

Please cite the Published Version

Zhou, Jian Guo  (2021) Macroscopic Lattice Boltzmann Method. Water, 13 (1). 61

DOI: <https://doi.org/10.3390/w13010061>

Publisher: MDPI

Version: Published Version

Downloaded from: <https://e-space.mmu.ac.uk/622432/>

Usage rights:  [Creative Commons: Attribution 4.0](https://creativecommons.org/licenses/by/4.0/)


Additional Information: Open Access article published by MDPI and copyright The Author.

Enquiries:

If you have questions about this document, contact openresearch@mmu.ac.uk. Please include the URL of the record in e-space. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from <https://www.mmu.ac.uk/library/using-the-library/policies-and-guidelines>)

Article

Macroscopic Lattice Boltzmann Method

Jian Guo Zhou 

Department of Computing and Mathematics, Manchester Metropolitan University, Manchester M1 5GD, UK; J.Zhou@mmu.ac.uk

Abstract: The lattice Boltzmann method (LBM) is a highly simplified model for fluid flows using a few limited fictitious particles. It has been developed into a very efficient and flexible alternative numerical method in computational physics, demonstrating its great power and potential for resolving more and more challenging physical problems in science and engineering covering a wide range of disciplines such as physics, chemistry, biology, material science and image analysis. The LBM is implemented through the two routine steps of streaming and collision using the three parameters of the lattice size, particle speed and collision operator. A fundamental question is if the two steps are integral to the method or if the three parameters can be reduced to one for a minimal lattice Boltzmann method. In this paper, it is shown that the collision step can be removed and the standard LBM can be reformulated into a simple macroscopic lattice Boltzmann method (MacLAB). This model relies on macroscopic physical variables only and is completely defined by one basic parameter of the lattice size δx , bringing the LBM into a precise “lattice” Boltzmann method. The viscous effect on flows is naturally embedded through the particle speed, making it an ideal automatic simulator for fluid flows. Three additional advantages compared to the existing LBMs are that: (i) physical variables can directly be retained as the boundary conditions; (ii) much less computational memory is required; and (iii) the model is unconditionally stable. The findings are demonstrated and confirmed with numerical tests including flows that are independent of and dependent on fluid viscosity, 2D and 3D cavity flows and an unsteady Taylor–Green vortex flow. This provides an efficient and powerful model for resolving physical problems in various disciplines of science and engineering.



Citation: Zhou, J.G. Macroscopic Lattice Boltzmann Method. *Water* **2021**, *13*, 61.
<https://doi.org/10.3390/w13010061>

Received: 15 November 2020
Accepted: 25 December 2020
Published: 30 December 2020

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: macroscopic lattice Boltzmann method; fluid flows; computational fluid dynamics; numerical method

PACS: 47.11.-j; 02.60.Cb; 02.70.-c

1. Introduction

The birth of the lattice Boltzmann method (LBM) fulfils a dream that simple arithmetic calculations can simulate complex fluid flows without solving complicated partial differential flow equations. The method is a highly simplified discrete model for fluid flows using a few limited fictitious particles that move one grid at a constant time interval and collide with each other at a grid point on uniform lattices, which are the two routine steps for the implementation of the method to simulate fluid flows. As such, a real complex particle dynamics is approximated as a regular particle model using three parameters of the lattice size, particle speed and collision operator. The LBM is characterised by its simplicity, parallel processing and easy treatment of boundary conditions [1]. The first fully discrete model for fluid flows on a square lattice was proposed by Hardy et al. [2] in 1976. Ten years later, Frisch et al. [3] for the first time obtained a correct lattice gas automata (LGA) for Navier–Stokes equations using a six-velocity hexagonal lattice. The LGA is comprised of two steps: streaming and collision. The two steps are represented by the lattice size, particle speed and a collision operator on a uniform lattice. In physics, the former and the latter simulate the phenomena of fluid movement and diffusion, respectively, which determine the basic feature of an LGA. Often, simulations generated using an LGA are

very noisy due to its Boolean variable with one for the presence and zero for the absence of particles at a lattice point [4,5]. Furthermore, the numerical procedure involves the calculations of particle probability, which reduces the efficiency of the model. To overcome these, the lattice Boltzmann method was proposed [6], and its basic difference from the LGA is that the Boolean variable is replaced with a particle distribution function. Such an approach eliminates the statistical noise in an LGA and retains all the advantages of locality in the kinetic form of an LGA [1]. McNamara and Zanetti [6] first used the lattice Boltzmann method as an alternative to the LGA in 1988. As the collision operator takes a complex matrix form, this prevents the LBM from becoming a competing numerical method. A breakthrough in progress was made by Higuera and Jiménez [7] through linearisation of the collision term around its local equilibrium state. This greatly simplifies the collision operator. Noble et al. [8] used this idea to express the collision operator as $\Omega_{\alpha\beta}(f_{\beta}^{eq} - f_{\beta})$, in which f_{β} is the particle distribution function; f_{β}^{eq} is the local equilibrium distribution function; and $\Omega_{\alpha\beta}$ is a collision matrix. Later, several researchers [9,10] suggested a simple linearized form for the collision matrix by using a single time relaxation towards the local equilibrium distribution, $\Omega_{\alpha\beta} = -\delta_{\alpha\beta}/\tau$, which is popularly known as the Bhatnagar–Gross–Krook (BGK) collision operator [11], where $\delta_{\alpha\beta}$ is the Kronecker delta function taking one when $\alpha = \beta$, or otherwise zero, and τ is called the single relaxation time that takes a value. The correct choice of τ together with the lattice size and time step can achieve a desired fluid viscosity. This leads to the single relaxation time lattice Boltzmann method (SRT LBM), which has been the most efficient and used so far,

$$f_{\alpha}(x_j + e_{\alpha j}\delta t, t + \delta t) = f_{\alpha}(x_j, t) + \frac{1}{\tau}[f_{\alpha}^{eq}(x_j, t) - f_{\alpha}(x_j, t)], \quad (1)$$

where x_j is a lattice coordinate along the j -axis in the Cartesian coordinate system, i.e., $j = x, y$ in the two-dimensional space; t is time; $e_{\alpha j}$ is the j^{th} component of the particle velocity vector \mathbf{e}_{α} in α -link of the lattice and defined by time step δt and lattice size δx , e.g., $\mathbf{e}_{\alpha} = (0, 0), (e, 0), (0, e), (-e, 0), (0, -e), (e, e), (-e, e), (-e, -e), (e, -e)$ when $\alpha = 0 - 8$ for nine particles moving in the two-dimensional uniform square lattice (D2Q9), in which e is the particle speed and defined as $e = \delta x/\delta t$; and f_{α}^{eq} is the local equilibrium distribution function given by:

$$f_{\alpha}^{eq} = w_{\alpha}\rho\left(1 + 3\frac{e_{\alpha i}u_i}{e^2} + \frac{9}{2}\frac{e_{\alpha i}e_{\alpha j}u_i u_j}{e^4} - \frac{3}{2}\frac{u_i u_i}{e^2}\right), \quad (2)$$

in which ρ is the fluid density and w_{α} is a weighting factor depending on the lattice pattern, e.g., $w_{\alpha} = 4/9$ when $\alpha = 0$, $w_{\alpha} = 1/9$ when $\alpha = 1 - 4$ and $w_{\alpha} = 1/36$ when $\alpha = 5 - 8$ on D2Q9. After the distribution function is calculated from the lattice Boltzmann Equation (1), the macroscopic physical variables, density and velocity are simply updated as:

$$\rho(x_j, t) = \sum_{\alpha} f_{\alpha}(x_j, t), \quad u_i(x_j, t) = \frac{1}{\rho(x_j, t)} \sum_{\alpha} e_{\alpha i} f_{\alpha}(x_j, t). \quad (3)$$

Since then, the study of the lattice Boltzmann method and the applications of the method have received extensive attentions, making it become a very powerful modelling tool in many areas such as thermodynamics [12], aerodynamics [13], multiphase flows [14], turbulent flows [15], hemodynamics [16], biomechanics [17], image analysis [18], biology [19] and environmental science [20].

In applications, it is found that the SRT LBM suffers from a numerical instability. To remedy this, the multiple relaxation time (MRT) collision operator was introduced in 1992 [21,22]. This improves the stability, but it reduces the efficiency. To accelerate the simulation, a two relaxation time collision operator was developed in 2008 [23], which has almost the same efficiency as the SRT LBM. As research progresses, it is noticed that MRT or TRT still suffers from numerical instabilities when fluid flows with very small viscosity are simulated. After realising that this is caused by an insufficient degree of Galilean

invariance in the collision step, Geier et al. [24] proposed a cascaded lattice Boltzmann method by relaxing the particle distribution function to its local equilibrium state in the central moment space, making the LBM stable for simulating flows with small viscosity close to zero. In 2015, Geier et al. [25] further improved their central moment LBM using the cumulant in collision operator, which is called the cumulant lattice Boltzmann method (CLBM). Despite such enhancements, these schemes are more complicated and less efficient due to a manipulation of the matrices. In addition, these schemes share the same drawback as the existing LBM in that the boundary conditions for a physical variable such as velocity cannot be implemented without being converted to particle distribution functions, which further reduces the efficiency and accuracy of the methods. Recently, Chen et al. [26] developed a simplified lattice Boltzmann method without the evolution of the particle distribution function, which successfully removes this drawback and enables the direct use of a physical variable as the boundary conditions. However, the method involves the two steps of the predictor and corrector, which is more complicated than the SRT LBM. Nevertheless, through all this research, the LBM has greatly been improved and developed to a point where it has become a very efficient and flexible alternative numerical method in computational physics. Its potential power is much beyond the original scope, being explored and demonstrated in various disciplines of science and engineering with time [27–30].

2. Macroscopic Lattice Boltzmann Model

The above literature review highlights that the major research to improve the method has been carried out on the collision operator for resolving the well-known instability problem in the SRT LBM since 1992, leading to four representative variants of the method, MRT, TRT, central moment and cumulant LBMs. Even so, choosing suitable parameters or values for the collision operators is not clear, and they cannot be tuned without using trial and error during simulations, which becomes more complicated in a non-single relaxation time scheme due to its complexity. This unnecessarily wastes time and computing resource, and it may become awkward to simulate a large-scale flow system, preventing the LBM from becoming an automatic simulator for any scale flows when a super-fast computer such as a quantum computer becomes available one day. As the central problem comes from the collision operator, such a problem may be resolved forever if the collision operator can be removed. Since the function of the collision is to relax the distribution function towards its local equilibrium state, it may be removed and replaced with the local equilibrium distribution function by setting $\tau = 1$ in Equation (1). Following this idea, after mathematical manipulations, we obtain the following simple macroscopic lattice Boltzmann model (MacLAB),

$$\rho(x_j, t) = \sum f_{\alpha}^{eq}(x_j - e_{\alpha j} \delta t, t - \delta t), \quad (4)$$

$$u_i(x_j, t) = \frac{1}{\rho(x_j, t)} \sum e_{\alpha j} f_{\alpha}^{eq}(x_j - e_{\alpha j} \delta t, t - \delta t), \quad (5)$$

to determine the physical density and velocity directly from the local equilibrium distribution function without calculating the distribution function using Equation (1) unlike existing LBMs (the theoretical derivation is given in Section 3). Apparently, the model involves the local equilibrium distribution function only. However, doing so brings a new problem of how to consider the fluid viscosity in the absence of the collision step. We overcome this from the finding, through the recovery of the Navier–Stokes equations from Equations (4) and (5), that if the particle speed e is determined using:

$$e = 6v/\delta x, \quad (6)$$

which is employed in Equation (2) instead of using $e = \delta x/\delta t$ to calculate the local equilibrium distribution function f_{α}^{eq} , the flow viscosity is naturally taken into account in the model. In the MacLAB, once a lattice size δx is chosen, the model is ready to simulate a

flow with a viscosity ν because $(x_j - e_{\alpha j}\delta t)$ stands for a neighbouring lattice point; f_{α}^{eq} at time of $(t - \delta t)$ represents its known quantity at the current time; and the particle speed e is determined from Equation (6) for use in the computation of f_{α}^{eq} . In addition, the time step δt is no longer an independent parameter and needs to be determined as $\delta t = \delta x/e$, which is used in the simulations of unsteady flows. Consequently, only the lattice size δx is required in the MacLAB for the simulation of fluid flows, bringing the LBM into a precise “lattice” Boltzmann method. This enables the model to become an automatic simulator for any scale flows without tuning other simulation parameters, making it possible and easy to model a large flow system when a super-fast computer such as a quantum computer becomes available in the future. The model is unconditionally stable as it shares the same valid condition as that for f_{α}^{eq} , or the Mach number $M = U_c/e$ is much smaller than one, in which U_c is a characteristic flow speed. The Mach number can also be expressed as a lattice Reynolds number of $R_{le} = U_c\delta x/\nu$ via Equation (6). In practical simulations, it is found that the model is stable if $R_{le} = U_m\delta x/\nu < 1$ where U_m is the maximum flow speed and is used as the characteristic flow speed. The main features are that there is no collision operator and only macroscopic physical variables such as the density and velocity are required, which are directly retained as boundary conditions with a minimum memory requirement. The implementation of the model starting from the initial density and velocity is to (i) choose the lattice size δx and determine the particle speed e from Equation (6), (ii) calculate f_{α}^{eq} from Equation (2) using the density and velocity, (iii) update the density and velocity using Equations (4) and (5), (iv) apply the boundary conditions, and (v) repeat Step (ii) until a solution is reached. The only limitation of the described model is that, for very small viscosity or high speed flow, the chosen lattice size after satisfying $R_{le} < 1$ may turn out to generate very large lattice points (lattice points, e.g., for one dimension with a length of L are calculated as $N_L = L/\delta x$, and N_L is the lattice points); if the total lattice points are too big such that the demanding computations are beyond the current power of a computer, the simulation cannot be carried out. Such difficulties may be solved or relaxed through parallel computing using computer techniques such as GPU processors and multiple servers and will be largely or completely removed using quantum computing when a quantum computer becomes available.

3. Recovery of the Navier–Stokes Equations

Setting $\tau = 1$ in Equation (1) leads to:

$$f_{\alpha}(x_j + e_{\alpha j}\delta t, t + \delta t) = f_{\alpha}^{eq}(x_j, t), \quad (7)$$

which can be rewritten as:

$$f_{\alpha}(x_j, t) = f_{\alpha}^{eq}(x_j - e_{\alpha j}\delta t, t - \delta t). \quad (8)$$

Taking \sum Equation (8) and $\sum e_{\alpha i}$ Equation (8) yields:

$$\sum f_{\alpha}(x_j, t) = \sum f_{\alpha}^{eq}(x_j - e_{\alpha j}\delta t, t - \delta t), \quad (9)$$

and:

$$\sum e_{\alpha i}f_{\alpha}(x_j, t) = \sum e_{\alpha i}f_{\alpha}^{eq}(x_j - e_{\alpha j}\delta t, t - \delta t), \quad (10)$$

respectively. In the lattice Boltzmann method, the density and velocity are determined using the distribution function as:

$$\rho(x_j, t) = \sum_{\alpha} f_{\alpha}(x_j, t), \quad u_i(x_j, t) = \frac{1}{\rho} \sum_{\alpha} e_{\alpha i}f_{\alpha}(x_j, t). \quad (11)$$

Combining Equation (11) with Equations (9) and (10) results in the current MacLAB, Equations (4) and (5). Since the local equilibrium distribution function f_α^{eq} has the features of:

$$\sum_\alpha f_\alpha^{eq}(x_j, t) = \rho(x_j, t), \quad \frac{1}{\rho} \sum_\alpha e_{\alpha i} f_\alpha^{eq}(x_j, t) = u_i(x_j, t), \tag{12}$$

with reference to Equation (11), the following relationships,

$$\sum_\alpha f_\alpha(x_j, t) = \sum_\alpha f_\alpha^{eq}(x_j, t), \quad \sum_\alpha e_{\alpha i} f_\alpha(x_j, t) = \sum_\alpha e_{\alpha i} f_\alpha^{eq}(x_j, t), \tag{13}$$

hold, which are the conditions that retain the conservation of the mass and momentum in the lattice Boltzmann method.

Next, we prove that the continuity and Navier–Stokes equations can be recovered from Equations (4) and (5). Rewriting Equation (1) as:

$$\begin{aligned} f_\alpha(x_j, t) &= f_\alpha(x_j - e_{\alpha j} \delta t, t - \delta t) \\ &+ \frac{1}{\tau} [f_\alpha^{eq}(x_j - e_{\alpha j} \delta t, t - \delta t) \\ &- f_\alpha(x_j - e_{\alpha j} \delta t, t - \delta t)]. \end{aligned} \tag{14}$$

Apparently, when $\tau = 1$, the above equation becomes Equation (8), which leads to Equations (4) and (5); hence, Equation (14) is a general equation and is used in the following derivation. Applying a Taylor expansion to the two terms on the right-hand side of Equation (14) in time and space at point (\mathbf{x}, t) yields:

$$\begin{aligned} f_\alpha(x_j - e_{\alpha j} \delta t, t - \delta t) &= f_\alpha(x_j, t) - \delta t \left(\frac{\partial}{\partial t} + e_{\alpha j} \frac{\partial}{\partial x_j} \right) f_\alpha \\ &+ \frac{1}{2} \delta t^2 \left(\frac{\partial}{\partial t} + e_{\alpha j} \frac{\partial}{\partial x_j} \right)^2 f_\alpha + \mathcal{O}(\delta t^3), \end{aligned} \tag{15}$$

and:

$$\begin{aligned} f_\alpha^{eq}(x_j - e_{\alpha j} \delta t, t - \delta t) &= f_\alpha^{eq}(x_j, t) - \delta t \left(\frac{\partial}{\partial t} + e_{\alpha j} \frac{\partial}{\partial x_j} \right) f_\alpha^{eq} \\ &+ \frac{1}{2} \delta t^2 \left(\frac{\partial}{\partial t} + e_{\alpha j} \frac{\partial}{\partial x_j} \right)^2 f_\alpha^{eq} + \mathcal{O}(\delta t^3). \end{aligned} \tag{16}$$

According to the Chapman–Enskog analysis, f_α can be expanded around $f_\alpha^{(0)}$:

$$f_\alpha = f_\alpha^{(0)} + f_\alpha^{(1)} \delta t + f_\alpha^{(2)} \delta t^2 + \mathcal{O}(\delta t^3). \tag{17}$$

After substituting Equations (15)–(17) into Equation (14), equating the coefficients of δt with various orders results for the order $(\delta t)^0$ in:

$$f_\alpha^{(0)} = f_\alpha^{eq}, \tag{18}$$

for the order $(\delta t)^1$ in:

$$\left(\frac{\partial}{\partial t} + e_{\alpha j} \frac{\partial}{\partial x_j} \right) f_\alpha^{(0)}, = -\frac{1}{\tau} f_\alpha^{(1)}, \tag{19}$$

and for the order $(\delta t)^2$ in:

$$\begin{aligned} & \left(\frac{\partial}{\partial t} + e_{\alpha j} \frac{\partial}{\partial x_j} \right) f_{\alpha}^{(1)} - \frac{1}{2} \left(\frac{\partial}{\partial t} + e_{\alpha j} \frac{\partial}{\partial x_j} \right)^2 f_{\alpha}^{(0)} \\ & = -\frac{1}{\tau} f_{\alpha}^{(2)} + \frac{1}{\tau} \left(\frac{\partial}{\partial t} + e_{\alpha j} \frac{\partial}{\partial x_j} \right) f_{\alpha}^{(1)}. \end{aligned} \quad (20)$$

Substitution of Equation (19) into the above equation gives:

$$\begin{aligned} & \left(\frac{\partial}{\partial t} + e_{\alpha j} \frac{\partial}{\partial x_j} \right) f_{\alpha}^{(1)} - \frac{1}{2} \left(\frac{\partial}{\partial t} + e_{\alpha j} \frac{\partial}{\partial x_j} \right) \left(-\frac{1}{\tau} f_{\alpha}^{(1)} \right) \\ & = -\frac{1}{\tau} f_{\alpha}^{(2)} + \frac{1}{\tau} \left(\frac{\partial}{\partial t} + e_{\alpha j} \frac{\partial}{\partial x_j} \right) f_{\alpha}^{(1)}, \end{aligned} \quad (21)$$

which is rearranged as:

$$\left(1 - \frac{1}{2\tau} \right) \left(\frac{\partial}{\partial t} + e_{\alpha j} \frac{\partial}{\partial x_j} \right) f_{\alpha}^{(1)} = -\frac{1}{\tau} f_{\alpha}^{(2)}. \quad (22)$$

From Equation (19) + Equation (22) $\times \delta t$, we have:

$$\begin{aligned} & \left(\frac{\partial}{\partial t} + e_{\alpha j} \frac{\partial}{\partial x_j} \right) f_{\alpha}^{(0)} + \delta t \left(1 - \frac{1}{2\tau} \right) \left(\frac{\partial}{\partial t} + e_{\alpha j} \frac{\partial}{\partial x_j} \right) f_{\alpha}^{(1)} \\ & = -\frac{1}{\tau} (f_{\alpha}^{(1)} + \delta t f_{\alpha}^{(2)}). \end{aligned} \quad (23)$$

Now, taking Σ Equation (23) leads to:

$$\Sigma \left(\frac{\partial}{\partial t} + e_{\alpha j} \frac{\partial}{\partial x_j} \right) f_{\alpha}^{(0)} = 0 \quad (24)$$

as:

$$\Sigma_{\alpha} f_{\alpha}^{(1)} = \Sigma_{\alpha} f_{\alpha}^{(2)} = \Sigma_{\alpha} e_{\alpha i} f_{\alpha}^{(1)} = \Sigma_{\alpha} e_{\alpha i} f_{\alpha}^{(2)} = 0 \quad (25)$$

due to the condition of the conservation of mass and momentum Equation (13). Evaluating the terms in the above equation using Equation (2) produces the exact continuity equation,

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_j)}{\partial x_j} = 0. \quad (26)$$

Multiplying Equation (23) by $e_{\alpha i}$ provides:

$$\begin{aligned} & \left(\frac{\partial}{\partial t} + e_{\alpha j} \frac{\partial}{\partial x_j} \right) e_{\alpha i} f_{\alpha}^{(0)} + \delta t \left(1 - \frac{1}{2\tau} \right) \left(\frac{\partial}{\partial t} + e_{\alpha j} \frac{\partial}{\partial x_j} \right) e_{\alpha i} f_{\alpha}^{(1)} \\ & = -\frac{1}{\tau} (e_{\alpha i} f_{\alpha}^{(1)} + \delta t e_{\alpha i} f_{\alpha}^{(2)}). \end{aligned} \quad (27)$$

Taking Σ Equation (27) leads to:

$$\begin{aligned} & \sum \left(\frac{\partial}{\partial t} + e_{\alpha j} \frac{\partial}{\partial x_j} \right) e_{\alpha i} f_{\alpha}^{(0)} \\ & + \delta t \left(1 - \frac{1}{2\tau} \right) \sum \left(\frac{\partial}{\partial t} + e_{\alpha j} \frac{\partial}{\partial x_j} \right) e_{\alpha i} f_{\alpha}^{(1)} = 0 \end{aligned} \quad (28)$$

under the same condition (25) as that in the derivation of Equation (24). Evaluating the terms in the above equation using Equation (2) produces the exact momentum equation, the Navier–Stokes equation at second-order accuracy on the condition that the Mach number $M = U_c/e \ll 1$,

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial(\rho u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2(\rho u_i)}{\partial x_j^2}, \quad (29)$$

where pressure p is defined as:

$$p = \frac{1}{3}\rho e^2 \quad (30)$$

and the kinematic viscosity is:

$$\nu = \frac{1}{6}(2\tau - 1)e\delta x. \quad (31)$$

As τ takes a constant, the use of $\tau = 1$ will recover the continuity and the Navier–Stokes equations at the second-order accuracy, as the above derivation shows. In this case, Equation (31) becomes Equation (6), which determines the particle speed e in the proposed model.

4. Numerical Tests

In order to demonstrate the validation of the described model, five numerical simulations were carried out using D2Q9 and D3Q19 lattices for 2D and 3D flows, respectively. For D3Q19, the particle velocity vector is $\mathbf{e}_{\alpha} = (e_{\alpha x}, e_{\alpha y}, e_{\alpha z}) = (0, 0, 0), (e, 0, 0), (-e, 0, 0), (0, e, 0), (0, -e, 0), (0, 0, e), (0, 0, -e), (e, e, 0), (-e, -e, 0), (-e, e, 0), (e, -e, 0), (0, e, e), (0, -e, -e), (0, -e, e), (0, e, -e), (e, 0, e), (-e, 0, -e), (-e, 0, e), (e, 0, -e)$, and the weighting factor w_{α} for Equation (2) is $w_{\alpha} = 1/3$ when $\alpha = 0$, $w_{\alpha} = 1/18$ when $\alpha = 1 - 6$ and $w_{\alpha} = 1/36$ when $\alpha = 7 - 18$. Physical variables in SI units without explicit notation are used in the numerical simulations, e.g., the density $\rho = 1$ implies $\rho = 1 \text{ kg/m}^3$.

4.1. Couette Flow

The first test is a Couette flow through two parallel plates without a pressure gradient. The distance between the plates is $h = 1$. The top plate moves at a velocity of $u_x = u_0 = 0.1$ in the streamwise direction, and the bottom plate is fixed. If x stands for the streamwise direction and y for the vertical direction, the analytical solution is:

$$u_x(y) = \frac{u_0}{h}y, \quad (32)$$

which is the same test as that used by Chen et al. [26]. This is a very interesting case as the steady flow is independent of the flow viscosity according to the theory (32). We use $\delta x = 0.02$ and 20×50 lattices in the x and y directions for three simulations of flows with three kinematic viscosities of $\nu_1 = 0.01$, $\nu_2 = 0.001$ and $\nu_3 = 0.0006$, respectively. The periodic boundary conditions are applied at the inflow and outflow boundaries. After the steady solutions are reached, the results are indeed independent of the viscosities, and one of those is shown in Figure 1, demonstrating excellent agreement with the analytical solution.

4.2. Couette Flow with a Pressure Gradient

The second test is the same flow as that in Test 1 except that a pressure gradient of $\partial p/\partial x = -0.0001$ is specified, which is added to the right-hand side of Equation (5) as $+\delta x/(e\rho)\partial p/\partial x$ [31]. Both plates are fixed with zero velocities at the top and bottom boundaries at which no calculations are needed. The flow is affected by viscosity, and the analytical solution is:

$$u(y) = \frac{u_0}{h}y + \frac{1}{2\rho\nu} \frac{\partial p}{\partial x} (y^2 - hy). \quad (33)$$

The flow is simulated using three viscosities of $\nu_1 = 0.003$, $\nu_2 = 0.001$ and $\nu_3 = 0.0006$. The numerical results are plotted in Figure 2, showing the effect of viscosity on the flow, in excellent agreement with the analytical solutions. This confirms the unique feature that the current model can simulate viscous flow correctly due to the use of Equation (6), although no explicit effect of viscosity on flows is taken into account.

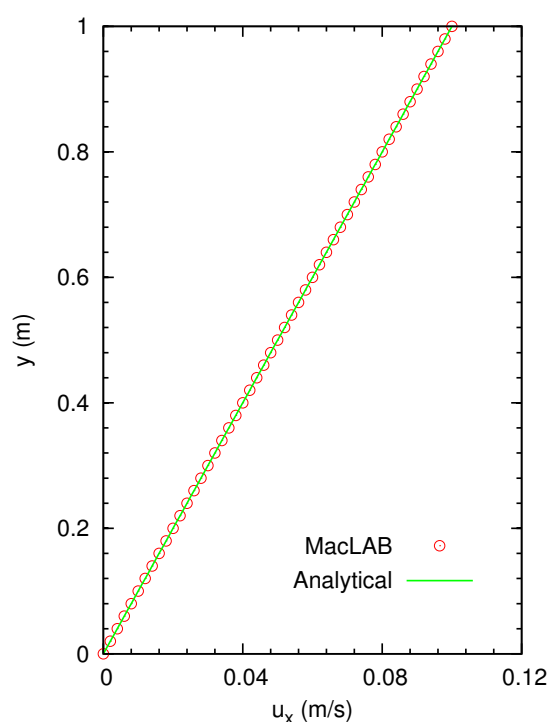


Figure 1. Couette flow through two parallel plates without a pressure gradient. The distance between the plates is $h = 1$. The top plate moves at a velocity of 0.1 in the streamwise direction, and the bottom plate is fixed, where no calculations are required at the top or bottom boundaries. The periodic boundary conditions are applied at the inflow and outflow boundaries. $\delta x = 0.02$ is used for three simulations of flows with three kinematic viscosities of $\nu_1 = 0.01$, $\nu_2 = 0.001$ and $\nu_3 = 0.0006$, respectively. All the steady numerical results are almost identical and are independent of flow viscosity, as shown here in the comparison of one numerical result with the analytical solution. MacLAB, macroscopic lattice Boltzmann method.

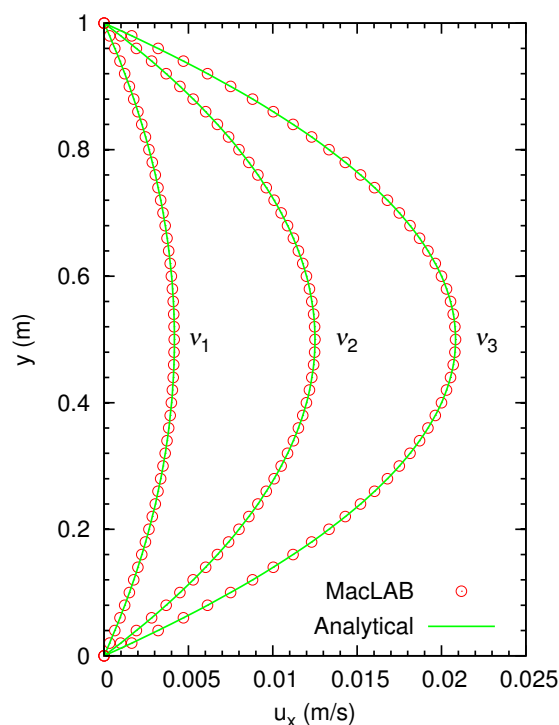


Figure 2. Couette flow through two parallel plates under a pressure gradient of $\partial p/\partial x = -0.0001$. The distance between the plates is $h = 1$. Both plates are fixed with zero velocities at the top and bottom boundaries, where no calculations are needed. The steady numerical results are dependent on flow viscosity and shown in this figure through simulations of flows with three viscosities of $\nu_1 = 0.003$, $\nu_2 = 0.001$, and $\nu_3 = 0.0006$.

4.3. 2D Cavity Flow

The third test is a 2D cavity flow, which is a well-known complex flow within a simple geometry. The domain is a 1×1 square. The boundary conditions are that the top lid moves at a velocity of $u_x = u_0$ and $u_y = 0$ with $u_0 = 1$; the other three sides are fixed, or no slip boundary condition is applied, i.e., $u_x = 0$ and $u_y = 0$. The Reynolds number $Re = u_0/\nu = 1000$. We use $\delta x = 0.0025$ or 400×400 lattices in the simulation, which is carried out on the inside of the cavity excluding the four sides where the velocities are retained as boundary conditions. After the steady solution is obtained, the flow pattern in the velocity vectors is shown in Figure 3, which closely agrees with the well-known study by Ghia et al. [32]. The results are further compared against their numerical solutions for velocity profiles of u_x and u_y along the y and x directions through the geometric centre of the cavity in Figures 4 and 5, respectively, demonstrating very good agreements.

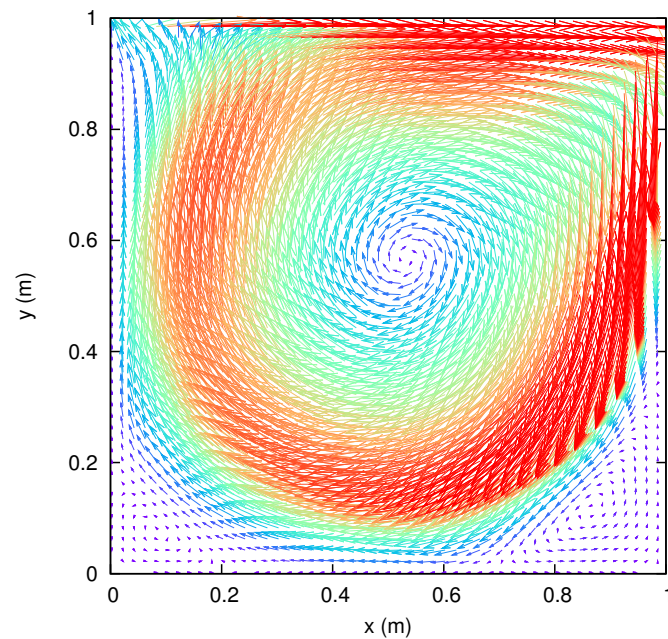


Figure 3. 2D cavity flow within a 1×1 square for $Re = 1000$. The top lid moves at a velocity of $u_x = 1$ and $u_y = 0$, and the other three sides are fixed, or no slip boundary condition is applied. After the steady solution is obtained, the flow pattern in velocity vectors shows a primary vortex and two secondary vortices.

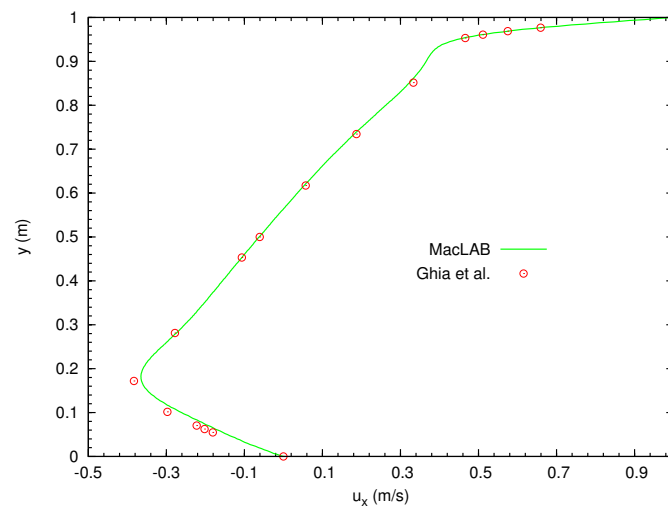


Figure 4. 2D cavity flow within a 1×1 square for $Re = 1000$. The top lid moves at a velocity of $u_x = 1$ and $u_y = 0$, and the other three sides are fixed, or no slip boundary condition is applied. After the steady solution is obtained, the comparison of the velocity u_x profile along the y direction through the geometric centre of the cavity with the numerical solution by Ghia et al. [32], showing good agreement.

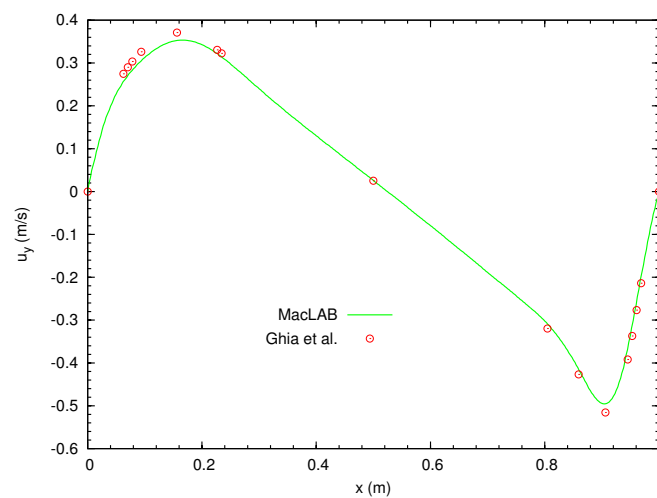


Figure 5. 2D cavity flow within a 1×1 square for $Re = 1000$. The top lid moves at velocity of $u_x = 1$ and $u_y = 0$, and the other three sides are fixed, or no slip boundary condition is applied. After the steady solution is obtained, the comparison of the velocity u_y profile along the x direction through the geometric centre of the cavity with the numerical solution by Ghia et al. [32], showing good agreement.

4.4. 2D Taylor–Green Vortex

The fourth test is a 2D Taylor–Green vortex. This is an unsteady flow driven by decaying vortices for which there is an exact solution of the incompressible Navier–Stokes equations, and it is often applied to the validation of a numerical method for the solution to the incompressible Navier–Stokes equations. The initial conditions are $u_x(x, y, 0) = -u_0 \cos(x) \sin(y)$ and $u_y(x, y, 0) = u_0 \sin(x) \cos(y)$. The analytical solutions are $u_x(x, y, t) = -u_0 \cos(x) \sin(y) \exp(-2\nu t)$ and $u_y(x, y, t) = u_0 \sin(x) \cos(y) \exp(-2\nu t)$. The time for an unsteady flow from the initial state is accumulated by its increase with time step δt . We use $\delta x = 0.157$ or 40×40 lattices for a square domain of $2\pi \times 2\pi$ with a kinematic viscosity of $\nu = 0.0314$ and $u_0 = 0.05$, which gives the Reynolds number of $Re = 2\pi u_0 / \nu = 10$. The periodic boundary conditions are used. The simulation is run for the total time of 30 s. The velocity field is plotted in Figure 6, showing the correct flow pattern. The velocity profiles for u_x at $x = \pi$ and u_y at $x = \pi/2$ along the y direction are depicted and compared with the analytical solutions in Figure 7, showing excellent agreements and confirming the accuracy of the method for an unsteady flow.

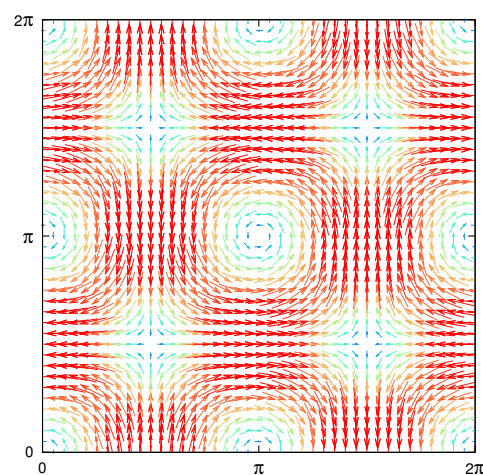


Figure 6. Taylor–Green vortex within the $2\pi \times 2\pi$ domain for $Re = 10$. The initial conditions are $u_x(x, y, 0) = -u_0 \cos(x) \sin(y)$ and $u_y(x, y, 0) = u_0 \sin(x) \cos(y)$ with $u_0 = 0.05$. The periodic boundary conditions are used. Shown here is the flow pattern in velocity vectors at $t = 30$ s, the same vortex pattern remaining as that at the initial state.

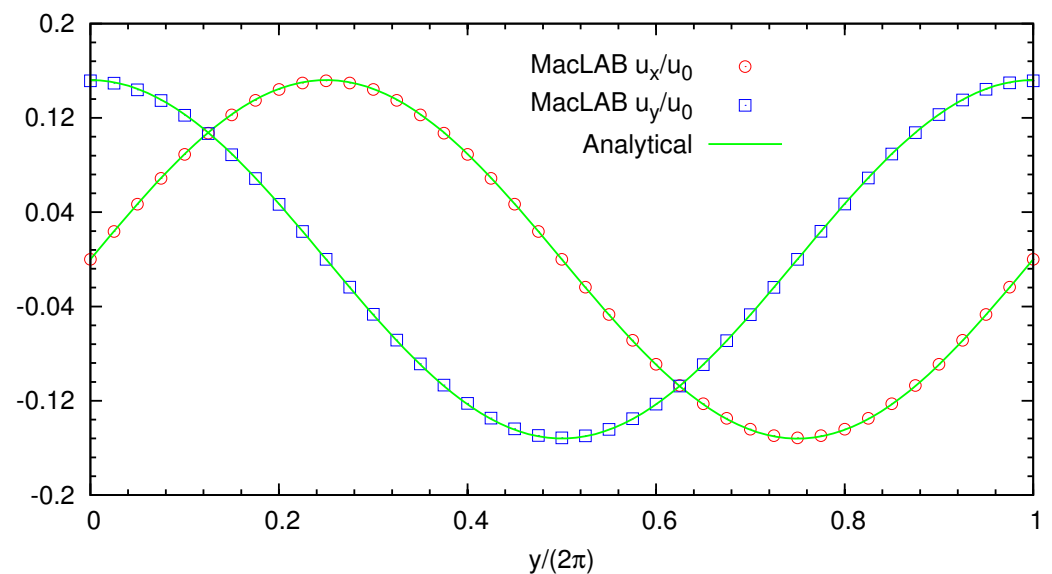


Figure 7. Taylor–Green vortex within the $2\pi \times 2\pi$ domain for $Re = 10$. The initial conditions are $u_x(x, y, 0) = -u_0 \cos(x) \sin(y)$ and $u_y(x, y, 0) = u_0 \sin(x) \cos(y)$ with $u_0 = 0.05$. The periodic boundary conditions are used. Shown here is the comparisons of the relative velocity profiles for u_x/u_0 at $x = \pi$ and u_y/u_0 at $x = \pi/2$, matching the analytical solutions of $u_x(x, y, t) = -u_0 \cos(x) \sin(y) \exp(-2\nu t)$ and $u_y(x, y, t) = u_0 \sin(x) \cos(y) \exp(-2\nu t)$ at $t = 30$ s.

4.5. 3D Cavity Flow

The final test is a 3D cavity flow. This is again a well-known complex flow involving 3D vortices within a simple cube with dimensions of $1 \times 1 \times 1$ in streamwise direction x , spanwise direction y and vertical direction z . No-slip boundary conditions, i.e. $u_x = 0$, $u_y = 0$ and $u_z = 0$, are applied to five fixed sides except for the top lid, where $u_x = u_0$, $u_y = 0$ and $u_z = 0$ with $u_0 = 1$ are specified. The Reynolds number is $Re = u_0/\nu = 400$. $\delta x = 0.004$ or total lattices of $250 \times 250 \times 250$ are used, and the simulation is undertaken only within the cube, excluding the boundaries where the velocities are retained. After the steady solution is reached, the flow characterises are displayed through the two-dimensional planar projections of the velocity vector field on the x - z , y - z and x - y centroidal planes of the cube in Figures 8–10, respectively, demonstrating flow patterns in good agreement with those by Wong and Baker [33]. In addition, the distribution of the velocity component u_x on the vertical plane centerline is widely used as a 3D lid-driven cavity benchmark test. We compare this velocity component against the results by Wong and Baker [33] and also by Jiang et al. [34] in Figure 11, showing good agreements.

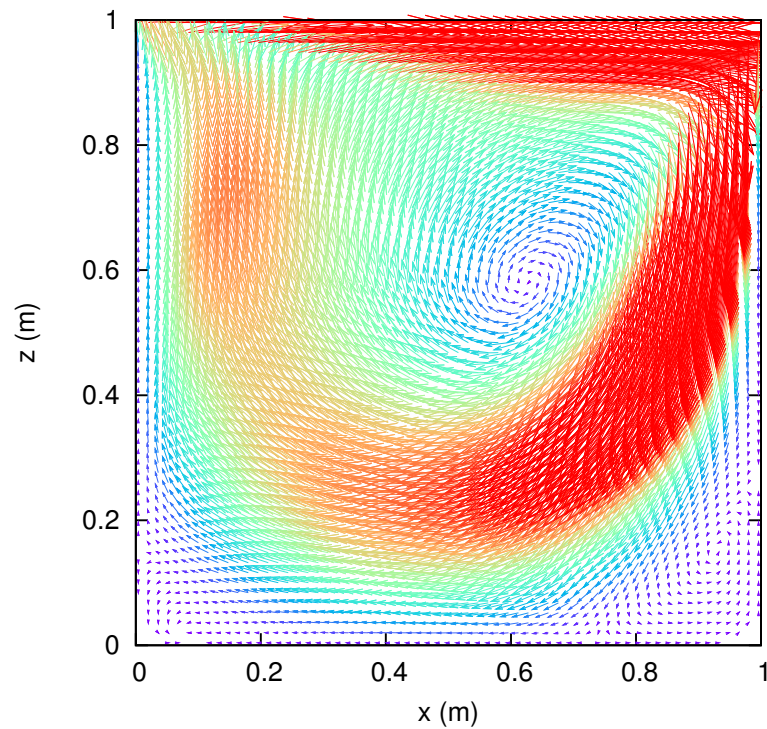


Figure 8. 3D cavity flow within a $1 \times 1 \times 1$ cube for $Re = 400$. The top lid moves at a velocity of $u_x = 1, u_y = 0$ and $u_z = 0$, and the other five sides are fixed, or no slip boundary condition is applied. After the steady solution is reached, the flow pattern in vectors in the $x - z$ centroidal plane show the primary and secondary vortices.

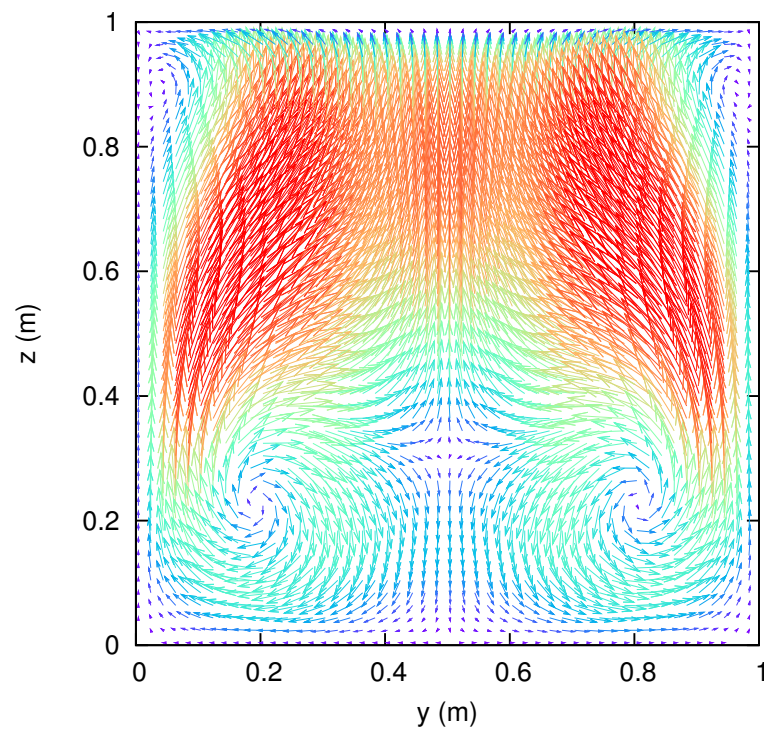


Figure 9. 3D cavity flow within a $1 \times 1 \times 1$ cube for $Re = 400$. The top lid moves at a velocity of $u_x = 1, u_y = 0$ and $u_z = 0$, and the other five sides are fixed, or no slip boundary condition is applied. After the steady solution is reached, the flow pattern in vectors in the $y - z$ centroidal plane shows one pair of strong secondary vortices at the bottom and one pair of weak secondary vortices at the top.

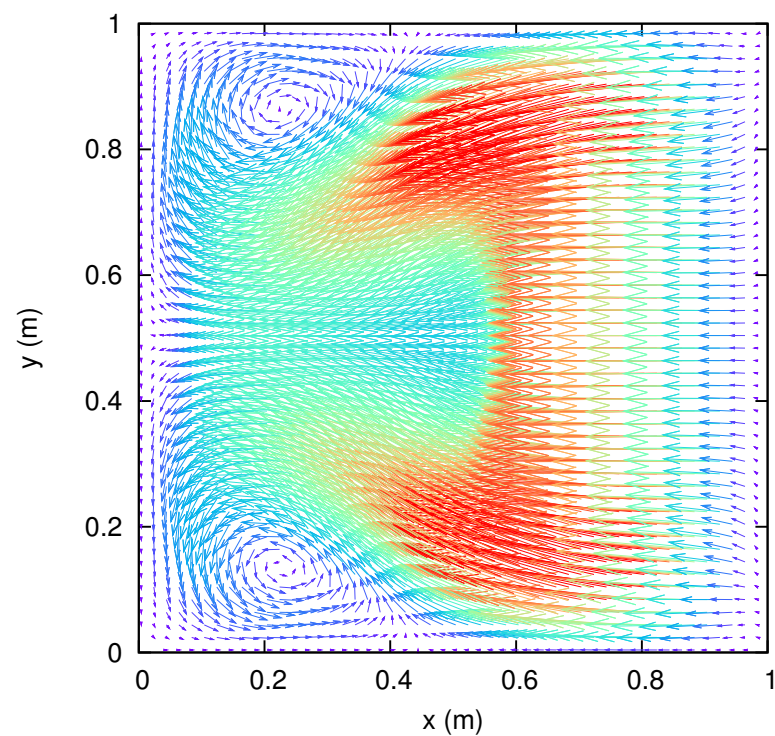


Figure 10. 3D cavity flow within a $1 \times 1 \times 1$ cube for $Re = 400$. The top lid moves at a velocity of $u_x = 1, u_y = 0$ and $u_z = 0$, and the other five sides are fixed, or no slip boundary condition is applied. After the steady solution is reached, the flow pattern in vectors in the $x - y$ centroidal plane shows a pair of third vortices close to the inflow boundary.

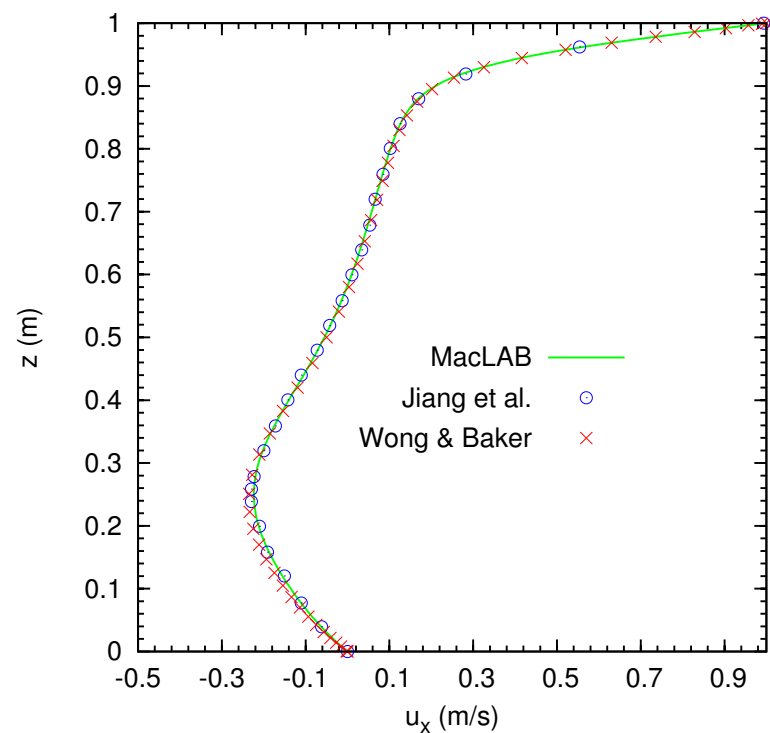


Figure 11. 3D cavity flow within a $1 \times 1 \times 1$ cube for $Re = 400$. The top lid moves at a velocity of $u_x = 1, u_y = 0$ and $u_z = 0$, and the other five sides are fixed, or no slip boundary condition is applied. After the steady solution is reached, the comparisons of the distribution of the velocity component u_x on the vertical plane centerline with the results by Wong and Baker [33] and Jiang et al. [34].

5. Conclusions

A novel macroscopic lattice Boltzmann method (MacLAB) for the Navier–Stokes equations is proposed and validated. It can accurately simulate fluid flows using only the lattice size, bringing the LBM into a precise lattice Boltzmann method and revolutionising the standard LBM theory. This takes the research on the method into a new era in which future work may focus on improving the accuracy of or formulating a new local equilibrium distribution function. The particle speed is determined through the viscosity and lattice size, and the time step δt is calculated as $\delta t = \delta x/e$. The model is unconditional stable as long as the valid condition for the local equilibrium distribution function holds. When a super-fast computer such as a quantum computer becomes available in the future, this will enable the MacLAB to take any fine lattice to simulate most challenging fluid flows such as turbulent flows without relying on a turbulent flow model. All these make the method an automatic simulator for fluid flows in all relevant problems. The method is straightforward to extend for resolving other physical problems in different disciplines such as chemical and environmental engineering.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The author declares no conflict of interest.

References

- Chen, S.; Doolen, G.D. Lattice Boltzmann Method for Fluid Flows. *Annu. Rev. Fluid Mech.* **1998**, *30*, 329–364.
- Hardy, J.; de Pazzis, O.; Pomeau, Y. Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions. *Phys. Rev. A* **1976**, *13*, 1949–1961.
- Frisch, U.; Hasslacher, B.; Pomeau, Y. Lattice-Gas Automata for the Navier–Stokes Equation. *Phys. Rev. Lett.* **1986**, *56*, 1505–1508.
- Chopard, B.; Droz, M. *Cellular Automata Modeling of Physical Systems*; Cambridge University Press: Cambridge, UK, 1998.
- Rivet, J.P.; Boon, J.P. *Lattice Gas Hydrodynamics*; Cambridge University Press: Cambridge, UK, 2001.
- McNamara, G.R.; Zanetti, G. Use of the Boltzmann equation to simulate lattice-gas automata. *Phys. Rev. Lett.* **1988**, *61*, 2332–2335.
- Higuera, F.; Jiménez, J. Boltzmann approach to lattice gas simulations. *Eur. Lett.* **1989**, *9*, 663–668.
- Noble, D.R.; Chen, S.; Georgiadis, J.G.; Buckius, R.O. A consistent hydrodynamic boundary condition for the lattice Boltzmann method. *Phys. Fluids* **1995**, *7*, 203–209.
- Qian, Y.H. Lattice Gas and Lattice Kinetic Theory Applied to the Navier–Stokes Equations. Ph.D. Thesis, Université Pierre et Marie Curie, Paris, France, 1990.
- Chen, S.; Chen, H.D.; Martinez, D.; Matthaeus, W. Lattice Boltzmann model for simulation of magnetohydrodynamics. *Phys. Rev. Lett.* **1991**, *67*, 3776–3779.
- Bhatnagar, P.L.; Gross, E.P.; Krook, M. A model for collision processes in gases. I: Small amplitude processes in charged and neutral one-component system. *Phys. Rev.* **1954**, *94*, 511–525.
- Zarghami, A.; den Akker, H.E.A.V. Thermohydrodynamics of an evaporating droplet studied using a multiphase lattice Boltzmann method. *Phys. Rev. E* **2017**, *95*, 043310.
- Mohamad, H.H.; Masoud, M. Continuous and Discrete Adjoint Approach Based on Lattice Boltzmann Method in Aerodynamic Optimization Part I: Mathematical Derivation of Adjoint Lattice Boltzmann Equations. *Adv. Appl. Math. Mech.* **2014**, *6*, 570–589.
- Shan, X.; Chen, H. Lattice Boltzmann model for simulating flows with multiple phases and components. *Phys. Rev. E* **1993**, *47*, 1815–1819, doi:10.1103/PhysRevE.47.1815.
- Chen, H.; Kandasamy, S.; Orszag, S.; Shock, R.; Succi, S.; Yakhot, V. Extended Boltzmann Kinetic Equation for Turbulent Flows. *Science* **2003**, *301*, 633–636.
- Golbert, D.R.; Blanco, P.J.; Clausse, A.; Feijóo, R.A. Tuning a lattice-Boltzmann model for applications in computational hemodynamics. *Med. Eng. Phys.* **2012**, *34*, 339–349.
- Javed, S.; Sohail, A.; Maqbool, K.; Butt, S.I.; Chaudhry, Q.A. The Lattice Boltzmann method and computational analysis of bone dynamics-I. *Complex Adapt. Syst. Model.* **2017**, *5*, 1–14.
- Chen, J.; Chai, Z.; Shi, B.; Zhang, W. Lattice Boltzmann method for filtering and contour detection of the natural images. *Comput. Math. Appl.* **2014**, *68*, 257–268.
- Finck, M.; Hänel, D.; Wlokas, I. Simulation of nasal flow by lattice Boltzmann methods. *Comput. Biol. Med.* **2007**, *37*, 739–749.

20. Zhou, J.G.; Haygarth, P.M.; Withers, P.J.A.; Macleod, C.J.A.; Falloon, P.D.; Beven, K.J.; Ockenden, M.C.; Forber, K.J.; Hollaway, M.J.; Evans, R.; et al. Lattice Boltzmann method for the fractional advection-diffusion equation. *Phys. Rev. E* **2016**, *93*, 043310.
21. d’Humières, D. Generalized lattice Boltzmann equations. In *Rarefied Gas Dynamics: Theory and Simulations, Progress in Astronautics and Aeronautics*; Shizgal, B.D., P.Weaver, D., Eds.; 1992; Volume 159, pp. 450–458.
22. Lallemand, P.; Luo, L.S. Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability. *Phys. Rev. E* **2000**, *61*, 6546–6562.
23. Ginzburg, I.; Verhaeghe, F.; d’Humières, D. Two-Relaxation-Time Lattice Boltzmann Scheme: About Parametrization, Velocity, Pressure and Mixed Boundary Conditions. *Commun. Comput. Phys.* **2008**, *3*, 427–478.
24. Geier, M.; Greiner, A.; Korvink, J.G. Cascaded digital lattice Boltzmann automata for high Reynolds number flow. *Phys. Rev. E* **2006**, *73*, 066705.
25. Geier, M.; Schönherr, M.; Pasquali, A.; Krafczyk, M. The cumulant lattice Boltzmann equation in three dimensions: Theory and validation. *Comput. Math. Appl.* **2015**, *70*, 507–547.
26. Chen, Z.; Shu, C.; Wang, Y.; Yang, L.M.; Tan, D. A Simplified Lattice Boltzmann Method without Evolution of Distribution Function. *Adv. Appl. Math. Mech.* **2017**, *9*, 1–22.
27. Succi, S. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*; Oxford University Press: Oxford, UK, 2001.
28. Wolf-Gladrow, D. *Lattice-Gas Cellular Automata and Lattice Boltzmann Models*; Springer: Berlin/Heidelberg, Germany, 2000.
29. Guo, Z.L.; Shu, C. *Lattice Boltzmann Method and Its Applications in Engineering*; World Scientific Publishing: Singapore, 2013.
30. Krüger, T.; Kusumaatmaja, H.; Kuzmin, A.; Shardt, O.; Silva, G.; Viggen, E.M. *The Lattice Boltzmann Method: Principles and Practice*; Springer: Berlin, Germany, 2017.
31. Zhou, J.G. *Lattice Boltzmann Methods for Shallow Water Flows*; Springer: Berlin, Germany, 2004.
32. Ghia, U.; Ghia, K.; Shin, C. High-Re Solutions for Incompressible Flow Using the Navier–Stokes Equations and a Multigrid Method. *J. Comput. Phys.* **1982**, *48*, 387–411.
33. Wong, K.L.; Baker, A.J. A 3D incompressible Navier–Stokes velocity-vorticity weak form finite element algorithm. *Int. J. Numer. Methods Fluids* **2002**, *38*, 99–123.
34. Jiang, B.N.; Lin, T.L.; Povinelli, L.A. Large scale computation of incompressible viscous flows by least-squares finite element method. *Comput. Methods Appl. Mech. Eng.* **1994**, *114*, 213–231.