Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 7 Issue 9, September - 2020

KYAMOS Multiphysics Software Lattice Boltzmann Solver

Antonis P. Papadakis
KYAMOS LTD, 37 Polyneikis Street, Strovolos, 2047, Nicosia, Cyprus
ceo@kyamosmultiphysics.com

***Abstract*—KYAMOS LTD is a newly developed start-up company in the Computer Aided Engineering industry which aims to develop sophisticated multiphysics software that will provide improved and enhanced simulation experience into the design process of academics, researchers, engineers and industrialists. In this paper, we describe KYAMOS software attributes, as well as the development of a new innovative Lattice Boltzmann equation solver developed within an InfiniBand distributed GPU computing cluster. Lattice Boltzmann solvers are well known for treating boundary conditions with ease and are highly suitable for parallel processing. Hence, an InfiniBand GPU cluster is customary developed in operating system Linux/CENTOS. Its purpose is through cluster computing, to provide considerable computational power to every client to solve large problems with the desired speed and accuracy, having deployed a very friendly Graphical User Interface environment. Most importantly, KYAMOS software is based on in house developed algorithms and ready-made, well tested modules. Benchmark results for the time dependent, lid driven cavity flow test case is presented and discussed. It is found that the GPU based LB solver is approximately 25 times faster than an identical serial solver developed using competitor's software.**

Keywords—KYAMOS software, High Performance Computing, CUDA, Multiphysics, GPU;

I. INTRODUCTION

The Computer Aided Engineering (CAE) market is currently €6 billion and is growing at an annual growth rate of 8-10% and it is expected to reach €12 Billion by 2025. In the industry, there is lack of: (a) freely available multiphysics software with friendly GUI, and (b) cheap commercial multiphysics software. Furthermore, there is a lack of cloud-based, distributed GPU computing services which inevitably can offer much improved simulations in speed and accuracy, by offering high computing performance in the petascale with a potential to exascale computing. CAE analysts and CEOs of engineering leading companies believe that the expected growth rate is driven by new potential customers, possibly non-engineers or scientists, outside the research and development (R&D) departments of engineering companies that could potentially run multiphysics simulations. KYAMOS LTD is a Cyprus based company that wishes to try and close that gap, to the extent possible, by offering ready-made customized modules that will attract/tempt these scientists or non-engineers, to run multiphysics simulations for engineering problems, hence establishing itself in the market. Most of our competitors are US based companies with much higher average salaries, which gives us a clear advantage and the fact that our Serviceable Available Market is Europe, with a size of €3 Billion (50% of the total market).

Lattice Boltzmann (LB) solvers have recently gained increased attention and many papers are published regarding the capabilities of this method, when compared to the traditional finite difference, finite volume and finite elements methods. Specifically, literature states that the gap between the methods has closed and that there are clear advantages in some cases to the LB method, mostly due to its highly explicit scheme and highly parallelable nature, driven by the latest advancements of NVIDIA technology and GPU throughput utilization. Some of our main competitors are already utilizing LB equations in compressible flows in order to provide near real-live simulations on single computers. We wish to test this technology on cloud-based distributed GPU computing that will unleash its real capabilities through CUDA aware MPI.

In the long run, we intend to develop an experimental proof of concept that LB can be potentially used for simulations (highly responsive or even real time), when compared to traditional NS equation solutions and to validate this technology by utilizing cloud-based, distributed GPU in allied disciplines, such as biology, medicine and material science.

KYAMOS software is built on well established, open source and license free software such as OpenCascade as geometry editor, NETGEN as mesh editor and plPlot as viewing editor to achieve maximum impact with minimal cost. It is a cross-platform software and is expected to be a game changing concept due to its low cost, speed and accuracy of simulations.

KYAMOS software main objectives are as follows:

- Develop solvers for solving LB equations
- Embed the best freely available LB solvers in the market, if any, that can be used for proprietary software
- Examine compressible and incompressible flow performance of the LB equations

- Examine single and multiphase flow performance of the LB equations
- Deploy numerical mathematical algorithms well-proven for superior accuracy and performance
- Support cloud-based, distributed GPU solutions
- Support time-dependent solutions
- Test the ability of GPU simulations to provide near real-live insight simulations for LB equations
- Support 2D and 3D-simulations
- Support CUDA aware MPI for superior performance (peer to peer (P2P) and Remote Directory Memory Access (RDMA))
- Utilize external cloud computing services such as aws-amazon, if need be
- Test LB solvers performance through the InfiniBand network and ability to scale
- Perform benchmark and analytical solutions test cases to discover the limits, strengths and bottlenecks of the software and hardware
- Compare the LB equations with the NS equations in terms of speed and accuracy
- Identify different LB solving methods which have superiority in different types of flows
- Optimize the developed software to take into account attributes of the local, shared and global memory of the GPU
- Minimize communication of the GPUs with the CPU by performing almost all calculations on the GPU
- Avoid read/writing into and out of the host hard disk to avoid unnecessary computational delays
- Automate the treatment of initial and boundary conditions in LB solutions
- Scalability test of LB equations with multiple distributed GPUs
- Test shock capturing capability of LB methods using well known benchmark test cases such as the square wave propagation
- Validate superiority of LB equations on complex geometries and porous media
- Validate suitability in low Mach and difficulty in high Mach number compressible flows
- Test NVIDIA Tesla K80 GPUs' appropriateness to conduct LB parallel simulations
- Apply turbulent flow solutions for COVID-19 particle propagation (a governmental grant has been secured and project will start in October 2020).

KYAMOS software is anticipated to be released as a Minimum Viable Product (MVP) beginning of 2021.

II. Literature Review for Lattice Boltzmann

A. Existing knowledge

The LB methods have received immense interest recently when compared to the traditional finite difference, finite element and finite volume methods. The LB equations provide a new attractive alternative which is generally applicable to transient flows. Instead of being based on a continuum assumption of the macroscopic fluid properties, it uses a mesoscopic kinetic description from the Boltzmann equation. In contrast to the traditional methods, there is no need for the solution of matrices which are ill-conditioned, since it is typically decomposed into two schemes, a non-linear purely local scheme and a linear non-local scheme, which is ideal for parallelization, especially on the GPUs.

A review from Perumal et al. [1] states that the LB method has excellent numerical stability and constitutive versatility, hence receiving tremendous impetus with its spectacular use in incompressible and compressible fluid flow and heat transfer problems. From a computational point of view, LB equations can be solved locally, explicitly, and efficiently on parallel computers.

A conference paper by Manhartsgruber [2] states that the LB method has become an interesting alternative to classical finite volume based discretization methods because the flow domain is not meshed in the classical sense, but only voxelized and geometrically complex boundaries can be introduced in an easy form by bounce-back or off-Lattice boundary conditions. The above method works very well to simulations of channel flows inside hydraulic components.

Ubertini et al. [3] develop a new finite volume discretization of a generalized LB equation on unstructured grids. They create a new scheme named the unstructured LB equation with memory and show that it advances in time with larger time steps and producing similar results when compared to traditional LB solvers.

Succi et al. [4] performed a review on the mesoscopic modeling of water-like fluids using the LB method. By enriching the basic LB hydrodynamics with angular degrees of freedom responding to suitable directional potentials between water-like molecules, the model is able to reproduce microscopic features of liquid water and shows promising when coupled with dynamics of suspended bodies, for the simulation of complex biofluidic problems.

Li et al. [5] claim that there has been tremendous progress in the development of particle-based discrete simulation methods versus the conventional continuum-based methods. In particular, the LB method has evolved from a theoretical novelty to a ubiquitous, versatile and powerful computational methodology for both fundamental research and engineering applications. Applications of the LB method are now found in a wide range of disciplines including physics, chemistry, materials, biomedicine and various branches of engineering. The present

work provides a comprehensive review of the LB method for thermofluids and energy applications, focusing on multiphase flows, thermal flows and thermal multiphase flows with phase change. Examples of applications are provided in fuel cells and batteries, droplet collision, boiling heat transfer and evaporation, and energy storage. Finally, further developments and future prospect of the LB method are outlined for thermofluids and energy applications.

Jahanshallo et al. [6] perform a review on the turbulence flow application of the LB in Direct Numerical Simulations (DNS), Large Eddy Simulations (LES) and Reynolds Average Navier-Stokes (RANS) methods which show similar performance with the NS equations counterparts and are more convenient to implement due to their parallelable nature.

Chai et al. [7] develop a multiple relaxation time LB model (MRT) for non-linear anisotropic convection-diffusion equations and test it through analytical results and benchmark test cases. They show that the convergence rate of MRT in space is second order when compared to its first order counterpart, the single relaxation time LB model (BGK).

The LB equation is used by Sidic et al. [8] to study magnetohydrodynamic flow utilizing Cu–water nanofluid in a concentric annulus and found to be in excellent agreement with other numerical methods. He et al. [9] state that the LB method has come a long way since its introduction 30 years ago and has achieved great success in modeling transport phenomena involving complex boundaries and interfacial dynamics. The LB method can be considered to be an efficient numerical tool for fluid flow and heat transfer in porous media. Moreover, since LB method is inherently transient, it is especially useful for investigating transient solid-liquid phase-change processes, wherein the interfacial behaviors are very important.

Buick et al. [10] review the LB technique with particular emphasis on its application to biological systems. Further, they consider its application to arterial flows and discuss its potential for simulating flow on length-scales, where traditional numerical approaches can be troublesome. Finally, they present results from a preliminary investigation which demonstrates the suitability of the LB model for simulating oscillatory flows. Liu et al. [11] claim that LB methods have become a popular technique in the flow of complex geometries such as porous media. They identify a number of extensions able to study multiphase and multicomponent flows on a pore scale level.

Dhuri et al. [12] use the LB method to simulate linear acoustic wave propagation in heterogeneous media by deploying the single relaxation time Bhatnagar-Gross-Krook and the multirelaxation time collision operators and show that the LB performance is comparable with the classic second order finite difference scheme. Since the LB method is friendly to parallelization, it can be a cost-effective alternative for the simulation of linear acoustic waves in complex geometries and multiphase media.

Musavi et al. [13] simulate using a mesh-free LB method for the solution of geometrically complex fluid flow problems. They consider the streaming equation as a pure advection equation rather than a perfect shift, so that the physical space discretization becomes independent of the lattice. Two benchmark problems, namely the Poiseuille flow and the lid-driven cavity flow are used for validation purposes of the proposed method, where the results show that the proposed method outperforms the conventional LB method in the simulation of geometrically complicated flows.

To conclude, we anticipate that the usage of state-of-the-art hardware for LB simulations that supports CUDA aware MPI with cloud-based, distributed GPU solutions will outstrip its conventional software counterparts that utilize in the best-case scenario desktop type GPU cards and pave the way for near, real-live simulations of large computationally intensive problems.

B. Theoretical comparison between NS and LB solvers

The simulation of any three-dimensional phenomena relies heavily on the partial differential equations, which are the Navier-Stokes Equations (NSE). Another prospective equation is the Lattice Boltzmann Equation (LBE). In this paper, a theoretical comparison is conducted with mathematical equations and a table comparison representation is shown in Table 1 below.

The solution of partial differential equations can be obtained analytically and numerically. An analytical solution cannot always be implemented due to non-linearity terms or complex boundary conditions. Alternatively, NSE and LBE can be solved numerically through discretization schemes. Many methods have been invented to solve the NSE using different discretizing techniques, such as the finite difference, finite volume and finite element methods. These techniques convert the NSE into a system of algebraic equations that can be solved iteratively. However, the complexity of the phenomena leads to a complex NSE, leading to difficulties when solving it with the aforementioned methods. This equation can generally be solved in discretized form. In contrast, LBM has only one discretization technique, which converts the LBE into a system of algebraic equations. This paper will concentrate on the general characteristics of both NSE and LBE, in order to identify the main factors in each equation which may ease or complicate these equations.

C. Navier-Stokes equations

The complexity of the NSE depends on the type of flow, whether it is a steady state or unsteady state, compressible/incompressible, which lead to different branches of the NSE. The general form of incompressible flow can be written as follows:

$$\rho\frac{\partial \boldsymbol{u}}{\partial t}+\rho(\boldsymbol{u}.\nabla)\boldsymbol{u}-\mu\nabla^2\boldsymbol{u}=-\rho\nabla p+\rho\boldsymbol{g} \quad (1)$$

where ρ represents the density, **u** represents the velocity vector, μ the kinematic viscosity, p is the pressure which is treated as an internal source, **g** is a vector representing the body acceleration, which multiplied by the density represent an external force term.

The first term is linear and represents the changing of the velocity with respect to time (unsteady state), the second term represents the convective term which is a non-linear term, the third term is the diffusion term with constant viscosity. If the viscosity is not constant, this term becomes non-linear and eventually complicates the NSE solution. On the right-hand side, the pressure term and the body gravity are linear terms. By expanding equation "1" into three dimensions, we get the following:

$$\rho\begin{pmatrix}\frac{\partial u_x}{\partial t}\\ \frac{\partial u_y}{\partial t}\\ \frac{\partial u_z}{\partial t}\end{pmatrix}+\rho\left[\begin{pmatrix}u_x\\u_y\\u_z\end{pmatrix}\bullet\begin{pmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\\ \frac{\partial}{\partial z}\end{pmatrix}\right]\begin{pmatrix}u_x\\u_y\\u_z\end{pmatrix}$$

$$-\mu\left[\begin{pmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\\ \frac{\partial}{\partial z}\end{pmatrix}\bullet\begin{pmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\\ \frac{\partial}{\partial z}\end{pmatrix}\right]\begin{pmatrix}u_x\\u_y\\u_z\end{pmatrix}=-\rho\begin{pmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\\ \frac{\partial}{\partial z}\end{pmatrix}p+\rho\begin{pmatrix}g_x\\g_y\\g_z\end{pmatrix} \quad (2)$$

By expanding the dot product:

$$\rho\begin{pmatrix}\frac{\partial u_x}{\partial t}\\ \frac{\partial u_y}{\partial t}\\ \frac{\partial u_z}{\partial t}\end{pmatrix}+\rho\left[u_x\frac{\partial}{\partial x}+u_y\frac{\partial}{\partial y}+u_z\frac{\partial}{\partial z}\right]\begin{pmatrix}u_x\\u_y\\u_z\end{pmatrix}$$

$$-\mu\left[\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}+\frac{\partial^2}{\partial z^2}\right]\begin{pmatrix}u_x\\u_y\\u_z\end{pmatrix}=-\rho\begin{pmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\\ \frac{\partial}{\partial z}\end{pmatrix}p+\rho\begin{pmatrix}g_x\\g_y\\g_z\end{pmatrix} \quad (3)$$

By inserting the scalars:

$$\begin{pmatrix}\rho\frac{\partial u_x}{\partial t}\\ \rho\frac{\partial u_y}{\partial t}\\ \rho\frac{\partial u_z}{\partial t}\end{pmatrix}+\begin{pmatrix}\rho u_x\frac{\partial u_x}{\partial x}+\rho u_y\frac{\partial u_x}{\partial y}+\rho u_z\frac{\partial u_x}{\partial z}\\ \rho u_x\frac{\partial u_y}{\partial x}+\rho u_y\frac{\partial u_y}{\partial y}+\rho u_z\frac{\partial u_y}{\partial z}\\ \rho u_x\frac{\partial u_z}{\partial x}+\rho u_y\frac{\partial u_z}{\partial y}+\rho u_z\frac{\partial u_z}{\partial z}\end{pmatrix}$$

$$-\begin{pmatrix}\mu\frac{\partial^2 u_x}{\partial x^2}+\mu\frac{\partial^2 u_x}{\partial y^2}+\mu\frac{\partial^2 u_x}{\partial z^2}\\ \mu\frac{\partial^2 u_y}{\partial x^2}+\mu\frac{\partial^2 u_y}{\partial y^2}+\mu\frac{\partial^2 u_y}{\partial z^2}\\ \mu\frac{\partial^2 u_z}{\partial x^2}+\mu\frac{\partial^2 u_z}{\partial y^2}+\mu\frac{\partial^2 u_z}{\partial z^2}\end{pmatrix}=\begin{pmatrix}-\rho\frac{\partial p}{\partial x}\\ -\rho\frac{\partial p}{\partial y}\\ -\rho\frac{\partial p}{\partial z}\end{pmatrix}+\begin{pmatrix}\rho g_x\\ \rho g_y\\ \rho g_z\end{pmatrix} \quad (4)$$

Then it becomes:

$$\rho\frac{\partial u_x}{\partial t}+\rho u_x\frac{\partial u_x}{\partial x}+\rho u_y\frac{\partial u_x}{\partial y}+\rho u_z\frac{\partial u_x}{\partial z}=$$
$$-\rho\frac{\partial p}{\partial x}+\mu\frac{\partial^2 u_x}{\partial x^2}+\mu\frac{\partial^2 u_x}{\partial y^2}+\mu\frac{\partial^2 u_x}{\partial z^2}+\rho g_x \quad (5)$$

$$\rho\frac{\partial u_y}{\partial t}+\rho u_x\frac{\partial u_y}{\partial x}+\rho u_y\frac{\partial u_y}{\partial y}+\rho u_z\frac{\partial u_y}{\partial z}=$$
$$-\rho\frac{\partial p}{\partial y}+\mu\frac{\partial^2 u_y}{\partial x^2}+\mu\frac{\partial^2 u_y}{\partial y^2}+\mu\frac{\partial^2 u_y}{\partial z^2}+\rho g_y \quad (6)$$

$$\rho\frac{\partial u_z}{\partial t}+\rho u_x\frac{\partial u_z}{\partial x}+\rho u_y\frac{\partial u_z}{\partial y}+\rho u_z\frac{\partial u_z}{\partial z}=$$
$$-\rho\frac{\partial p}{\partial z}+\mu\frac{\partial^2 u_z}{\partial x^2}+\mu\frac{\partial^2 u_z}{\partial y^2}+\mu\frac{\partial^2 u_z}{\partial z^2}+\rho g_z \quad (7)$$

One can see that equations "5-7", have each two dependent variables, which makes the solution more complicated. Furthermore, an explicit equation for the pressure term cannot be obtained.

Hence, an explicit equation for the velocity can be obtained by using the continuity equation for incompressible flow as follows:

$$\frac{\partial\rho}{\partial t}+\frac{\partial\rho u_x}{\partial x}+\frac{\partial\rho u_y}{\partial y}+\frac{\partial\rho u_z}{\partial z}=0 \quad (8)$$

The density can be assumed constant if the fluid is incompressible. This assumption is valid if the Mach number does not exceed 0.3. Equation "8" can be reduced as follows:

$$\frac{\partial u_x}{\partial x}+\frac{\partial u_y}{\partial y}+\frac{\partial u_z}{\partial z}=0 \quad (9)$$

which can be written as:

$$(\nabla\cdot)\boldsymbol{u}=0 \quad (10)$$

Equation "10" represents the explicit form of the velocity, which has been obtained under certain conditions.

For compressible flow, the most general form is:

$$\rho\frac{\partial \boldsymbol{u}}{\partial t}+\rho(\boldsymbol{u}.\nabla)\boldsymbol{u}-\mu\nabla^2\boldsymbol{u}=-\rho\nabla P+\frac{1}{3}\mu\nabla(\nabla.\boldsymbol{u})+\rho\boldsymbol{g} \quad (11)$$

where P is the mechanical pressure.

We can follow a similar procedure used for expanding the incompressible flow to study the compressible flow behavior:

$$\rho\begin{pmatrix}\frac{\partial u_x}{\partial t}\\ \frac{\partial u_y}{\partial t}\\ \frac{\partial u_z}{\partial t}\end{pmatrix}+\rho\left[\begin{pmatrix}u_x\\u_y\\u_z\end{pmatrix}\bullet\begin{pmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\\ \frac{\partial}{\partial z}\end{pmatrix}\right]\begin{pmatrix}u_x\\u_y\\u_z\end{pmatrix}-$$
$$\mu\left[\begin{pmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\\ \frac{\partial}{\partial z}\end{pmatrix}\bullet\begin{pmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\\ \frac{\partial}{\partial z}\end{pmatrix}\right]\begin{pmatrix}u_x\\u_y\\u_z\end{pmatrix}=-\rho\begin{pmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\\ \frac{\partial}{\partial z}\end{pmatrix}P+$$
$$\frac{1}{3}\mu\begin{pmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\\ \frac{\partial}{\partial z}\end{pmatrix}\left[\begin{pmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\\ \frac{\partial}{\partial z}\end{pmatrix}\bullet\begin{pmatrix}u_x\\u_y\\u_z\end{pmatrix}\right]+\rho\begin{pmatrix}g_x\\g_y\\g_z\end{pmatrix} \quad (12)$$

By expanding the dot product:

$$\rho\begin{pmatrix}\frac{\partial u_x}{\partial t}\\ \frac{\partial u_y}{\partial t}\\ \frac{\partial u_z}{\partial t}\end{pmatrix}+\rho\left[u_x\frac{\partial}{\partial x}+u_y\frac{\partial}{\partial y}+u_z\frac{\partial}{\partial z}\right]\begin{pmatrix}u_x\\ u_y\\ u_z\end{pmatrix}-$$

$$\mu\left[\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}+\frac{\partial^2}{\partial z^2}\right]\begin{pmatrix}u_x\\ u_y\\ u_z\end{pmatrix}=-\rho\begin{pmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\\ \frac{\partial}{\partial z}\end{pmatrix}P+$$

$$\frac{1}{3}\mu\begin{pmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\\ \frac{\partial}{\partial z}\end{pmatrix}\left[\frac{\partial u_x}{\partial x}+\frac{\partial u_y}{\partial y}+\frac{\partial u_z}{\partial z}\right]+\rho\begin{pmatrix}g_x\\ g_y\\ g_z\end{pmatrix} \quad (13)$$

By inserting the scalars:

$$\begin{pmatrix}\rho\frac{\partial u_x}{\partial t}\\ \rho\frac{\partial u_y}{\partial t}\\ \rho\frac{\partial u_z}{\partial t}\end{pmatrix}+\begin{pmatrix}\rho u_x\frac{\partial u_x}{\partial x}+\rho u_y\frac{\partial u_x}{\partial y}+\rho u_z\frac{\partial u_x}{\partial z}\\ \rho u_x\frac{\partial u_y}{\partial x}+\rho u_y\frac{\partial u_y}{\partial y}+\rho u_z\frac{\partial u_y}{\partial z}\\ \rho u_x\frac{\partial u_z}{\partial x}+\rho u_y\frac{\partial u_z}{\partial y}+\rho u_z\frac{\partial u_z}{\partial z}\end{pmatrix}$$

$$-\begin{pmatrix}\mu\frac{\partial^2 u_x}{\partial x^2}+\mu\frac{\partial^2 u_x}{\partial y^2}+\mu\frac{\partial^2 u_x}{\partial z^2}\\ \mu\frac{\partial^2 u_y}{\partial x^2}+\mu\frac{\partial^2 u_y}{\partial y^2}+\mu\frac{\partial^2 u_y}{\partial z^2}\\ \mu\frac{\partial^2 u_z}{\partial x^2}+\mu\frac{\partial^2 u_z}{\partial y^2}+\mu\frac{\partial^2 u_z}{\partial z^2}\end{pmatrix}=\begin{pmatrix}-\rho\frac{\partial P}{\partial x}\\ -\rho\frac{\partial P}{\partial y}\\ -\rho\frac{\partial P}{\partial z}\end{pmatrix}+$$

$$\begin{pmatrix}\frac{1}{3}\mu\frac{\partial}{\partial x}\left[\frac{\partial u_x}{\partial x}+\frac{\partial u_y}{\partial y}+\frac{\partial u_z}{\partial z}\right]\\ \frac{1}{3}\mu\frac{\partial}{\partial y}\left[\frac{\partial u_x}{\partial x}+\frac{\partial u_y}{\partial y}+\frac{\partial u_z}{\partial z}\right]\\ \frac{1}{3}\mu\frac{\partial}{\partial z}\left[\frac{\partial u_x}{\partial x}+\frac{\partial u_y}{\partial y}+\frac{\partial u_z}{\partial z}\right]\end{pmatrix}+\begin{pmatrix}\rho g_x\\ \rho g_y\\ \rho g_z\end{pmatrix} \quad (14)$$

$$\rho\frac{\partial u_x}{\partial t}+\rho u_x\frac{\partial u_x}{\partial x}+\rho u_y\frac{\partial u_x}{\partial y}+\rho u_z\frac{\partial u_x}{\partial z}=$$
$$-\rho\frac{\partial P}{\partial x}+\mu\frac{\partial^2 u_x}{\partial x^2}+\mu\frac{\partial^2 u_x}{\partial y^2}+\mu\frac{\partial^2 u_x}{\partial z^2}+$$
$$\frac{1}{3}\mu\frac{\partial}{\partial x}\left[\frac{\partial u_x}{\partial x}+\frac{\partial u_y}{\partial y}+\frac{\partial u_z}{\partial z}\right]+\rho g_x \quad (15)$$

$$\rho\frac{\partial u_y}{\partial t}+\rho u_x\frac{\partial u_y}{\partial x}+\rho u_y\frac{\partial u_y}{\partial y}+\rho u_z\frac{\partial u_y}{\partial z}=$$
$$-\rho\frac{\partial P}{\partial y}+\mu\frac{\partial^2 u_y}{\partial x^2}+\mu\frac{\partial^2 u_y}{\partial y^2}+\mu\frac{\partial^2 u_y}{\partial z^2}+$$
$$\frac{1}{3}\mu\frac{\partial}{\partial y}\left[\frac{\partial u_x}{\partial x}+\frac{\partial u_y}{\partial y}+\frac{\partial u_z}{\partial z}\right]+\rho g_y \quad (16)$$

$$\rho\frac{\partial u_z}{\partial t}+\rho u_x\frac{\partial u_z}{\partial x}+\rho u_y\frac{\partial u_z}{\partial y}+\rho u_z\frac{\partial u_z}{\partial z}=$$
$$-\rho\frac{\partial P}{\partial z}+\mu\frac{\partial^2 u_z}{\partial x^2}+\mu\frac{\partial^2 u_z}{\partial y^2}+\mu\frac{\partial^2 u_z}{\partial z^2}+$$
$$\frac{1}{3}\mu\frac{\partial}{\partial z}\left[\frac{\partial u_x}{\partial x}+\frac{\partial u_y}{\partial y}+\frac{\partial u_z}{\partial z}\right]+\rho g_z \quad (17)$$

The equation becomes more complicated due to the addition of the new viscosity term on the right-hand side. Moreover, the mechanical pressure will complicate the equation as well. For simplicity, the mechanical pressure will be assumed as a thermodynamic pressure. Additionally, the continuity equation adds another complexity, since one cannot get an explicit equation for the velocity because the density cannot be constant.

D. Lattice Boltzmann Equation

The LBE emerged as an alternative equation to solving physical phenomena and engineering problems. As in NSE, the mathematical formulation of the equation depends on the model.

1) The Boltzmann Transport Equation

The formulation of the LBE is based on statistical mechanics. Statistical mechanics are a way to translate the microscopic to macroscopic properties of a fluid, by using the probability to describe the density of particles in a specific range of location, having the velocities in a certain range at a specific time. This is called distribution function and has the notation $\boldsymbol{f}$.

The distribution function is the number of particles per volume per cubed velocity. Mathematically, it can be represented as a function of three variables: location, velocity and time, so it can be written as:

$$\boldsymbol{f(x, c_x, t)} \quad (18)$$

where x is the location if it is one dimension, c_x is the velocity in the x-direction, and t is the time. In three dimensions, it can be written in a generic form:

$$\boldsymbol{f(x, y, z, c_x, c_y, c_z, t)} \quad (19)$$

If we imagine a force acting on a cluster of particles, the distribution function will be changed. Fig. 1 below describes the distribution function before and after a force has acted on a particle.

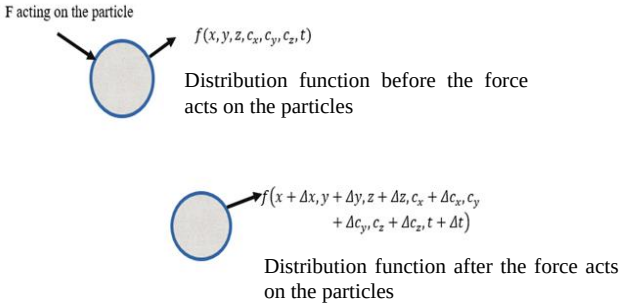

Fig. 1. Distribution functions before and after a force acts on the particles

If there is no collision to take place between the particles, the number of particles before and after will not change. However, collisions do exist between the particles and the rate of change of the distribution function is called the collision operator $\boldsymbol{\Omega(f)}$. The equation can be written mathematically as:

$$f(x + \Delta x, y + \Delta y, z + \Delta z, c_x + \Delta c_x, c_y + \Delta c_y, c_z + \Delta c_z, t + \Delta t)\,\Delta x\Delta y\Delta z\Delta c_x\Delta c_y\Delta c_z -$$
$$f(x, y, z, c_x, c_y, c_z, t)\Delta x\Delta y\Delta z\Delta c_x\Delta c_y\Delta c_z =$$
$$\boldsymbol{\Omega(f)}\Delta x\Delta y\Delta z\Delta c_x\Delta c_y\Delta c_z\Delta t \quad (20)$$

Dividing both sides by $\boldsymbol{\Omega(f)}\Delta x\Delta y\Delta z\Delta c_x\Delta c_y\Delta c_z\Delta t$ and taking the limit of the left side $\Delta t \to 0$:

$$\lim_{\Delta t\to 0}\frac{f(x+\Delta x,y+\Delta y,z+\Delta z,c_x+\Delta c_x,c_y+\Delta c_y,c_z+\Delta c_z,t+\Delta t)-f(x,y,z,c_x,c_y,c_z,t)}{\Delta t} \quad (21)$$

The changing in the x, y and z-directions can be written as:

$$\Delta x = c_x\Delta t, \Delta y = c_y\Delta t, \Delta z = c_z\Delta t \quad (22)$$

From Newton's second law, one obtains these relations:

$$\Delta c_x = \frac{F_x}{m}\Delta t, \Delta c_y = \frac{F_y}{m}\Delta t, \Delta c_z = \frac{F_z}{m}\Delta t \quad (23)$$

Substituting equation "22" and "23" into "21" to satisfy the necessity of the limit:

$$\lim_{\Delta t\to 0}\frac{f\left(x+\Delta x,y+\Delta y,z+\Delta z,c_x+\frac{F_x}{m}\Delta t,,c_y+\frac{F_y}{m}\Delta t,c_z+\frac{F_z}{m}\Delta t,t+\Delta t\right)-f(x,y,z,c_x,c_y,c_z,t)}{\Delta t} \quad (24)$$

leads to:

$$\frac{df}{dt} = \boldsymbol{\Omega}(\boldsymbol{f}) \quad (25)$$

Since $\boldsymbol{f}$ is a function of space, velocity and time, the derivative of $\boldsymbol{f}$ with respect to time can be expanded as follows:

$$\frac{\partial f}{\partial t}+\frac{\partial f}{\partial x}\frac{\partial x}{\partial t}+\frac{\partial f}{\partial y}\frac{\partial y}{\partial t}+\frac{\partial f}{\partial z}\frac{\partial z}{\partial t}+\frac{\partial f}{\partial c_x}\frac{\partial c_x}{\partial t}+\frac{\partial f}{\partial c_y}\frac{\partial c_y}{\partial t}+\frac{\partial f}{\partial c_z}\frac{\partial c_z}{\partial t} = \boldsymbol{\Omega}(\boldsymbol{f}) \quad (26)$$

where the partial derivative of x, y and z with respect to time is equal to the velocity of the particles:

$$\frac{\partial x}{\partial t} = c_x, \frac{\partial y}{\partial t} = c_y, \frac{\partial z}{\partial t} = c_z \quad (27)$$

The partial derivative with respect to the velocity of the particles in x, y and z-directions are equal to the acceleration of particles and from Newton's second law, one obtains these relations:

$$\frac{\partial c_x}{\partial t} = \frac{F_x}{m}, \frac{\partial c_y}{\partial t} = \frac{F_y}{m}, \frac{\partial c_z}{\partial t} = \frac{F_z}{m} \quad (28)$$

where the $F_x, F_y\ F_z$ is the force in x, y and z-directions and m is the mass of the particles. Substituting equation "27" and "28" into "26", we obtain:

$$\frac{\partial f}{\partial t}+c_x\frac{\partial f}{\partial x}+c_y\frac{\partial f}{\partial y}+c_z\frac{\partial f}{\partial z}+\frac{F_x}{m}\frac{\partial f}{\partial c_x}+\frac{F_y}{m}\frac{\partial f}{\partial c_y}+\frac{F_z}{m}\frac{\partial f}{\partial c_z} = \boldsymbol{\Omega}(\boldsymbol{f}) \quad (29)$$

Conveniently, to write the velocity, as well as the force as a vector to reduce the shape of the equation, we consider the velocity and the force as vectors:

$$\overrightarrow{\boldsymbol{F}} = F_x i + F_y j + F_z k, \overrightarrow{\boldsymbol{c}} = c_x i + c_y j + c_z k \quad (30)$$

We can then use the del notation:

$$\boldsymbol{\nabla} = \frac{\partial}{\partial x}\boldsymbol{i}+\frac{\partial}{\partial y}\boldsymbol{j}+\frac{\partial}{\partial z}\boldsymbol{k} \quad (31)$$

Additionally, we can assume:

$$\frac{\partial}{\partial \boldsymbol{c}} = \frac{\partial}{\partial c_x}\boldsymbol{i}+\frac{\partial}{\partial c_y}\boldsymbol{j}+\frac{\partial}{\partial c_z}\boldsymbol{k} \quad (32)$$

Substituting equation "30", "31" and "32" into "29" and taking the dot product between the del and the vector, leads to:

$$\frac{\partial \boldsymbol{f}}{\partial t}+\boldsymbol{c}.\nabla\boldsymbol{f}+\frac{\boldsymbol{F}}{m}.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{c}} = \boldsymbol{\Omega}(\boldsymbol{f}) \quad (33)$$

Equation "33" is the generic form of the Boltzmann transport equation in continuum form, the first term on the left-hand side describes the changing of the distribution function with time, the second term represents the changing of the distribution function with space, an advection term with the effect of the velocity, the third term represents the changing of the distribution function with the velocity, as well as the force acting on it. On the right-hand side, the collision operator is a function of the distribution function and represents the source term.

For a phenomenon which does not have a force acting on it, we can eliminate the force term and the equation simplifies to:

$$\frac{\partial \boldsymbol{f}}{\partial t}+\boldsymbol{c}.\nabla\boldsymbol{f} = \boldsymbol{\Omega}(\boldsymbol{f}) \quad (34)$$

The relation between the microscopic properties like the velocity of particles and the energy of particles with the macroscopic properties like the density of the physical parameters (temperature, fluid, magnetic field and other parameters), the momentum of the physical parameters and the energy of the physical parameters can be illustrated mathematically as follows:

$$\rho = \int m\boldsymbol{f}\, d\boldsymbol{c} \quad (35)$$

$$\rho\boldsymbol{u} = \int m\boldsymbol{c}\boldsymbol{f}\, d\boldsymbol{c} \quad (36)$$

$$\rho e = \frac{1}{2}\int m\boldsymbol{v}^2\boldsymbol{f}\, d\boldsymbol{c} \quad (37)$$

where ρ is the density, **u** is the macroscopic velocity, e is the internal energy and the **v** is the particle velocity relative to the macroscopic velocity, which is equal the particle velocity minus the macroscopic velocity as follows:

$$\boldsymbol{v} = \boldsymbol{c} - \boldsymbol{u} \quad (38)$$

where **v**, **c**, **u** are vectors.

Equations "35-38" are the workhorse for solving a system without external force. If the collision operator is known in a simple form, the LB solution can be obtained through various methods. However, the collision operator is not known and needs to be determined approximately. The popular method to find approximated collision operator is Bhatnagar, Gross and Krook (BGK) approximation.

2) BGK Approximation

The three scientists Bhatnagar, Gross and Krook introduced a very good approximation which leads to a very good outcome with little error:

$$\boldsymbol{\Omega}(\boldsymbol{f}) = \frac{1}{\tau}(\boldsymbol{f}^{eq}-\boldsymbol{f}) \quad (39)$$

where $\boldsymbol{f}^{eq}$ represents the local equilibrium distribution function and τ is the relaxation factor.

Additionally, it is in a very simple form, the simplicity comes from the linearity of the form which simplifies the Boltzmann transport equation's solution.

3) Discretization of the Boltzmann Transport Equation

The Boltzmann transport equation has been discretized according to the traveling path of the particles. As the particles travel in a straight line, with the assumption of limited particles at each node in the space (discretized space not continuum), this will create something similar to the shape of the lattice. If we consider the direction of the particles, the Boltzmann transport equation should be implemented for each direction separately, so it can be written in this form:

$$\frac{\partial f_k}{\partial t} + \boldsymbol{c}_k.\nabla \boldsymbol{f}_k = \frac{1}{\tau}\left(\boldsymbol{f}_k^{eq} - \boldsymbol{f}_k\right) \quad (40)$$

where k represents the number of the particles and each particle has a specific direction. Equation "40" can be discretized according to the direction of the particles, therefore, the equation will handle each direction separately as follows:

$$\frac{f_k(x,y,z,t+\Delta t)-f_k(x,y,z,t)}{\Delta t} + c_x\frac{f_k(x+\Delta x,y,z,t+\Delta t)-f_k(x,y,z,t+\Delta t)}{\Delta x} = \frac{1}{\tau}\left(f_k^{eq}(x,y,z,t) - f_k(x,y,z,t)\right) \quad (41)$$

$$\frac{f_k(x,y,z,t+\Delta t)-f_k(x,y,z,t)}{\Delta t} + c_y\frac{f_k(x,y+\Delta y,z,t+\Delta t)-f(x,y,z,t+\Delta t)}{\Delta y} = \frac{1}{\tau}\left(f_k^{eq}(x,y,z,t) - f_k(x,y,z,t)\right) \quad (42)$$

$$\frac{f_k(x,y,z,t+\Delta t)-f_k(x,y,z,t)}{\Delta t} + c_z\frac{f_k(x,y,z+\Delta z,t+\Delta t)-f_k(x,y,z,t+\Delta t)}{\Delta z} = \frac{1}{\tau}\left(f_k^{eq}(x,y,z,t) - f_k(x,y,z,t)\right) \quad (43)$$

In discretized form, the velocity in the x, y and z directions take this form:

$$\frac{\Delta x}{\Delta t} = c_x, \frac{\Delta y}{\Delta t} = c_y, \frac{\Delta z}{\Delta t} = c_z \quad (44)$$

Substituting equation "44" into "41", "42" and "43" leads to:

$$\frac{f_k(x,y,z,t+\Delta t)-f_k(x,y,z,t)}{\Delta t} + \frac{\Delta x}{\Delta t}\frac{f_k(x+\Delta x,y,z,t+\Delta t)-f_k(x,y,z,t+\Delta t)}{\Delta x} = \frac{1}{\tau}\left(f_k^{eq}(x,y,z,t) - f_k(x,y,z,t)\right) \quad (45)$$

$$\frac{f_k(x,y,z,t+\Delta t)-f_k(x,y,z,t)}{\Delta t} + \frac{\Delta y}{\Delta t}\frac{f_k(x,y+\Delta y,z,t+\Delta t)-f(x,y,z,t+\Delta t)}{\Delta y} = \frac{1}{\tau}\left(f_k^{eq}(x,y,z,t) - f_k(x,y,z,t)\right) \quad (46)$$

$$\frac{f_k(x,y,z,t+\Delta t)-f_k(x,y,z,t)}{\Delta t} + \frac{\Delta z}{\Delta t}\frac{f_k(x,y,z+\Delta z,t+\Delta t)-f_k(x,y,z,t+\Delta t)}{\Delta z} = \frac{1}{\tau}\left(f_k^{eq}(x,y,z,t) - f_k(x,y,z,t)\right) \quad (47)$$

Rearranging the above equations, one obtains:

$$f_k(x+\Delta x,y,z,t+\Delta t) = \left(1-\frac{\Delta t}{\tau}\right)f_k(x,y,z,t) + \frac{\Delta t}{\tau}f_k^{eq}(x,y,z,t) \quad (48)$$

$$f_k(x,y+\Delta y,z,t+\Delta t) = \left(1-\frac{\Delta t}{\tau}\right)f_k(x,y,z,t) + \frac{\Delta t}{\tau}f_k^{eq}(x,y,z,t) \quad (49)$$

$$f_k(x,y,z+\Delta z,t+\Delta t) = \left(1-\frac{\Delta t}{\tau}\right)f_k(x,y,z,t) + \frac{\Delta t}{\tau}f_k^{eq}(x,y,z,t) \quad (50)$$

Assuming:

$$\omega = \frac{\Delta t}{\tau} \quad (51)$$

Substituting equation "51" into "48", "49", and "50" leads to:

$$f_k(x+\Delta x,y,z,t+\Delta t) = (1-\omega)f_k(x,y,z,t) + \omega f_k^{eq}(x,y,z,t) \quad (52)$$

$$f_k(x,y+\Delta y,z,t+\Delta t) = (1-\omega)f_k(x,y,z,t) + \omega f_k^{eq}(x,y,z,t) \quad (53)$$

$$f_k(x,y,z+\Delta z,t+\Delta t) = (1-\omega)f_k(x,y,z,t) + \omega f_k^{eq}(x,y,z,t) \quad (54)$$

The new variable ω is called relaxation time. The relation between the microscopic properties and macroscopic can be written in discretized form as follows:

$$\rho = \sum_k \boldsymbol{f}_k \quad (55)$$

$$\rho\boldsymbol{u} = \sum_k \boldsymbol{c_k f_k} \quad (56)$$

$$\rho e = \frac{1}{2}\sum_k \boldsymbol{c_k^2} f_k \quad (57)$$

Two things need to be determined, the relaxation time and the equilibrium distribution function. The relaxation time can be determined according to the type of the system that one wants to solve. This is a crucial step to find the relation between ω with the kinetic viscosity or thermal diffusivity and other macroscopic parameters depend on the model. This can be done by Chapman–Enskog expansion method, where the equilibrium distribution function can be obtained by applying mass and momentum conservation with different assumption according to the model that needs to be solved, since the derivation of the relaxation time and equilibrium distribution function rely on the model of the system. Following, a 1D diffusion problem is used to illustrate the derivation.

4) Equilibrium Distribution Function for 1D Diffusion Problems

Assume that we have two particles at each node for one dimension, this is called D1Q2, where D1 refers to one dimension and Q2 refers to the number of particles which is two, so k equal two. One of the particles travels to the right and the other travels to the left. From mass conservation, one obtains:

$$\rho = \sum_2 \boldsymbol{f}_k^{eq} \quad (58)$$

which leads to:

$$\rho = f_1^{eq} + f_2^{eq} \quad (59)$$

Since we are dealing with diffusion problem, we do not have a macroscopic velocity, therefore, the momentum conservation is:

$$\sum_2 c_k f_k^{eq} = 0 \quad (60)$$

Generally, in the LB method, the velocity of the particle in the right and left direction can be assumed as:

$$c_1 = 1, c_2 = -1 \quad (61)$$

Substituting equation "61" into "60", one obtains:

$$f_1^{eq} - f_2^{eq} = 0 \quad (62)$$

Because macroscopic velocity is absent, we assume the equilibrium distribution function is constant:

$$f_1^{eq} = b_1, f_2^{eq} = b_2 \quad (63)$$

Substituting equation “63” into “62” and then into equation “59”, leads to:

$$b_1 = \frac{\rho}{2}, b_2 = \frac{\rho}{2} \quad (64)$$

Generally, this can be written as a weight function, as follows:

$$f_k^{eq} = w_k \rho \quad (65)$$

where w is the weight factor, for the D1Q2 is:

$$w_1 = \frac{1}{2}, w_2 = \frac{1}{2} \quad (66)$$

The weight factor will be changed depending on the number of particles and the dimensions.

5) Chapman-Enskog Expansion Method

Suppose that we want to find the distribution of the temperature along the x-axis and this distribution will be variant with the time, mathematically, it can be represented in this equation:

$$\frac{\partial T}{\partial t} = \Lambda \frac{\partial^2 T}{\partial x^2} \quad (67)$$

where T is the temperature and Λ is the diffusion coefficient.

We can solve equation “67” by using LBE:

$$f_k(x + \Delta x, y, z, t + \Delta t) = (1 - \omega) f_k(x, y, z, t) + \omega f_k^{eq}(x, y, z, t) \quad (68)$$

Because it is 1D, it can be written as:

$$f_k(x + \Delta x, t + \Delta t) = (1 - \omega) f_k(x, t) + \omega f_k^{eq}(x, t) \quad (69)$$

The LBE does not contain macroscopic parameters, hence one can obtain the temperature which is a macroscopic parameter by:

$$T = \sum_k f_k \quad (70)$$

However, we cannot cover, directly, the diffusion coefficient Λ. The main approach to solve this issue is to convert the shape of the LBE to the similar shape of equation “67” and to compare the terms. This can be achieved mathematically through Chapman- Enskog method.

Starting from equation “69”, expanding the first term on the left side by using Taylor series:

$$f_k(x + \Delta x, t + \Delta t) = \sum_{n=0}^{n=\infty} \frac{1}{n!} \left(\Delta x \frac{\partial}{\partial x} + \Delta t \frac{\partial}{\partial t} \right)^n f_k(x, t) \quad (71)$$

The Δx can be replaced by $\Delta t c_x$ and rescale the equation as follows:

$$\frac{\partial}{\partial x} = \frac{\varepsilon \partial}{\partial x^*}, \frac{\partial}{\partial t} = \frac{\varepsilon^2 \partial}{\partial t^*} \quad (72)$$

where ε is the Knudson number.

Substituting equation “72” into “71” and replacing Δx by $\Delta t c_x$ leads to:

$$f_k(x + \Delta t c_x, t + \Delta t) = \sum_{n=0}^{n=\infty} \frac{1}{n!} \left(\varepsilon \Delta t c_x \frac{\partial}{\partial x^*} + \varepsilon^2 \Delta t \frac{\partial}{\partial t^*} \right)^n f_k(x, t) \quad (73)$$

Truncating from equation “73”, the third order and on, gives:

$$f_k(x + \Delta t c_x, t + \Delta t) = f_k(x, t) + \varepsilon \Delta t c_x \frac{\partial f_k(x,t)}{\partial x^*} + \varepsilon^2 \Delta t \frac{\partial f_k(x,t)}{\partial t^*} + \frac{1}{2} \varepsilon^2 \Delta t^2 c_x^2 \frac{\partial^2 f_k(x,t)}{\partial x^{*2}} + \varepsilon^3 \Delta t^2 c_x \frac{\partial^2 f_k(x,t)}{\partial x^* \partial t^*} + \frac{1}{2} \varepsilon^4 \Delta t^2 \frac{\partial^2 f_k(x,t)}{\partial t^{*2}} \quad (74)$$

Substituting equation “74” into “70” and replacing ω by Δt/τ and rearranging it, one gets:

$$-\frac{1}{\tau} \left(f_k(x,t) - f_k^{eq}(x,t) \right) = \varepsilon c_x \frac{\partial f_k(x,t)}{\partial x^*} + \varepsilon^2 \frac{\partial f_k(x,t)}{\partial t^*} + \frac{1}{2} \varepsilon^2 \Delta t c_x^2 \frac{\partial^2 f_k(x,t)}{\partial x^{*2}} + \varepsilon^3 \Delta t c_x \frac{\partial^2 f_k(x,t)}{\partial x^* \partial t^*} + \frac{1}{2} \varepsilon^4 \Delta t \frac{\partial^2 f_k(x,t)}{\partial t^*} \quad (75)$$

Truncating the ε from the third order and on, one obtains:

$$-\frac{1}{\tau} \left(f_k(x,t) - f_k^{eq}(x,t) \right) = \varepsilon c_x \frac{\partial f_k(x,t)}{\partial x^*} + \varepsilon^2 \frac{\partial f_k(x,t)}{\partial t^*} + \frac{1}{2} \varepsilon^2 \Delta t c_x^2 \frac{\partial^2 f_k(x,t)}{\partial x^{*2}} \quad (76)$$

Expanding $\boldsymbol{f}$ by using perturbation series, one gets:

$$f_k(x,t) = f_k^{eq}(x,t) + \varepsilon f_k(x,t)^{(1)} + \varepsilon^2 f_k(x,t)^{(2)} + \cdots \quad (77)$$

Truncating the second order and on, then substituting equation “77” into “76”, leads to:

$$-\frac{1}{\tau} \varepsilon f_k(x,t)^{(1)} = \varepsilon c_x \frac{\partial f_k^{eq}(x,t)}{\partial x^*} + \varepsilon^2 c_x \frac{\partial f_k(x,t)^{(1)}}{\partial x^*} + \varepsilon^2 \frac{\partial f_k^{eq}(x,t)}{\partial t^*} + \varepsilon^3 \frac{\partial f_k(x,t)^{(1)}}{\partial t^*} + \frac{1}{2} \varepsilon^2 \Delta t c_x^2 \frac{\partial^2 f_k^{eq}(x,t)}{\partial x^{*2}} + \frac{1}{2} \varepsilon^3 \Delta t c_x^2 \frac{\partial^2 f_k(x,t)^{(1)}}{\partial x^{*2}} \quad (78)$$

Equating equation “78” according to the order of ε and neglecting the third order of ε, leads to:

$$-\frac{1}{\tau} \varepsilon f_k(x,t)^{(1)} = \varepsilon c_x \frac{\partial f_k^{eq}(x,t)}{\partial x^*} \quad (79)$$

$$0 = \varepsilon^2 c_x \frac{\partial f_k(x,t)^{(1)}}{\partial x^*} + \varepsilon^2 \frac{\partial f_k^{eq}(x,t)}{\partial t^*} + \frac{1}{2} \varepsilon^2 \Delta t c_x^2 \frac{\partial^2 f_k^{eq}(x,t)}{\partial x^{*2}} \quad (80)$$

By taking the partial derivative of equation “78” with respect to x, this will lead to:

$$\frac{\partial f_k(x,t)^{(1)}}{\partial x^*} = -\tau c_x \frac{\partial^2 f_k^{eq}(x,t)}{\partial x^{*2}} \quad (81)$$

Substituting equation “81” into “80”, one obtains:

$$0 = -\tau c_x^2 \frac{\partial^2 f_k^{eq}(x,t)}{\partial x^{*2}} + \frac{\partial f_k^{eq}(x,t)}{\partial t^*} + \frac{1}{2} \Delta t c_x^2 \frac{\partial^2 f_k^{eq}(x,t)}{\partial x^{*2}} \quad (82)$$

Because c_x takes the value either 1 or -1 depending on the direction, the square will cancel out the signs and the result will be equal 1, so equation “82” can be written as:

$$0 = -\tau \frac{\partial^2 f_k^{eq}(x,t)}{\partial x^{*2}} + \frac{\partial f_k^{eq}(x,t)}{\partial t^*} + \frac{1}{2} \Delta t \frac{\partial^2 f_k^{eq}(x,t)}{\partial x^{*2}} \quad (83)$$

Re-arranging and taking the sum for equation “83”, gives:

$$\frac{\partial \sum_2 f_k^{eq}(x,t)}{\partial t^*} + \left(\frac{1}{2} \Delta t - \tau \right) \frac{\partial^2 \sum_2 f_k^{eq}(x,t)}{\partial x^{*2}} = 0 \quad (84)$$

The sum of the equilibrium distribution function gives the macroscopic parameter temperature, hence equation “84” can be written as:

$$\frac{\partial T}{\partial t^*} = \left(\tau - \frac{1}{2}\Delta t\right)\frac{\partial^2 T}{\partial x^{*2}} \quad (85)$$

LBE	NSE
The general form of the Boltzmann Transport Equation: $\frac{\partial f}{\partial t} + c.\nabla f + \frac{F}{m}.\frac{\partial f}{\partial c} = \Omega(f)$	The general form for the compressible Navier-Stoles flow: $\rho\frac{\partial \boldsymbol{u}}{\partial t} + \rho(\boldsymbol{u}.\nabla)\boldsymbol{u} - \mu\nabla^2\boldsymbol{u} = -\rho\nabla P + \frac{1}{3}\mu\nabla(\nabla.\boldsymbol{u}) + \rho\boldsymbol{g}$
Linear equation but the non-linearity can be embedded in the collision operators. **Notice** (depending on the model, some are completely linear such as the diffusion equation without advection)	Non-linear equations. **Notice** (it can be simplified to become linear under certain conditions like neglecting the advection term or setting up the velocities as constant etc.)
Has one discretizing technique called the Lattice Boltzmann method (LBM)	It has three popular discretizing techniques: Finite difference, finite volume and finite element methods.
It has an explicit equation for the microscopic parameter (Distribution function).	It does not have an explicit equation for the macroscopic parameter. **Notice** (an explicit equation for specific parameters can be obtained under certain conditions)
Difficult to capture high Mach numbers	It can capture high Mach numbers
Efficient in parallel processing	Less efficient than LBM in parallel processing.

By comparing equation “85” with “67”, one gets the relation between the diffusion coefficient and the relaxation factor:

$$\Lambda = \left(\tau - \frac{1}{2}\Delta t\right) \quad (86)$$

which leads to:

$$\Lambda = \Delta t\left(\frac{1}{\omega} - \frac{1}{2}\right) \quad (87)$$

A summary table which summarizes and compares the various characteristics of the LBE and NSE is shown below in Table No1.

TABLE NO1: COMPARISON BETWEEN NSE AND LME

III. KYAMOS SOFTWARE

A. High performance computing

1) Hardware

One of the recently developed algorithms by KYAMOS LTD is a finite volume scheme in MPI for time-dependent, convection-diffusion equation solutions. This method ensures that the advective wave is free from non-physical oscillations and artificial diffusion, by adding the correct diffusion through the Finite Volume-Total Variation Diminishing scheme. This highly accurate algorithm is expected to be competitive against our competitors which either utilize similar algorithms or inferior. However, we wish to go a step further and outstrip our competitors by utilizing a newly developed promising technique, an alternative to the NS equations. Hence, the task of developing LB solvers is an innovative task in terms of developing a highly parallelizable new technique compared to the well-established traditional NS solutions, to provide a technology beyond the state-of-the-art software in terms of computational capabilities, ease of use and cost efficiency. KYAMOS software utilizes ready-made software released under the LGPL, MIT, Apache, BSD licenses in order to cut down dramatically any software development time and support costs, hence gaining very high-value, at very low-cost.

For computational efficiency, generic source code applicable for multiple cases that will just make the execution time slower is avoided. By converting our codes to be developed through CUDA aware MPI on cloud-based, distributed GPU computing services, we intend to outperform all existing software, either free or proprietary, and go beyond the state-of-the-art at least in computational efficiency and accuracy. Specifically, we are currently exploiting the full capabilities of the GPU such as L1 cache and shared memory size, number of blocks and number of blocks per thread in the GPU, as well as the nvpp CUDA GPU visual profiler, to squeeze out even the minute speed performance from our Tesla K80 cards.

We do not expect that we can compete with multibillion-dollar companies in terms of modules and variety of solutions, at least in the beginning; however, we believe that we can give a formidable alterative to a few specific areas, until KYAMOS has the resources to expand at the next level, by offering broader simulation solutions for a number of industries such as the manufacturing, automotive, electronics, etc.

A forum on the KYAMOS website exists so that future users can comment, complain and suggest improvements in LB GPU software development. Speaking to investors, they showed great interest in KYAMOS endeavor and some requested for follow up. Regardless of whether investment arrives or not, what is important is that investors are excited and convinced of the potential of KYAMOS succeeding in the business. Finally, we hope to become pioneers in establishing an innovation culture here in Cyprus and convince the authorities to invest in high-technological software companies from Cyprus, likewise with Israel which is considered the start-up nation.

The GPU server holds one of the fastest intra-communication that can be achieved between GPUs (128 PCIe lanes on motherboard) and by using FDR InfiniBand technology (56 Gbps), one can sustain the same speed for inter-communication between nodes, such that it can be expanded to multiple nodes. To

avoid congestion, we intend to use InfiniBand bonding or link aggregation to double the throughput speed interconnection between nodes and ensuring RDMA is successful at very high speeds across nodes by reducing latency and maximizing bandwidth throughput. Specifically, regarding the InfiniBand technology, a Mellanox SX6015 FDR switch (18 ports at 56 Gbps) together with ConnectX-3 VPI HCA-FDR speed cards and original Mellanox passive copper, 1 m length cables with transceivers are utilized.

Another alternative to the PCI-Express 3.0 that we currently use, are the PCI-Express 4.0 and PCI-Express 5.0 that have entered the market in 2020. PCI-Express 4.0 doubles the speed, whereas PCI-Express 5.0 doubles the speed again, reaching 4 times more speed than the existing PCI-Express 3.0 technology. Taking this into consideration, one may exploit this option, as an alternative, when it is available and economically affordable. The NVLink technology is a new alternative as well, however it is brand new and very expensive to acquire. Another alternative is the aws-amazon cloud that charges for a p3.16xlarge instance which includes 8 GPUs with NVIDIA Tesla V100 card, $24.48 per hour. This configuration offers 56 TBytes of double precision arithmetic with 168 GB GPU memory and network bandwidth of 25 Gbps, which is superior in TFlops and inferior in inter-communication speed to the GPU server we are potentially building. However, since KYAMOS with its current resources, can only support two such machines, we may need to have access to multiple machines through aws-services. To conclude, KYAMOS wishes to create new innovative technology that excels in computational efficiency and accuracy with an attempt to be made widely available. This is expected to improve quality of life, stimulate economic growth and social progress both for Cyprus (KYAMOS revenues, incomes to government though taxes and to our employees) and for our potential customers abroad that will design better engineering systems that will make people's lives potentially better and safer.

We have already created a Tier-4 based server room of around 15 m^2 area with UPS online support, isolator switches for protection, smoke detectors, intruder alarm detectors, HD cameras for surveillance, 24/7 air-conditioned environment with remote access Wi-Fi control, automatic fire extinguishing system, thermal and sound proofing doors, thermometer/hygrometer to control the server room.

B. Industrial application potential

The utilization of LB solvers in the CAE industry merely exists. If it is combined with the cloud-based, distributed GPU computing, it will provide immense potential that will establish KYAMOS as a pioneering company in the field by providing capability for near real-live simulations of computationally intensive problems that is not possible, until now. Hence, we anticipate a huge industrial application potential emanating through the LB solvers.

C. KYAMOS significance and impact

1) Significance and impact

Economic: KYAMOS software will cost much lower than its competitors since it encompasses freely available software, without compromising on quality or design ability. It is a Cyprus based company which is benefiting from the current favor regime of reduced taxes and corporate services. We are currently looking for investors such that to scale the company at an international level and to be established as a company in the CAE industry.

KYAMOS high performance computing activities encompass cloud-based distributed GPU, software development, cheap cloud-based solutions, algorithms parallelization, multiphysics engineering knowhow, GUI development, CUDA aware MPI setup of servers, cross-platform manipulation, utilization of geometry CAD, mesh and plotting editors, mathematical algorithm optimization and many other skills.

IV. RESULTS

A. Introduction

A well-chosen benchmark test case has been solved which involves all-natural phenomena such as turbulence, convection and diffusion solution to test the accuracy, as well as the speed of the LB software and find its strengths and weaknesses.

B. Lid-driven cavity flow

Lid-driven cavity flow is the most well-known test case for the testing of incompressible flows. In this case, we test the following configuration. A geometry of 1 m by 1 m is considered in two-dimensions with an initial density ρ = 1 kg/m^3. The Reynolds number Re = 200 and the lid-driven velocity is set to U = 0.1 ms^{-1}. In order to simulate the lid-driven cavity flow in Lattice Boltzmann, we utilize a D2Q9 model. The dynamic viscosity is calculated as:

$$\mu = \frac{UL_x}{Re} \tag{88}$$

The relaxation time is calculated as:

$$\tau = \frac{3\mu}{c^2 dt} + 0.5 \tag{89}$$

where c is the lattice speed of sound (c =1 ms^{-1}) and dt is calculated as:

$$dt = \frac{dx}{c} \tag{90}$$

We set the residual value for convergence to $1x10^{-5}$, which is calculated by finding the relative error between consecutive time steps in the velocity field:

$$Error = \sum \frac{(u(t+1)-u(t))^2}{u(t)^2} \tag{91}$$

It is necessary that more than 20,000 steps are necessary for the error to converge. The GPU parallel based simulation time of LB solver is found to be 25 times faster than an identical serial solver developed in a competitor's software. Our solver is expected to

be even faster when profiled in the future. On top of that, with the beginning of conducting sales, we will exploit much faster GPU cards that can be as 3 times faster (Tesla V100) and much faster InfiniBand networks at 100 (EDR) and 200 (HDR) Gb/s. Fig. 2 shows the velocity field and Fig. 3 the pressure field that is developed in two-dimensions and it is shown that the simulation is able to capture the vortices created.

We are utilizing dynamic parallelism in the GPU cards which allows a number of threads to fire a number of other threads. On top of that, we utilize streams which allows the execution of independent functions simultaneously. Finally, we utilize Hyper-Q technology which allows the GPU to be accessed not only from a single thread or core, but from multiple ones simultaneously, provided there are available threads on the GPU card. Additionally, we enable peer to peer process which allows direct copy of data from one GPU to the other GPU, bypassing the host memory, as well as utilize the message passing interface to allow distributed computing and scaling to multiple nodes. The partitioning of the mesh through the various nodes is done in such a way to minimize the necessary communication between the nodes and on top of that, we utilize hallo elements to minimize intra- and inter-communication even further.

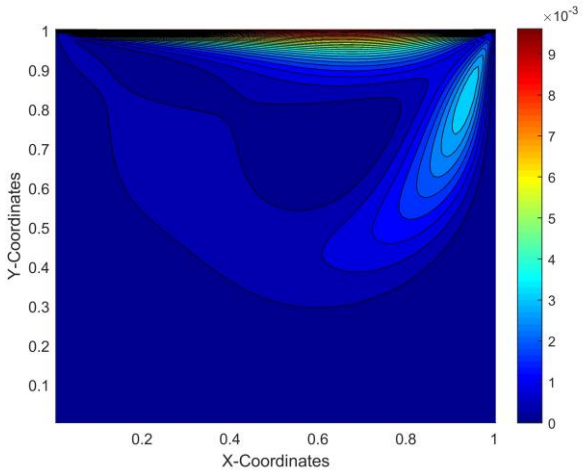

Fig. 2. Velocity field on a 255 x 255 Lattice Boltzmann grid at steady state

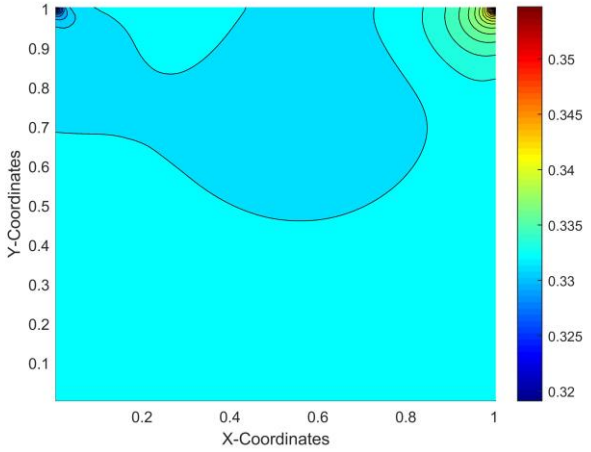

Fig. 3. Pressure field on a 255 x 255 Lattice Boltzmann grid at steady state

One should bear in mind that for small scale simulations with fairly small grid, the simulations will run faster when run on a single GPU, since KYAMOS software is highly GPU parallel implemented based and if one card has enough GPU RAM and processing threads to run a simulation, it will run faster than multiple GPUs since one has the additional burden of overhead communication. Peer to peer and CUDA aware MPI becomes effective only when one is limited by a single GPU's threads or RAM and additional resources are necessary.

Fig. 4 below shows a flow chart depicting the main steps in developing a Lattice Boltzmann solver for the lid-driven cavity flow.

V. CONCLUSIONS

KYAMOS software aims to realize the formulation, development, validation, and optimization of LB solvers that can be applied to engineering problems by utilizing high performance computing through cloud-based distributed GPUs and state-of-the-art mathematical algorithms. It is believed that its future potential release as state-of-the-art proprietary multiphysics engineering software which realizes cloud-based distributed GPU and near real-live simulations, will help contribute in addressing key economic and social challenges for achieving sustainable development in Cyprus.

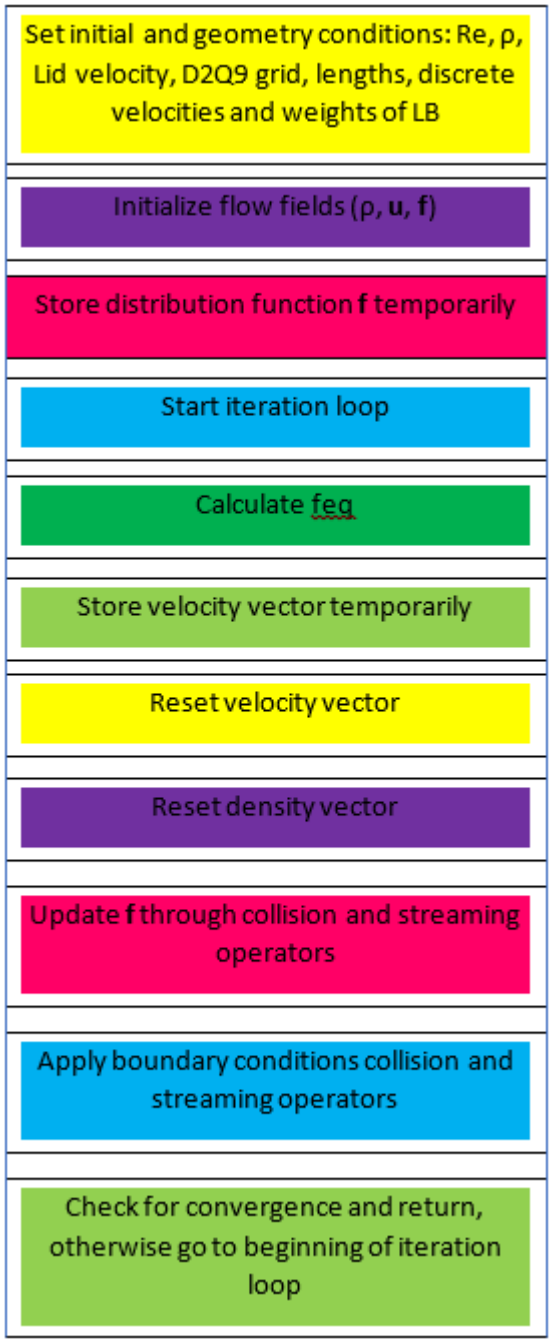

Fig. 4. Flow chart for LB solver

ACKNOWLEDGMENT

This work was co-funded by the European Regional Development Fund and the Republic of Cyprus through the Research and Innovation Foundation (Project: PROOF OF CONCEPT/0618/0012).

REFERENCES

[1] D. A. Perumal and A. K. Dass, "A Review on the development of lattice Boltzmann computation of macro fluid flows and heat transfer," *Alexandria Engineering Journal,* vol. 54, no. 4, pp. 955-971, 2015.

[2] B. Manhartsgruber, "The Lattice Boltzmann Method used for fluid flow modeling in hydraulic components," in *Proceedings of 15: th Scandinavian International Conference on Fluid Power, June 7-9, 2017, Linköping, Sweden*, 2017, no. 144, pp. 295-300: Linköping University Electronic Press.

[3] S. Ubertini and S. Succi, "A generalised lattice Boltzmann equation on unstructured grids," *Communications in Computational Physics,* vol. 3, no. 2, pp. 342-356, 2008.

[4] S. Succi, N. Moradi, A. Greiner, and S. Melchionna, "Lattice Boltzmann modeling of water-like fluids," *Frontiers in Physics,* vol. 2, p. 22, 2014.

[5] Q. Li, K. H. Luo, Q. Kang, Y. He, Q. Chen, and Q. Liu, "Lattice Boltzmann methods for multiphase flow and phase-change heat transfer," *Progress in Energy and Combustion Science,* vol. 52, pp. 62-105, 2016.

[6] L. Jahanshaloo, E. Pouryazdanpanah, and N. A. Che Sidik, "A review on the application of the lattice Boltzmann method for turbulent flow simulation," *Numerical Heat Transfer, Part A: Applications,* vol. 64, no. 11, pp. 938-953, 2013.

[7] Z. Chai, B. Shi, and Z. Guo, "A multiple-relaxation-time lattice Boltzmann model for general nonlinear anisotropic convection–diffusion equations," *Journal of Scientific Computing,* vol. 69, no. 1, pp. 355-390, 2016.

[8] N. A. C. Sidik and S. A. Razali, "Lattice Boltzmann method for convective heat transfer of nanofluids–A review," *Renewable and Sustainable Energy Reviews,* vol. 38, pp. 864-875, 2014.

[9] Y.-L. He, Q. Liu, Q. Li, and W.-Q. Tao, "Lattice Boltzmann methods for single-phase and solid-liquid phase-change heat transfer in porous media: A review," *International Journal of Heat and Mass Transfer,* vol. 129, pp. 160-197, 2019.

[10] J. Buick, J. Cosgrove, S. Tonge, M. Collins, A. Mulholland, and B. Steves, "The lattice Boltzmann equation for modelling arterial flows: review and application," *Biomedicine and Pharmacotherapy,* vol. 56, no. 7, pp. 345-346, 2003.

[11] H. Liu *et al.*, "Multiphase lattice Boltzmann simulations for porous media applications," *Computational Geosciences,* vol. 20, no. 4, pp. 777-805, 2016.

[12] D. B. Dhuri, S. M. Hanasoge, P. Perlekar, and J. O. Robertsson, "Numerical analysis of the lattice Boltzmann method for simulation of linear acoustic waves," *Physical Review E,* vol. 95, no. 4, p. 043306, 2017.

[13] S. H. Musavi and M. Ashrafizaadeh, "A mesh-free lattice Boltzmann solver for flows in complex geometries," *International journal of heat and fluid flow,* vol. 59, pp. 10-19, 2016.