


ListLBM

Soroush Khajepour

 sk451@hw.ac.uk

 [Google Scholar](#)

 [LBM in Practice](#)

1 Introduction

I have designed listLBM as a part of my PhD program to simulate multiphase flow in porous media. The code is written with object-oriented programming model provided by Fortran 2003 and 2008 standards. The main point of the code is the elimination of solid nodes which are not interacting with the fluid from calculations. The focused system is a 3D/2D sparse matrix. The nodes are mapped on the *mainlist* (main vector) while knowing their neighbors. Therefore, the system is treated as an unstructured mesh. The nodes which are experiencing the same type of collision are placed in the same *sublist*. Therefore, each *sublist* has a collision which should be initialized at the beginning of the code. The *mainlist* and *sublists* form a *part* which handles a lattice. For multicomponent simulations, multiple *parts* are defined. The code is written in a general way that it can handle D2Q9, D3Q15, D3Q19, and D3Q27 lattices with minimum changes. While the code has a great potential for parallelization, I haven't done it yet.

2 Installation

To install the code, a modern GFortran compiler (GCC) is needed which supports Fortran 2003 and 2008 standards. The code is successfully tested with GFortran 5.4.1 in Ubuntu (Linux). It looks possible to run the code on Windows; however, I've never tried. The makefile is generated using CMake, therefore it must be installed beforehand. The VTK files are created with VTKFortran library which must be set-up before compiling the code which is explained in section 2.1.

2.1 VTK Fortran library

VTKFortran (Lib_VTK_IO) library is necessary for compiling and getting VTK files output from the code. The library is published under different free source licenses including GPL v3 and MIT on Github:

<https://github.com/szaghi/VTKFortran/wiki/Download>

The only version I have tested with listLBM is VTKFortran-1.1.0, therefore, I recommend users to do the same. Simply download the library on your Linux system, in its directory, run

make

Because listLBM currently runs only in serial mode, ensure you compile the VTKFortran without *OpenMP* and *MPI* (their value is equal to “no” in the makefile). These are the default settings when I have downloaded the version 1.1.0. The compiler I have used is GFortran 5.4.1 which is used for

compiling listLBM as well. After successfully compiling VTKFortran, do these file-copies

SomePath/VTKFortran-1.1.0/mod/lib_vtk_io.mod => *SomePath/listLBM/extlib*

SomePath/VTKFortran-1.1.0/obj/ir_precision.o => *SomePath/listLBM/extlib*

SomePath/VTKFortran-1.1.0/obj/lib_base64.o => *SomePath/listLBM/extlib*

SomePath/VTKFortran-1.1.0/obj/lib_vtk_io.o => *SomePath/listLBM/extlib*

This process should be done just once on a specific machine. But remember if you change the operating system, change your compiler, or even upgrade the compiler, these files must be made again. The existence of these files is necessary for a successful compilation of listLBM.

2.2 Compiling listLBM

The code is compiled with the aid of CMake as explained below. Ensure you have added, VTKFortran library before compiling (see section 2.1).

1) You should choose the desired example. Examples are all placed in the *SomePath/listLBM/examples* directory and divided into *2D* and *3D*. Pick one, for instance, *shanchen*. Open *SomePath/listLBM/CmakeLists.txt* file, and set example path generally like this:

```
set(example_path ../examples/PathOfExample/main.f90)
```

for instance, for the 3D shanchen case:

```
set(example_path ../examples/3d/shanchen/main.f90)
```

Define the project name specific to the example like:

```
set (project_name shanchen)
```

Save and close *SomePath/listLBM/CmakeLists.txt* file.

2) Create a directory inside the listLBM directory, with a name specific to the example, for instance, *build_shanchen*.

3) Open terminal, go into the directory you have made in step 2 and run command

```
cmake ..
```

```
make
```

4) Run the example

./shanchen

The VTK files are created and put in the *vtk* directory, for instance, *SomePath/listLBM/build_shanchen/vtk*.

Open the VTK files with ParaView software. If you want to make another example, start with step 1. In this way, each compiled example has its own directory. You can set the build type in terminal as:

cmake -DCMAKE_BUILD_TYPE=Debug ..

cmake -DCMAKE_BUILD_TYPE=Release ..

Otherwise "Release" is the default build type. For information about compiler flags, refer to *SomePath/listLBM/CmakeLists.txt* file.

2.3 Editing an example

Examples are designed in the *main.f90* files placed in their directory. After compiling an example successfully from section 2.2, open the related *main.f90* file in a text editor and change parameters, add functions, and so forth as you wish. Then, save the file and in the terminal from the build directory, for instance, *SomePath/listLBM/build_shanchen/* run

make

and run the example executable file, for instance

./shanchen

A good point about this design is that the libraries of listLBM are compiled only once and after every make only *main.f90* is compiled.

3 Examples

Currently, the below examples are available.

2D cases are

- 1) Shan-Chen multi-component drop rising
- 2) Shan-Chen multiphase liquid drop in equilibrium with its vapor
- 3) Single-phase Von Karman vortex.

3D cases are

- 1) Shan-Chen multiphase vapor bubble in equilibrium with liquid surroundings.
- 2) Single-phase flow in porous media. Make sure the “3D_porous_250x250x4.dat” file is placed in the build directory. Density/pressure boundaries are set at the left and right of the domain.
- 3) Single-phase flow over a sphere (pressure boundary condition)
- 4) Single-phase flow over a sphere (velocity boundary condition)
- 5) Single-phase flow in a T-junction.