Module Interface Specification for Lattice Boltzmann Solvers

Peter Michalski

November 17, 2019

1 Revision History

Date	Version	Notes
Nov. 25, 2019	1.0	Initial Document

2 Symbols, Abbreviations and Acronyms

See CA Documentation for Lattice Boltzmann Solvers (Michalski, a).

Contents

3 Introduction

The following document details the Module Interface Specifications for Lattice Boltzmann Solvers, which provides a library of 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, 1998).

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found here (Michalski, b).

4 Notation

[You should describe your notation. You can use what is below as a starting point. —SS]

The structure of the MIS for modules comes from Hoffman and Strooper (1999), with the addition that template modules have been adapted from Ghezzi et al. (2003). The mathematical notation comes from Chapter 3 of Hoffman and Strooper (1999). For instance, the symbol := is used for a multiple assignment statement and conditional rules follow the form $(c_1 \Rightarrow r_1 | c_2 \Rightarrow r_2 | ... | c_n \Rightarrow r_n)$.

The following table summarizes the primitive data types used by Lattice Boltzmann Solvers.

Data Type	**Notation**	**Description**
character	char	a single symbol or digit
integer	$\mathbb{Z}$	a number without a fractional component in (-∞, ∞)
natural number	$\mathbb{N}$	a number without a fractional component in [1, ∞)
real	$\mathbb{R}$	any number in (-∞, ∞)

The specification of Lattice Boltzmann Solvers uses some derived data types: sequences, strings, and tuples. Sequences are lists filled with elements of the same data type. Strings are sequences of characters. Tuples contain a list of values, potentially of different types. In addition, Lattice Boltzmann Solvers uses functions, which are defined by the data types of their inputs and outputs. Local functions are described by giving their type signature followed by their specification.

5 Module Decomposition

The following table is taken directly from the Module Guide document for this project.

Level 1	**Level 2**
Hardware-Hiding Module	M1: Hardware Hiding Module
Behaviour-Hiding Module	M2: System Control Module M3: Input Reading Module M4: Input Checking Module M5: LBM Control Module M6: Streaming Module M7: Collision Module M8: Problem Module M9: Lattice Module M10: Boundary Module
Software Decision Module	M11: Output Module

Table 1: Module Hierarchy

6 MIS of M1: Hardware Hiding Module

The secrets of this module are the data structure and algorithms used to implement the virtual hardware.

6.1 Module

Hardware Hiding

6.2 Uses

N/A

6.3 Syntax

N/A

6.3.1 Exported Constants

N/A

6.3.2 Exported Access Programs

N/A

6.4 Semantics

N/A

6.4.1 State Variables

N/A

6.4.2 Environment Variables

The module has external interaction with the environment when either the Output Module M11 (Section 16) or Input Reading Module M3 (Section 8) require its services for reading inputs or writing outputs.

6.4.3 Assumptions

M11 (Section 16) or M3 (Section 8) have called the modules services.

6.4.4 Access Routine Semantics

N/A

6.4.5 Local Functions

N/A

7 MIS of M2: System Control Module

The secret of this module is the algorithm to control Lattice Boltzmann Solvers.

7.1 Module

System Control

7.2 Uses

- Output (Section 16)
- Input Reading (Section 8)
- LBM Control (Section 10)
- Problem Parameter (Section 13)

7.3 Syntax

7.3.1 Exported Constants

N/A

7.3.2 Exported Access Programs

N/A

7.4 Semantics

7.4.1 State Variables

LIBRARY_IN: string
DIMENSIONS_IN: $\mathbb{N}$
VEL_DIR_IN: $\mathbb{N}$
inputValues: array of strings, $\mathbb{R}$, $\mathbb{N}$
LBMout: array of $\mathbb{R}$

7.4.2 Environment Variables

N/A

7.4.3 Assumptions

The user has run the Lattice Boltzmann Solvers program.

7.4.4 Access Routine Semantics

N/A

7.4.5 Local Functions

Name	**In**	**Out**	**Exceptions**
LIBRARY_IN	array of strings, $\mathbb{R}$, $\mathbb{N}$	string	-
DIMENSIONS_IN	array of strings, $\mathbb{R}$, $\mathbb{N}$	$\mathbb{N}$	-
VEL_DIR_IN	array of strings, $\mathbb{R}$, $\mathbb{N}$	$\mathbb{N}$	-

8 MIS of M3: Input Reading Module

The secret of this module is the algorithm that gathers the input data.

8.1 Module

Input Reading

8.2 Uses

- Input Checking
- Hardware Hiding

8.3 Syntax

8.3.1 Exported Constants

inputValues: array of strings, $\mathbb{R}$, $\mathbb{N}$

8.3.2 Exported Access Programs

Name	In	Out	Exceptions
inputArray		array of strings, $\mathbb{R}$, $\mathbb{N}$	-

8.4 Semantics

8.4.1 State Variables

Input_Location: string

8.4.2 Environment Variables

N/A

8.4.3 Assumptions

The System Control Module M2 (Section 7) has called the inputArray function of this module.

8.4.4 Access Routine Semantics

inputArray():

- output: inputValues := $+(\text{inputValues}[i] \mid \text{string} \cup \mathbb{R} \cup \mathbb{N} \mid \text{:readline})$
- exception: N/A

The function will read all lines from the input file and place each value into an array in the form array[key:string][value: string $\cup$ $\mathbb{R}$ $\cup$ $\mathbb{N}$]. There is no input into the file, the output will be the array. The location of the input file is in the state variable Input_Location.

8.4.5 Local Functions

Name	**In**	**Out**	**Exceptions**
open	string	object	NOT_FOUND ERR_READ
readline	object	-	NO_LINES OUT_BOUNDS

9 MIS of M4: Input Checking Module

This secret of this module is the algorithm that checks if input values fall within allowable parameters.

9.1 Module

Input Checking

9.2 Uses

N/A

9.3 Syntax

9.3.1 Exported Constants

N/A

9.3.2 Exported Access Programs

Name	**In**	**Out**	**Exceptions**
verifyInputs	string array		OUT_OF_BOUNDS UNKN_PARM

9.4 Semantics

9.4.1 State Variables

LIBRARIES: set of strings: {pyLBM}
DIMENSIONS: set of $\mathbb{N}$: {2}
VEL_DIRS: set of $\mathbb{N}$: {9}
REYNOLDS_MIN: $\mathbb{R}$: {0.001}
REYNOLDS_MAX: $\mathbb{R}$: {5000}
DENSITY_MIN: $\mathbb{R}$: {0.0708}
DENSITY_MIN: $\mathbb{R}$: {13.6}
BULK_VIS_MIN: $\mathbb{R}$: {0.0001}
BULK_VIS_MIN: $\mathbb{R}$: {20000}
SHEAR_VIS_MIN: $\mathbb{R}$: {0.001}
SHEAR_VIS_MIN: $\mathbb{R}$: {20000}
TIME_MIN: $\mathbb{N}$: {1}
LIBRARY_IN: string
DIMENSIONS_IN: $\mathbb{N}$
VEL_DIR_IN: $\mathbb{N}$

Re: $\mathbb{R}$
ρ: $\mathbb{R}$
η_b: $\mathbb{R}$
η_s: $\mathbb{R}$
t: $\mathbb{N}$
a: $\mathbb{R}$
c_s: $\mathbb{R}$
e: $\mathbb{R}$
F: $\mathbb{R}$
kg: $\mathbb{R}$
A: $\mathbb{R}$
$\mathbf{u}$: $\mathbb{R}$

9.4.2 Environment Variables

N/A

9.4.3 Assumptions

The Input Reading Module M3 (Section 8) has called the verifyInputs function of this module.

9.4.4 Access Routine Semantics

verifyInputs():

- output: N/A
- exception: OUT_OF_BOUNDS := if inputValue(i_2) $>$ parm_max $\cup$ $<$ parm_min $\Rightarrow$ ERROR
- exception: UNKN_PARM := if inputValue(i_1) != known_parm $\Rightarrow$ ERROR

The function will iterate through each inputValue array key and check if the key is known to the program, and if the values of known keys falls within an acceptable range.

9.4.5 Local Functions

N/A

10 MIS of M5: LBM Control Module

The secret of this module is the algorithm which controls the LBM library.

10.1 Module

LBM Control

10.2 Uses

- Streaming
- Collision

10.3 Syntax

10.3.1 Exported Constants

10.3.2 Exported Access Programs

Name	In	Out	Exceptions
vortVal	array of strings, $\mathbb{R}$, $\mathbb{N}$	array of $\mathbb{R}$	-

10.4 Semantics

10.4.1 State Variables

array of vorticity vector values

10.4.2 Environment Variables

N/A

10.4.3 Assumptions

The System Control Module M2 (Section 7) has called the vortVal function of this module.

10.4.4 Access Routine Semantics

vortVal():

- output: vortVal := (inputValues[0...n]| array of $\mathbb{N}$|:streaming()*collision())
- exception: N/A

The function will calculate the vorticity vector values, iterating through each velocity direction, calling the streaming and collision module functions.

10.4.5 Local Functions

N/A

11 MIS of M6: Streaming Module

The secret of this module is the algorithm to calculate the streaming pf particles.

11.1 Module

Streaming

11.2 Uses

N/A

11.3 Syntax

11.3.1 Exported Constants

N/A

11.3.2 Exported Access Programs

Name	**In**	**Out**	**Exceptions**
bgkCollision	array of strings, $\mathbb{R}$, $\mathbb{N}$	$\mathbb{R}$	NAN_ERROR

11.4 Semantics

11.4.1 State Variables

N/A

11.4.2 Environment Variables

N/A

11.4.3 Assumptions

The LBM Control Module M5 (Section 10) has called the bgkCollision function of this module.

11.4.4 Access Routine Semantics

[accessProg —SS]():

- output: bgkValue := (inputValues[0...n]|$\mathbb{N}$|:relatUpdate()*equilDistrib()*relaxRate())
- exception: if |var| $>$ |type| $\cup <$ |type| $\Rightarrow$ ERROR

The function calculates the Bhatnagar-Gross-Krook collision operator value for each velocity direction (i) using the functions for the relaxation rate towards equilibrium, equilibrium distribution function, and relaxation update.

11.4.5 Local Functions

Name	**In**	**Out**	**Exceptions**
relatUpdate	array of strings, $\mathbb{R}$, $\mathbb{N}$	$\mathbb{R}$	NAN_ERROR
equilDistrib	array of strings, $\mathbb{R}$, $\mathbb{N}$	$\mathbb{R}$	NAN_ERROR
relaxRate	$\mathbb{R}$, $\mathbb{N}$	$\mathbb{R}$	NAN_ERROR

12 MIS of M7: Collision Module

The secret of this module is the algorithm to calculate the collision of particles.

12.1 Module

Collision

12.2 Uses

N/A

12.3 Syntax

12.3.1 Exported Constants

N/A

12.3.2 Exported Access Programs

Name	In	Out	Exceptions
streamingFunc	array of strings, $\mathbb{R}$, $\mathbb{N}$	$\mathbb{R}$	NAN_ERROR

12.4 Semantics

12.4.1 State Variables

N/A

12.4.2 Environment Variables

N/A

12.4.3 Assumptions

The LBM Control Module M5 (Section 10) has called the streamingFunc function of this module.

12.4.4 Access Routine Semantics

[accessProg —SS]():

- output: strmValue := (inputValues[0...n]|$\mathbb{N}$|:streamingFunc())
- exception: if |var| > |type| $\cup$ < |type| $\Rightarrow$ ERROR

The function calculates the streaming step value for each velocity direction (i) using the probability density function.

12.4.5 Local Functions

Name	In	Out	Exceptions
prbDnsFunc	array of $\mathbb{R}$, $\mathbb{N}$	$\mathbb{R}$	NAN_ERROR

13 MIS of M8: Problem Module

The secret of this module is the structure of the LBM input parameters.

13.1 Module

Problem Parameter

13.2 Uses

- Lattice
- Boundary

13.3 Syntax

13.3.1 Exported Constants

N/A

13.3.2 Exported Access Programs

Name	**In**	**Out**	**Exceptions**
formatInput	array of $\mathbb{R}$, $\mathbb{N}$	array of $\mathbb{R}$, $\mathbb{N}$	-

13.4 Semantics

13.4.1 State Variables

N/A

13.4.2 Environment Variables

N/A

13.4.3 Assumptions

The System Control Module M2 (Section 7) has called the formatInput function of this module.

13.4.4 Access Routine Semantics

vortVal():

- output: vortVal := (inputValues| array of strings and $\mathbb{N}$|:setLattice()*boundFunc())

- exception: N/A

The function will set up the structure for the LBM input parameters based on the library that the user has requested. The function will use the lattice and boundary module functions.

13.4.5 Local Functions

N/A

14 MIS of M9: Lattice Module

The secret of this module is the structure of the lattice model.

14.1 Module

Lattice

14.2 Uses

N/A

14.3 Syntax

14.3.1 Exported Constants

N/A

14.3.2 Exported Access Programs

Name	In	Out	Exceptions
setLattice	array of $\mathbb{R}$	array of $\mathbb{N}$	No_LATTICE

14.4 Semantics

14.4.1 State Variables

Array of tuples: parameterValues:= $<\mathbb{R}><\mathbb{R}><\mathbb{R}, \mathbb{N}_0 ... \mathbb{N}_n>$

14.4.2 Environment Variables

N/A

14.4.3 Assumptions

The Problem Module M8 (Section 13) has called the setLattice function of this module.

14.4.4 Access Routine Semantics

setLattice():

- output: output := (DIMENSIONS_IN, VEL_DIR_IN | array of $\mathbb{N}$|:parameterValues)
- exception: if DIMENSIONS_IN $\cap$ VEL_DIR_IN $\notin$ parameterValues $\Rightarrow$ NO_LATTICE

The function sets coefficient weight data for the selected lattice model.

14.4.5 Local Functions

N/A

15 MIS of M10: Boundary Module

The secret of this module is the structure of the model boundary.

15.1 Module

Boundary

15.2 Uses

N/A

15.3 Syntax

15.3.1 Exported Constants

N/A

15.3.2 Exported Access Programs

Name	In	Out	Exceptions
boundFunc	array of $\mathbb{R}$	array of $\mathbb{R}$	-

15.4 Semantics

15.4.1 State Variables

N/A

15.4.2 Environment Variables

N/A

15.4.3 Assumptions

The Problem Module M8 (Section 13) has called the boundFunc function of this module.

15.4.4 Access Routine Semantics

boundFunc():

- output: output := (inputValues | array of $\mathbb{N}$|:inputValues[])
- exception: N/A

The function sets coefficient weight data for the selected lattice model.

15.4.5 Local Functions

N/A

16 MIS of M11: Output Module

The algorithm to convert the LBM output into an image.

16.1 Module

Output

16.2 Uses

- System Control Module

16.3 Syntax

16.3.1 Exported Constants

N/A

16.3.2 Exported Access Programs

Name	In	Out	Exceptions
imageFunc	array of $\mathbb{R}$	image format	-

16.4 Semantics

16.4.1 State Variables

N/A

16.4.2 Environment Variables

N/A

16.4.3 Assumptions

The System Control Module M2 (Section 13) has called the imageFunc function of this module.

16.4.4 Access Routine Semantics

imageFunc():

- output: (LBMout | image format |:external image rendering function)
- exception: N/A

The function converts the information from the LBM algorithm output into an image format.

16.4.5 Local Functions

N/A

References

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

Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2003.

Daniel Hoffman and Paul Strooper. Software Design, Automated Testing, and Maintenance–A Practical Approach. 1999.

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

Peter Michalski. Lattice Boltzmann Solvers, b. URL https://github.com/peter-michalski/LatticeBoltzmannSolvers.

17 Appendix

[Extra information if required —SS]