often incorporating additional physical complexities. Chen and Doolen [3]. These methods consider the behavious of a collection of particles as a single "unit" at the mesoscopic scale. on probability of particles rather than mapping every particle (miscoscopic scale - Hamilton's equation) which is heavy on data or the Navier -strokes method (macroscopic) which is difficult ot solve analyticlaly. ... very efficient...growing in popularity.. Lattice Boltzmann methods are a modern approach to computational fluid dynamics and close the gap betweeen the micro and macro scale.

3.1 Purpose of Document

This document will outline

3.2 Scope of the Family

3.3 Characteristics of Intended Reader

3.4 Organization of Document

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

[Your system context will likely include an explicit list of user and system responsibilities —TPLT]

- User Responsibilities:
  -
- Lattice Boltzmann Solvers Responsibilities:
  - Detect data type mismatch, such as a string of characters instead of a floating point number
  -

4.2 Potential User Characteristics

The end user of Lattice Boltzmann Solvers should have an understanding of undergraduate Level 1 Calculus and Physics.

4.3 Potential System Constraints

SC - The system will use a lattice structure to map the fluid flow

[You may not have any system constraints. —TPLT]

[If you need to make design decisions for your family, these decisions will be made here as constraints. For instance, if all inputs will have to use the same file format, this would be a constraint that would be included here. —TPLT]

[You should generally limit the number of constraints, to keep the CA abstract. —TPLT]

5 Commonalities

5.1 Background Overview

There are two steps...

The particles are mapped using a lattice structure. The lattice structure can be a !D, 2D, or 3D model with varying velocity directions. The notatin is DXQY, where X epresends the number of dimensions and Y represents the number of velociy directions. The lattice models are vast and standardized, but the individual solvers within the family may only use a subset of them. The LBM uses the distribution funciton to predict which direction of the specified model a group of particles will travel.

Sum of the weight coefficients is 1.

Particles bounce back when they encounter a boundary.

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. [Modify the examples below for your problem, and add additional definitions as appropriate. —TPLT]

Number	DD1
Label	**Velocity**
Symbol	e
SI Units	$\frac{\text{m}}{\text{s}}$
Equation	$\text{e} = \frac{d\text{r}}{dt}$
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 [8]
Ref. By	T1 T3

Number	DD2
Label	**Viscosity**
Symbol	η
SI Units	$\text{Pa} - \text{s}$
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 rate of shear (s^{-1}).
Sources	
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	mainonlinesource
Ref. By	T1

Also add about the lattice but in another section - figure out which one.

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

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

G_model: 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	$f(\mathrm{x}+\mathrm{e}dt,\mathrm{e}+\frac{\mathrm{F}}{\mathrm{m}}dt,t+dt)d\mathrm{x}d\mathrm{e}-f(\mathrm{x},\mathrm{e},t)d\mathrm{x}d\mathrm{e}=\Omega(f)d\mathrm{x}d\mathrm{e}$
Description	This equation determines the statistical description of a group of particles. The left part of the equation, $f(\mathrm{x}+\mathrm{e}dt,\mathrm{e}+\frac{\mathrm{F}}{\mathrm{kg}}dt,t+dt)d\mathrm{x}d\mathrm{e}$, 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 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 [8]
Ref. By	

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.

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 [8]
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 [8] [8] Mohamad [8]
Ref. By	T2

The Lattice-Boltzmann Methods compute velocity and pressure as momentum of the distribution functions.

The Lattice-Boltzmann Methods rely on a two step process for computing fluid dynamics. The first processes is streaming (define more). The methods consider particles at each grid point, moving to an adjacent grid point at its velocity. The second process is accounting for collision (define). The methods can be applied to a variety of computational fluid dynamics problems, and rely on further derivations of the above theoretical models in order to be applicable to various fluid dynamics scopes.

[In a CA, the TMs often do not need to be refined. However, this is not a rule. In some cases, it may make sense to introduce an IM, or possibly even a GD in between the TM and the IM. —TPLT]

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

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

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

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

A2: The fluid flows through space via a lattice structure, moving between lattice nodes via linkages Q.

A3: There is a visual presentation of the predicted fluid flow available in all libraries.

A4: Data representing the predicted fluid flow can be passed externally of the system through memory or a file.

A5: 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?

A6: [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]

Variability	Parameter of Variation

Table 2: Calculation Variabilities

Variability	Parameter of Variation

Table 3: Output Variabilities

6.2 Calculation

[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

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

[2] 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.

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

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

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

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

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

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

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[7]	≥0	≥0	3D	15	≥0	≥0	prompt
MUPHY[2]	≥0	≥0	3D	19	≥0	≥0	file
Walberla[11]	≥0	≥0	2D/3D	19	≥0	≥0	file
DL_Meso[12]	≥0	≥0	2D/3D	9,15,19,27	≥0	≥0	file
LB3D[10]	≥0	≥0	3D	19	≥0	≥0	file
Sailfish[6]	≥0	≥0	2D/3D	9,13,15, 19,27	≥0		
mplabs[4]	≥0	≥0	2D/3D	9,19	≥0		file
LBSIM[1]	≥0		2D/3D	6,19	≥0		
pylbm[5]	≥0	≥0	1D,2D,3D	2,3,5,9, 13,15,19	≥0	≥0	file

Table 4: LBM Solver Inputs

solver	computational model	decomposition technique	parallel interface
hemeLB[7]	D3Q15i	ParMETIS library	MPI
MUPHY[2]	D3Q19+	PT_Scotch library	MPI
Walberla[11]	D2Q9, D3Q19	block-wide decomposition	MPI
DL_Meso[12]	D2Q9, D3Q15, D3Q19, D3Q27	domain decomposition	MPI
LB3D[10]	D3Q19	spinodal decomposition	MPI
Sailfish[6]	D2Q9, D3Q13, D3Q15, D3Q19, D3Q27	spinoidal decomposition	MPI
mplabs[4]	D2Q9, D3Q19		MPI
LBSIM[1]	D2Q6, D3Q19	spinoidal decomposition	
pylbm[5]	D1Q2, D1Q3, D1Q5, D2Q9, D2Q13, D2Q15, D3Q15, D3Q19		MPI

Table 5: LBM Solver Computational Parameters

solver	wall pressure	flow velocity	graphical model
hemeLB[7]	≥0	≥0	2D/3D
MUPHY[2]			2D/3D
Walberla[11]		≥0	2D/3D
DL_Meso[12]		≥0	2D/3D
LB3D[10]		≥0	2D/3D
Sailfish[6]		≥0	2D
mplabs[4]		≥0	2D/3D
LBSIM[1]			2D/3D
pylbm[5]			2D/3D

Table 6: LBM Solver Output Parameters

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

Table 7: Lattice Model Coefficient Weights