2020

A Preconditioned Lattice Boltzmann Flux Solver for Steady Flows on Unstructured Hexahedral Grids

Brendan Walsh
Technological University Dublin, brendan.walsh@TUDublin.ie

Fergal Boyle
Technological University Dublin, fergal.boyle@tudublin.ie

Recommended Citation

Walsh, B. & Boyle. F.J. (2020) A Preconditioned Lattice Boltzmann Flux Solver for Steady Flows on Unstructured Hexahedral Grids, *Computers & Fluids,* Volume 210, 15 October 2020, DOI:10.1016/j.compfluid.2020.104634

Computers and Fluids 210 (2020) 104634

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

A preconditioned lattice Boltzmann flux solver for steady flows on unstructured hexahedral grids

Brendan Walsh*, Fergal J. Boyle

School of Mechanical and Design Engineering, College of Engineering and Built Environment, Technological University Dublin, Bolton Street, Dublin 1, Ireland

ARTICLE INFO

Article history:
Received 26 November 2019
Revised 21 March 2020
Accepted 19 June 2020
Available online 7 July 2020

Keywords:
Lattice Boltzmann Flux Solver
Unstructured Grids
Preconditioning
Steady Flows
Acceleration of Convergence

ABSTRACT

The lattice Boltzmann flux solver (LBFS), first introduced by Shu et al. (2014) on structured meshes, allows fluid flow problems to be solved on unstructured meshes discretised by the finite volume method. The solver calculates the macroscopic fluxes at the cell interfaces from a local reconstruction of the lattice Boltzmann solution. In this paper the LBFS is extended to three-dimensional unstructured hexahedral meshes and a preconditioned lattice Boltzmann flux solver (PLBFS) is presented. The PLBFS involves applying the preconditioning technique proposed by Guo (2004) to the LBFS and is achieved by modifying the equilibrium distribution function used to calculate the macroscopic fluxes at the cell interface. When the PLBFS is applied to steady flow problems, it is shown that convergence is significantly accelerated and the accuracy of predictions with unstructured grids is greatly improved when compared to the LBFS. This paper also introduces a strategy for choosing the optimal value of preconditioning factor with unstructured hexahedral meshes.

1. Introduction

In recent years the lattice Boltzmann method (LBM) has become a well established alternative for solving computational fluid dynamics (CFD) problems. It involves calculating the change in the density distribution of discrete particles at the mesoscopic level. The change in density distribution is due to the collision of the discrete particles and the subsequent streaming of the particles. Using a Chapman-Enskog expansion, it can be shown that the lattice Boltzmann equation (LBE) applied at a mesoscopic level is equivalent to the Navier-Stokes (NS) equations at a macroscopic level [1]. The LBM has many advantages compared with traditional NS equations solvers when solving transient problems as it does not involve the solution of the expensive Poisson equation for pressure and uses a highly algebraic and parrellisable approach to calculating densities and velocities in a flow problem. Limitations of the standard LBM include the necessary use of a uniform Cartesian grid and the restriction of its use to flow problems with low Mach numbers. The required use of a uniform Cartesian grid is a major restriction when it comes to the modelling of real life engineering problems such as the haemodynamics of biomedical devices and the aerodynamics of cars and aeroplanes. Issues encountered include approximating curved boundaries using a staircase approximation and domain wide refinement of the grid to resolve boundary layer fluid dynamics. Using a staircase approximation introduces a geometrical discretisation error while domain wide refinement of the grid leads to excessive computational effort. This has resulted in much research into fully unstructured LBMs which would enable the use of body fitted grids with complex geometries and have local refinement near boundaries.

The first attempt at an unstructured LBM was by Peng [2,3] and Xi [4]. This approach involved the integration of the differential form of the LBE in control volumes around the grid points. However it was found that the method suffered from significant instability issues. Over the years improvements to the stability of finite volume - lattice Boltzmann methods (FVLBM) have been made. Stiebler [5] introduced a least squares, linear reconstruction based upwind discretization scheme for the FVLBM. A total variation diminishing approach to the FVLBM was introduced by Patil [6] whereas Ubertini et al. [7] used a memory term to increase stability thresholds of the method. More recently Zarghami et al. [8] used upwind second order pressure biasing factors as flux correctors to improve stability.

An alternative innovative approach was proposed by Wang et al. [9] and Shu et al. [10] when they developed the lattice Boltzmann flux solver (LBFS). This involves discretising the domain into cells and calculating the macroscopic fluxes at the cell interface using

* Corresponding author.
E-mail addresses: Brendan.Walsh@TUDublin.ie (B. Walsh), fergal.boyle@TUDublin.ie (F.J. Boyle).

https://doi.org/10.1016/j.compfluid.2020.104634

a local reconstruction of the LBE. Unlike in the FVLBMs discussed above, the LBFS involves solving conservation equations for the macroscopic variables whereas the FVLBM involves solving conservation equations of the particle distributions. While the original method was implemented on non-uniform orthogonal meshes, lately it has been applied to fully unstructured 2D tetrahedral meshes by Pellerin et al. [11], Wu et al. [12] and more recently Liu et al. [13]. The latter introduces a high order least-squares approach that improves the order of accuracy of the LBFS from 2nd order to 4th order. The LBFS also been extended to model fluid-structure interactions on 3D geometries through the use of the immersed-boundary method by Wang et al. [14].

The LBFS was adopted in this work due to its advantages over the FVLBM with regards to stability and the implementation of boundary conditions. The LBFS does not require the use of pressure biasing factors or other such schemes to achieve stability at higher Reynolds numbers. The LBFS also allows the direct implementation of physical boundary conditions whereas the FVLBM relies on the standard LBM family of boundary conditions which are more difficult to implement.

The LBFS shares many traits with the family of artificial compressibility methods (ACM) which was initially introduced by Chorin [15]. In the ACM, an artificial relationship is introduced between pressure and density variables. This changes the nature of the governing equations from mixed elliptic/parabolic nature into the hyperbolic/parabolic. This enables the use of efficient time-stepping schemes in calculating the steady state solution of incompressible flows. The LBFS differs from the ACM, in that the relationship between pressure and density is set by an arbitrary parameter in the ACM [15]. More recently efforts by Turkel [16] and Malan [17] have provided a means of calculating the optimum value of this arbitrary parameter. In the LBFS the relationship between pressure and density is defined by the lattice discretisation and there is no need to optimise the arbitrary parameter required in the ACM. The ACM and LBFS both solve the NS equations, with the ACM using either a finite difference [18] or finite volume [17] spatial discretisation. The LBFS uses a finite volume approach. The ACM solves the NS equations but uses a perturbed continuity equation and, as described by He et al [19], this has no physical meaning in incompressible flow. In contrast the LBFS is a âæweaklyâ𝔤 compressible method and its continuity equation has a physical meaning. There is also a difference in the stencils used in the calculation of the fluxes; the original ACM proposed a central differencing or leapfrog scheme [15] and more recently 3rd order upwind and 4th order central compact schemes have been used [20]. In comparison the LBFS uses a local reconstruction of the LBE to calculate the fluxes. Further differences arise in that the ACM requires the incorporation of artificial dissipation whereas the LBFS does not [18][17]. According to Kozela[21], the ACM is shown to require dual-stepping or similar implicit methods to effect real time accuracy for unsteady flows whereas the LBFS can effect real time accuracy using explicit and implicit methods. Finally, while the original ACM can be thought of as a preconditioning approach to the continuity equation, this has also been extended to introduce preconditioning to the momentum equations [22]. This has the impact of improving the convergence and robustness of the ACM approach.

A key characteristic of the standard LBM is the grid independent ”compressibility” error which is directly proportional to the Mach number [23]. As a result LBM simulations limit this error by keeping the Mach number small, typically this involves keeping the Mach number less than 0.4 [24]. Reducing the Mach number has the knock on effect of increasing the disparity between the acoustic and convective wave speeds [25]. As the standard LBM typically employs explicit time-stepping, the Courant-Friedrichs-Levy (CFL) condition should be satisfied to ensure stability. At low Mach numbers this requires a time step inversely proportional to the largest eigenvalue in the system which is approximately the speed of sound. However the convective wave propagates information through the domain at the much lower fluid speed. As a result a large amount of time steps are required to reach steady-state convergence.

In recent years many researchers have made attempts to accelerate convergence of the standard LBM to steady-state convergence. A time independent formulation of the LBM has been proposed by Bernaschi et al. [26]. The steady-state solution is solved through iterative methods and this approach has also been implemented by Verberg [27] and Noble [28]. Tolke et al. [29] expanded on the time independent approach by using a multigrid approach to solve an implicit second-order finite difference scheme. An alternative approach for accelerating convergence of the LBM to steady-state is using implicit schemes to discretise the time-dependent equation. This allows larger time steps to be used and has been implemented by Lee [30], Seta [31] and Tolke et al. [32]. Further to this Mavriplis [33] proposed a non-linear form of multigrid solver with a non-linear LBE time-stepping scheme. These methods all accelerate convergence of the LBM to steady-state but at the cost of increased complexity compared to the standard LBM.

Preconditioning is another time-dependent steady-state acceleration technique which was successfully applied to the LBM by Guo [34] originally. This approach follows the same principal as the preconditioning method developed by Turkel [25] to solve the incompressible and low speed compressible NS equations. As mentioned above, at low Mach numbers there is a disparity between the acoustic and convective wave speeds. The ratio between these wave speeds is known as the condition number. The lower the condition number, the faster that a solution will converge to the steady-state solution. Preconditioning involves altering the eigenvalues of the NS equations to reduce the condition number. Guo implemented preconditioning in the LBM by applying a single preconditioning factor (γ-preconditioner) to the equilibrium distribution function. For steady flows this results in an equivalent form of the NS equations with a reduced condition number, which reduces the number of iterations required to reach steady-state.

There have been many additions to Guo's original work. Premnath et al. [35] extended the preconditioning approach to allow for forcing terms in force-driven fluid flow problems. Izquierdo et al. [36] extended it to the generalised form of the LBM including the multiple-relaxation time LBM. In this work a second preconditioning factor (β-preconditioner) was also used to improve the efficacy of preconditioning. They also investigated optimal values of the preconditioning values for the LBM and gave apriori guidelines for such values [37]. More recently the approach has been extended to a noncascaded central moments LBM [38], a cascaded LBM [39] and a Gailliean invariant cascaded LBM [40]. Meng et al. have also used an improved preconditioned multiple-relaxation time LBM to model flow through porous media [41].

In this work the γ-preconditioning approach of Guo's work is adopted as it has similar performance to the β-preconditioning approach of Izquierdo's work whilst being more efficient and also simpler to implement. It is applied to a LBFS on unstructured hexahedral grids and its performance is evaluated on two fluid problems: 3D lid-driven cavity flow and 3D flow over a stationary cylinder. The structure of the remainder of the paper is as follows: Section 2 introduces the LBFS and the preconditioned lattice Boltzmann flux solver (PLBFS), Section 3 discusses the use of preconditioning with unstructured grids and proposes a strategy for the optimal choice of preconditioning parameter, in Section 4 the results of numerical simulations are discussed, and a summary and conclusions are presented in Section 5.

2. Methodology

2.1. Governing equations

2.1.1. Navier-stokes equations

If body forces are neglected, the 3D unsteady NS equations can be written in conservative form for a finite control volume in a Cartesian coordinate system as:

$$\frac{\partial}{\partial t}\int_V (\mathbf{Q}dV) + \int_S \mathbf{F}\cdot\mathbf{n}\ dS = 0 \tag{1}$$

where t is time, V is the cell volume, S is the cell surface, $\mathbf{n}$ is the outward normal to element of area dS, $\mathbf{Q}$ is the vector of conserved variables $[\rho,\ \rho u,\ \rho v,\ \rho w]$, ρ is the density, and u, v and w are the components of the velocity vector $\mathbf{V}$ in the x, y and z directions respectively. Assuming a Newtonian, isotropic, isothermal fluid allows the flux tensor $\mathbf{F}$ to be defined as follows:

$$\mathbf{F} = \begin{bmatrix} \rho u & \rho v & \rho w \\ \rho u^2 + p - \mu\left(2\frac{\partial u}{\partial x}\right) & \rho uv - \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) & \rho uw - \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \\ \rho vu - \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) & \rho v^2 + p - \mu\left(2\frac{\partial v}{\partial y}\right) & \rho vw - \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \\ \rho wu - \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) & \rho wv - \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) & \rho w^2 + p - \mu\left(2\frac{\partial w}{\partial z}\right) \end{bmatrix} \tag{2}$$

where p is the static pressure and μ is the dynamic viscosity. The above equations can be used to simulate incompressible flow, when the Mach number is low and density variation is small.

Using a cell-centred finite volume approach for spatial discretisation, Eq. (1) is applied to every cell in the computational domain yielding a set of semi-discrete equations:

$$\frac{d\mathbf{Q}_i}{dt} + \frac{1}{V_i}\sum_{n=1}^{N_{faces}} \mathbf{F}_{i,n}\cdot\mathbf{n}_{i,n}S_{i,n} = 0 \tag{3}$$

where for cell i, $\mathbf{Q}_i$ is the vector of conserved variables, V_i is the cell volume, N_{faces} is the number of cell faces, $\mathbf{n}_{i,n}$ is the outward normal of face n, $S_{i,n}$ is the area of face n on cell, and $\mathbf{F}_{i,n}$ is the flux tensor at the centroid of face n.

2.1.2. Lattice boltzmann flux solver

In the LBFS the flux tensor at each cell interface is evaluated from a local reconstruction of the lattice Boltzmann solution (LBS) at the cell interface. The cell interface is defined as the centroid of the shared face between the two adjoining cells. In this work the single relaxation time Bhatnagar-Gross-Krook (BGK) collision model is employed. This leads to the following formulation for the flux tensor $\mathbf{F}$:

$$\mathbf{F} = \sum_{\alpha=0}^{N} f_\alpha^{eq}\begin{bmatrix} e_{\alpha,x} & e_{\alpha,y} & e_{\alpha,z} \\ e_{\alpha,x}e_{\alpha,x} & e_{\alpha,y}e_{\alpha,x} & e_{\alpha,z}e_{\alpha,x} \\ e_{\alpha,x}e_{\alpha,y} & e_{\alpha,y}e_{\alpha,y} & e_{\alpha,z}e_{\alpha,y} \\ e_{\alpha,x}e_{\alpha,z} & e_{\alpha,y}e_{\alpha,z} & e_{\alpha,z}e_{\alpha,z} \end{bmatrix} + \sum_{\alpha=0}^{N}\left[1 - \frac{1}{2\tau_s}\right]f_\alpha^{neq}\begin{bmatrix} 0 & 0 & 0 \\ e_{\alpha,x}e_{\alpha,x} & e_{\alpha,y}e_{\alpha,x} & e_{\alpha,z}e_{\alpha,x} \\ e_{\alpha,x}e_{\alpha,y} & e_{\alpha,y}e_{\alpha,y} & e_{\alpha,z}e_{\alpha,y} \\ e_{\alpha,x}e_{\alpha,z} & e_{\alpha,y}e_{\alpha,z} & e_{\alpha,z}e_{\alpha,z} \end{bmatrix} \tag{4}$$

where $e_{\alpha,x}$, $e_{\alpha,y}$ and $e_{\alpha,z}$ are the x, y and z components respectively of the particle velocity vector $\mathbf{e}_\alpha$ in the α direction, τ_s is the standard relaxation factor, f_α is the density distribution function in the α direction, f_α^{eq} is the equilibrium density distribution function in the α direction, f_α^{neq} is the non-equilibrium density distribution function in the α direction and is equal to $f_\alpha - f_\alpha^{eq}$, and N is the number of velocities in the lattice model.

To implement the LBS at the cell interface, one must first choose a lattice velocity set to define $\mathbf{e}_\alpha$. Velocity sets are usually denoted in DdQq form where d denotes the number of spatial dimensions covered by the velocity set and q is the number of velocities in the set. In this work the D3Q15 velocity set is chosen due to its computational efficiency and is shown in Table 1. **i, j** and **k** are the unit vectors in the x, y and z directions respectively, $c = \delta x/\delta t$, δx is the lattice spacing along each Cartesian axis and δt is the streaming time step. Typically δx is set equal to δt giving c equal to 1 and this approach is adopted in this work. In physical terms c is considered to be the lattice velocity. When the D3Q15 lattice velocity set is applied to the cell interface, the lattice is formed by 15 nodes. The Cartesian coordinates of these lattice nodes can be represented in terms of lattice velocities. A 3D illustration of the D3Q15 lattice velocity set implemented at a cell interface is shown in Fig. 1. In this figure the Cartesian coordinates of the two cell centres are referred to as $\mathbf{r}_i$ and $\mathbf{r}_{i+1}$ respectively. The cell interface is referred to as $\mathbf{r}$.

The LBE with the BGK collision model [42] can be applied to model a Newtonian fluid at the cell interface. This enables the finding of the equilibrium and non-equilibrium density distribution functions required to calculate the flux tensor. The LBE is given by:

$$\begin{aligned} f_\alpha(\mathbf{r},t) = {} & f_\alpha(\mathbf{r} - \mathbf{e}_\alpha\delta t, t - \delta t) \\ & + \frac{f_\alpha^{eq}(\mathbf{r} - \mathbf{e}_\alpha\delta t, t - \delta t) - f_\alpha(\mathbf{r} - \mathbf{e}_\alpha\delta t, t - \delta t)}{\tau_s} \\ & \times for\ \alpha = 0, 1 \ldots N \end{aligned} \tag{5}$$

where the standard relaxation factor τ_s is related to the viscosity and controls the influence of the viscous fluxes on the momentum equation. In physical terms it is the rate at which f_α tends to the equilibrium density distribution function f_α^{eq} and is given by:

$$\tau_s = \frac{\nu}{c_s^2\delta t} + 0.5 \tag{6}$$

where ν is the kinematic viscosity and, as per the literature, c_s is the speed of sound in the lattice grid equalling $c/\sqrt{3}$ for a D3Q15 lattice model. The equilibrium density distribution function f_α^{eq} is given by a Hermite series expansion of the Maxwell-Boltzmann distribution:

$$f_\alpha^{eq}(\mathbf{r},t) = \rho w_\alpha\left[1 + \frac{\mathbf{e}_\alpha\cdot\mathbf{V}}{c_s^2} + \frac{(\mathbf{e}_\alpha\cdot\mathbf{V})^2 - (c_s|\mathbf{V}|)^2}{2c_s^4}\right] + O(\mathbf{V}^3) \tag{7}$$

where the weights w_α are given as:

$$w_\alpha = \begin{bmatrix} 2/9 \\ 1/9 \\ 1/72 \end{bmatrix} \begin{matrix} \alpha = 0 \\ \alpha = 1 - 6 \\ \alpha = 7 - 14 \end{matrix} \tag{8}$$

As shown by Shu et al. [14], by using the Chapman-Enskog expansion and the linear Taylor series expansion of f_α^{eq}, the non-equilibrium density distribution function f_α^{neq} can be calculated as follows:

$$f_\alpha^{neq}(\mathbf{r},t) = -\tau_s\left[f_\alpha^{eq}(\mathbf{r},t) - f_\alpha^{eq}(\mathbf{r} - \mathbf{e}_\alpha\delta t, t - \delta t)\right] \tag{9}$$

In the localised reconstruction of the LBE at the cell interface, the flux tensor is now dependent on the post-streaming and pre-streaming equilibrium density distribution functions $f_\alpha^{eq}(\mathbf{r},t)$ and $f_\alpha^{eq}(\mathbf{r} - \mathbf{e}_\alpha\delta t, t - \delta t)$ respectively. These in turn are dependent on the density and velocity values at each lattice node. As f_α is a density distribution function, the density can be calculated by summing f_α up over all lattice velocities and similarly the momentum

Table 1
Explicit representation of the D3Q15 velocity set.

	α	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$\mathbf{e}_\alpha$	$c \cdot \mathbf{i}$	0	1	-1	0	0	0	0	1	-1	1	-1	1	-1	-1	1
	$c \cdot \mathbf{j}$	0	0	0	1	-1	0	0	1	-1	1	-1	-1	1	1	-1
	$c \cdot \mathbf{k}$	0	0	0	0	0	1	-1	1	-1	-1	1	1	-1	1	-1

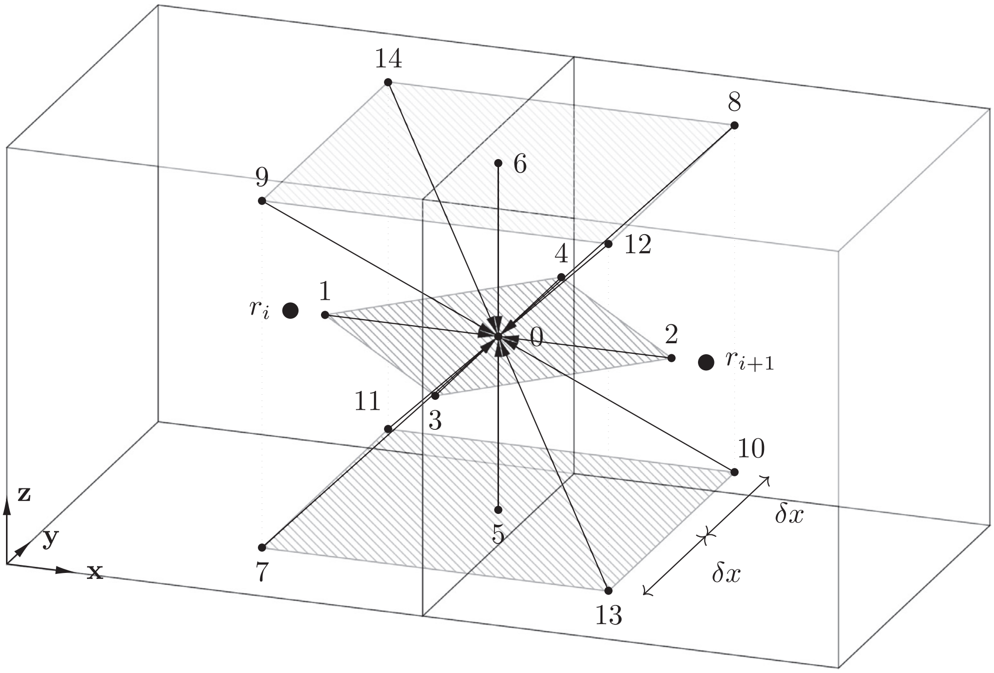


Fig. 1. Local reconstruction of the LBS at a cell interface implementing a D3Q15 lattice velocity set.

can be calculated by summing up the first moment of f_α, i.e:

$$\rho = \sum_{\alpha=0}^{N} f_\alpha$$
$$\rho\mathbf{V} = \sum_{\alpha=0}^{N} f_\alpha \mathbf{e}_\alpha \tag{10}$$

From the equation of state for an isothermal ideal gas, the pressure can be found using:

$$p = \rho c_s^2 \tag{11}$$

When a gas has been left alone for a sufficiently long period of time, it is assumed that f_α will reach an equilibrium which is defined by f_α^{eq}. We can make this assumption as collisions tend to even out the angular distribution of particle velocities in a gas around a mean velocity. As this convergence to equilibrium must conserve mass and momentum at all locations, equilibrium values can also be used to calculate the macroscopic density and momentum at any location as follows:

$$\rho = \sum_{\alpha=0}^{N} f_\alpha^{eq}$$
$$\rho\mathbf{V} = \sum_{\alpha=0}^{N} f_\alpha^{eq} \mathbf{e}_\alpha \tag{12}$$

The pre-streaming equilibrium density distribution function $f_\alpha^{eq}(\mathbf{r} - \mathbf{e}_\alpha \delta t, t - \delta t)$ is simply calculated by interpolating the macroscopic values from neighbouring cells at time $t - \delta t$:

$$\rho(\mathbf{r} - \mathbf{e}_\alpha \delta t) = \begin{cases} \rho(\mathbf{r}_i) + (\mathbf{r} - \mathbf{e}_\alpha \delta t - \mathbf{r}_i) \cdot \nabla \rho(\mathbf{r}_i) & \text{when } \mathbf{r} - \mathbf{e}_\alpha \delta t \text{ in cell } i \\ \rho(\mathbf{r}_{i+1}) + (\mathbf{r} - \mathbf{e}_\alpha \delta t - \mathbf{r}_{i+1}) \cdot \nabla \rho(\mathbf{r}_{i+1}) & \text{when } \mathbf{r} - \mathbf{e}_\alpha \delta t \text{ in cell } i+1 \end{cases} \tag{13}$$

$$\mathbf{V}(\mathbf{r} - \mathbf{e}_\alpha \delta t) = \begin{cases} \mathbf{V}(\mathbf{r}_i) + (\mathbf{r} - \mathbf{e}_\alpha \delta t - \mathbf{r}_i) \cdot \nabla \mathbf{V}(\mathbf{r}_i) & \text{when } \mathbf{r} - \mathbf{e}_\alpha \delta t \text{ in cell } i \\ \mathbf{V}(\mathbf{r}_{i+1}) + (\mathbf{r} - \mathbf{e}_\alpha \delta t - \mathbf{r}_{i+1}) \cdot \nabla \mathbf{V}(\mathbf{r}_{i+1}) & \text{when } \mathbf{r} - \mathbf{e}_\alpha \delta t \text{ in cell } i+1 \end{cases} \tag{14}$$

Inputting these values into Eq. (7) gives $f_\alpha^{eq}(\mathbf{r} - \mathbf{e}_\alpha \delta t, t - \delta t)$. The next step is to find a value for $f_\alpha^{eq}(\mathbf{r}, t)$. This is simply done by finding $\rho(\mathbf{r}, t)$ and $\mathbf{V}(\mathbf{r}, t)$. As mass and momentum are conserved $\rho(\mathbf{r}, t)$ and $\mathbf{V}(\mathbf{r}, t)$ can be calculated at the cell interface by summing the pre-streaming equilibrium distribution functions, i.e.:

$$\rho(\mathbf{r}, t) = \sum_{\alpha=0}^{N} f_\alpha^{eq}(\mathbf{r} - \mathbf{e}_\alpha \delta t, t - \delta t)$$
$$\rho(\mathbf{r}, t)\mathbf{V}(\mathbf{r}, t) = \sum_{\alpha=0}^{N} f_\alpha^{eq}(\mathbf{r} - \mathbf{e}_\alpha \delta t, t - \delta t)\mathbf{e}_\alpha \tag{15}$$

Inputting these values of $\rho(\mathbf{r}, t)$ and $\mathbf{V}(\mathbf{r}, t)$ into Eq. (7) will give $f_\alpha^{eq}(\mathbf{r}, t)$. Once $f_\alpha^{eq}(\mathbf{r}, t)$ and $f_\alpha^{eq}(\mathbf{r} - \mathbf{e}_\alpha \delta t, t - \delta t)$ have been found, $f_\alpha^{neq}(\mathbf{r}, t)$ can be calculated and the fluxes at the cell interface calculated using Eq. (4).

2.1.3. Gradient calculation

A key element of the LBFS is the accurate calculation of the gradient of the macroscopic variables at each cell centre as these gradients are then used in Eq. (13) and Eq. (14) to initialise the local LBS at the cell interface. An inaccurate calculation of the gradient will lead to a local LBS that does not reflect the flow field correctly. Two prominent methods used to calculate gradients on unstructured grids are the Green-Gauss and Least-Squares methods. Shima et al. [43] note how the Green-Gauss method is only fully accurate on uniform and symmetric grids but performs better on the thin and curved meshes with high aspect ratio cells that often exist within boundary layers for high Reynolds number flow calculations. Shima proposed a new hybrid Green-Gauss/Weighted-Least-Squares (GLSQ) approach for calculating the gradient. The GLSQ involves a geometry-dependent switch which applies the Green-Gauss method to those cells with a high aspect ratio and the Least-Squares method to cells with a low aspect ratio. This approach also has the benefits of ensuring monotonicity at cell interfaces, increasing the robustness of the solver, and allowing the accurate handling of hanging nodes in the gradient calculation. This is for a trivial increase in computational cost compared to the Least-Squares method. For these reasons the GLSQ approach is adopted in this work and is explained in detail in Appendix A.

2.1.4. Preconditioned lattice boltzmann flux solver

The PLBFS takes the same form as the LBFS with the preconditioned flux tensor $\mathbf{F}_p$ is given by:

$$\mathbf{F}_p = \sum_{\alpha=0}^{N} f_\alpha^{eq} \begin{bmatrix} e_{\alpha,x} & e_{\alpha,y} & e_{\alpha,z} \\ e_{\alpha,x}e_{\alpha,x} & e_{\alpha,y}e_{\alpha,x} & e_{\alpha,z}e_{\alpha,x} \\ e_{\alpha,x}e_{\alpha,y} & e_{\alpha,y}e_{\alpha,y} & e_{\alpha,z}e_{\alpha,y} \\ e_{\alpha,x}e_{\alpha,z} & e_{\alpha,y}e_{\alpha,z} & e_{\alpha,z}e_{\alpha,z} \end{bmatrix} + \sum_{\alpha=0}^{N} \left[1 - \frac{1}{2\tau_p}\right] f_\alpha^{neq} \begin{bmatrix} 0 & 0 & 0 \\ e_{\alpha,x}e_{\alpha,x} & e_{\alpha,y}e_{\alpha,x} & e_{\alpha,z}e_{\alpha,x} \\ e_{\alpha,x}e_{\alpha,y} & e_{\alpha,y}e_{\alpha,y} & e_{\alpha,z}e_{\alpha,y} \\ e_{\alpha,x}e_{\alpha,z} & e_{\alpha,y}e_{\alpha,z} & e_{\alpha,z}e_{\alpha,z} \end{bmatrix} \quad (16)$$

where f_α^{eq} is defined as

$$f_\alpha^{eq}(\mathbf{r}, t) = \rho w_\alpha \left[1 + \frac{\mathbf{e}_\alpha \cdot \mathbf{V}}{c_s^2} + \frac{(\mathbf{e}_\alpha \cdot \mathbf{V})^2 - (c_s|\mathbf{V}|)^2}{2\gamma c_s^4}\right] + O(\mathbf{V}^3) \quad (17)$$

and f_α^{neq} is defined as

$$f_\alpha^{neq}(\mathbf{r}, t) = -\tau_p\left[f_\alpha^{eq}(\mathbf{r}, t) - f_\alpha^{eq}(\mathbf{r} - \mathbf{e}_\alpha\delta t, t - \delta t)\right] \quad (18)$$

where γ is the preconditioning parameter and τ_p is the preconditioned relaxation factor. The preconditioning parameter γ relates τ_p to the standard relaxation factor τ_s in the following way:

$$\gamma = \frac{\tau_s - 0.5}{\tau_p - 0.5} \quad (19)$$

Letting the preconditioning matrix $\mathbf{P} = diag\left(1, \gamma^{-1}, \gamma^{-1}, \gamma^{-1}\right)$, then through the Chapman-Enskog expansion, the flux tensor in Eq. (1) for the preconditioned NS equations can be shown to be:

$$\mathbf{F}_p = \mathbf{P} \times \begin{bmatrix} \rho u & \rho v & \rho w \\ \rho u^2 + p^* - \mu\left(2\frac{\partial u}{\partial x}\right) & \rho uv - \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) & \rho uw - \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \\ \rho vu - \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) & \rho v^2 + p^* - \mu\left(2\frac{\partial v}{\partial y}\right) & \rho vw - \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \\ \rho wu - \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) & \rho wv - \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) & \rho w^2 + p^* - \mu\left(2\frac{\partial w}{\partial z}\right) \end{bmatrix} \quad (20)$$

where

$$p^* = \gamma c_s^2 \rho \quad (21)$$

and

$$\mu = \gamma \rho c_s^2 (\tau_p - 0.5)\delta t \quad (22)$$

For steady flows utilising Eq. (20) in Eq. (1) is equivalent to utilising Eq. (2) in Eq. (1) but with a different equation of state. To demonstrate the effect of the preconditioning parameter on the rate of convergence, the partial differential equation form of the preconditioned NS equations is employed:

$$\frac{\partial \mathbf{Q}}{\partial t} + \mathbf{PA}\frac{\partial \mathbf{Q}}{\partial x} + \mathbf{PB}\frac{\partial \mathbf{Q}}{\partial y} + \mathbf{PC}\frac{\partial \mathbf{Q}}{\partial z} = 0 \quad (23)$$

where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ are the Jacobians of the flux vectors given by:

$$\mathbf{A} = \frac{\partial \mathbf{F}_{p,x}}{\partial \mathbf{Q}} \quad \mathbf{B} = \frac{\partial \mathbf{F}_{p,y}}{\partial \mathbf{Q}} \quad \mathbf{C} = \frac{\partial \mathbf{F}_{p,z}}{\partial \mathbf{Q}} \quad (24)$$

and $\mathbf{F}_{p,x}$, $\mathbf{F}_{p,y}$, $\mathbf{F}_{p,z}$ are the components of the flux tensor in the x, y and z directions respectively. Considering only the inviscid terms, these are given by:

$$\mathbf{F}_{p,x} = \begin{bmatrix} \rho u \\ \rho u^2 + p^* \\ \rho vu \\ \rho wu \end{bmatrix} \quad \mathbf{F}_{p,y} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p^* \\ \rho wv \end{bmatrix} \quad \mathbf{F}_{p,z} = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p^* \end{bmatrix} \quad (25)$$

The preconditioned convection matrix $\mathbf{PA}$ is now written as:

$$\mathbf{PA} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\gamma} & 0 & 0 \\ 0 & 0 & \frac{1}{\gamma} & 0 \\ 0 & 0 & 0 & \frac{1}{\gamma} \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ \gamma c_s^2 - u^2 & 2u & 0 & 0 \\ -vu & v & u & 0 \\ -wu & w & 0 & u \end{bmatrix} \quad (26)$$

To calculate the eigenvalues of $\mathbf{PA}$, the determinant of $(\mathbf{PA} - \lambda\mathbf{I})$ is first calculated:

$$det(\mathbf{PA} - \lambda\mathbf{I}) = \frac{(u - \lambda)^2}{\gamma^3}\left[\lambda^2 - 2u\lambda + u^2 - \gamma c_s^2\right] \quad (27)$$

The eigenvalues λ are found by letting Eq. (27) equal 0 giving:

$$\lambda(\mathbf{PA}) = \frac{1}{\gamma}(u, u, u \pm \gamma c_s) \quad (28)$$

The aim of preconditioning is to reduce the stiffness of the system by scaling the eigenvalues of $\mathbf{PA}$ appropriately. The reduction in stiffness can be measured by the condition number (CN) given as:

$$CN(\mathbf{PA}) = \frac{|max[\lambda(\mathbf{PA})]|}{|min[\lambda(\mathbf{PA})]|} = 1 + \frac{\gamma c_s}{u} = 1 + \frac{\gamma}{Ma} \quad (29)$$

where Ma is the dimensionless Mach number. A CN close to 1 indicates a well balanced system with no stiffness. From Eq. (29) it can be seen that to achieve a CN close to 1, it is required that $Ma \to \infty$ or $\gamma \to 0$. As Ma is restricted to values < 0.4 to satisfy incompressiblity requirements, reducing γ can be used to reduce the stiffness of the system.

2.1.5. Time integration

One advantage of the LBFS is that it allows a decoupling of the lattice streaming time step δt and the time integration of the NS equations. This enables the use of many different methods for the time integration of the NS equations in the LBFS. A fourth order Runge-Kutta (RK4) time-stepping scheme is adopted in this work as it has been shown to be very efficient and robust for a cell-centred finite-volume discretisation [44,45]. While acknowledging that steady flow problems do not require high levels of time accu-

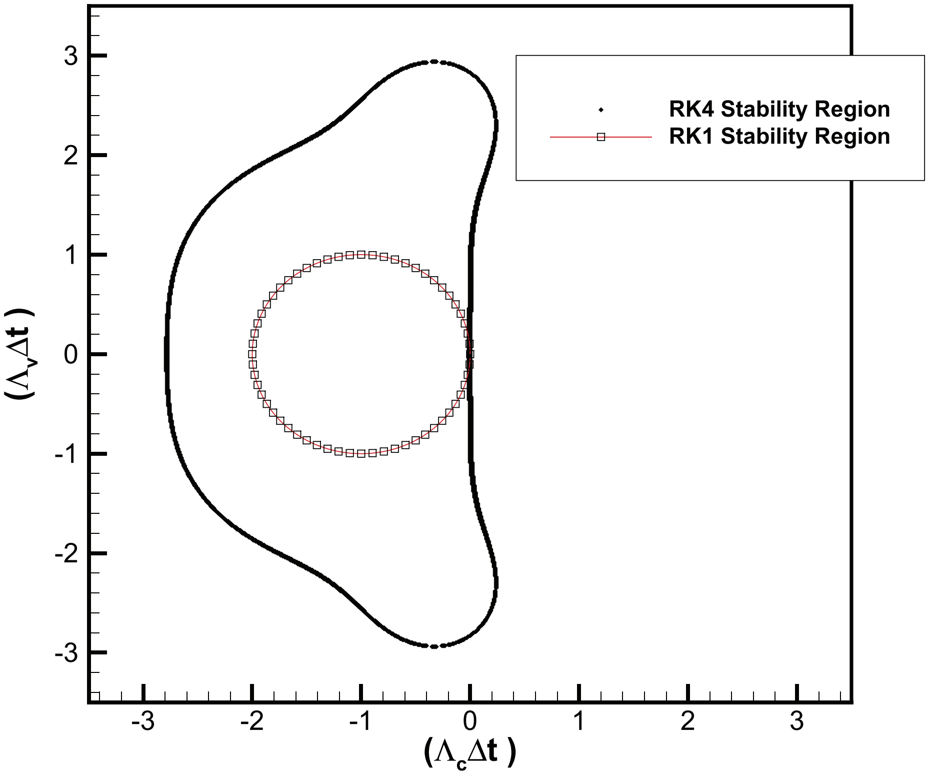


Fig. 2. Stability limits for first and fourth order Runge Kutte time integration methods for both viscous Λ_v and inviscid Λ_v eigenvalues.

racy, the RK4 scheme is far more robust than RK1/Forward Euler time-stepping schemes with regards to the stability limits of the magnitude of the viscous eigenvalues (see Fig. 2). As will be shown later in Section 3.3, applying preconditioning increases the viscous eigenvalue of the flow and necessitates the adoption of the RK4 scheme. Rewriting Eq. (3) in terms of a residual gives:

$$\frac{d}{dt}\mathbf{Q}_i + \mathbf{R}(\mathbf{Q}_i) = 0 \tag{30}$$

where $\mathbf{R}(\mathbf{Q}_i)$ is the residual function defined by:

$$\mathbf{R}(\mathbf{Q}_i) = \frac{1}{V_i}\mathcal{L}\mathbf{Q}_i \tag{31}$$

and $\mathcal{L}$ is the flux spatial discretisation operator defined as follows:

$$\mathcal{L}\mathbf{Q}_i = \sum_{n=1}^{N_{faces}} \mathbf{F}_{i,n} \cdot \mathbf{n}_{i,n} S_{i,n} \tag{32}$$

A RK4 integration scheme can be typically described as:

$$\begin{aligned}
\mathbf{Q}_i^{(0)} &= \mathbf{Q}_i^{(t)} \\
\mathbf{Q}_i^{(1)} &= \mathbf{Q}_i^{(0)} - \frac{\Delta t}{2}\mathbf{R}_i^{(0)} \\
\mathbf{Q}_i^{(2)} &= \mathbf{Q}_i^{(0)} - \frac{\Delta t}{2}\mathbf{R}_i^{(1)} \\
\mathbf{Q}_i^{(3)} &= \mathbf{Q}_i^{(0)} - \Delta t\mathbf{R}_i^{(2)} \\
\mathbf{Q}_i^{(t+1)} &= \mathbf{Q}_i^{(0)} - \frac{\Delta t}{6}(\mathbf{R}_i^{(0)} + 2\mathbf{R}_i^{(1)} + 2\mathbf{R}_i^{(2)} + \mathbf{R}_i^{(3)})
\end{aligned} \tag{33}$$

where $\mathbf{R}_i^{(q)} = \mathbf{R}(\mathbf{Q}_i^{(q)})$ for $q = 0$ to 3, Δt is the time-integration time step and t denotes the t^{th} time step in the simulation.

To ensure the stability of the time-stepping scheme, Δt must fulfil the CFL condition [46]. A CFL condition for an explicit scheme where the allowable CFL number $\sigma = 1$ means that Δt must be equal to or smaller than the time required to transport information across the stencil of the spatial discretisation scheme. Using higher order integration methods allows higher values of σ to be used. For a RK4 scheme $\sigma = 2.78$. There are many ways to estimate the maximum allowable time step on unstructured grids with the following approach adopted in this work [47]:

$$\Delta t_i = \sigma \frac{V_i}{(\Lambda_c + C\Lambda_v)_i} \tag{34}$$

where Λ_c and Λ_v represent estimates of the spectral radii of the convective and viscous Jacobians respectively for cell *i*. *C* is an arbitrary constant which is normally chosen to be $C = 4$. In the case of a cell-centred finite-volume scheme, the spectral radii are calculated as:

$$\begin{aligned}
\Lambda_c &= \Lambda_{c,x} + \Lambda_{c,y} + \Lambda_{c,z} \\
\Lambda_v &= \Lambda_{v,x} + \Lambda_{v,y} + \Lambda_{v,z}
\end{aligned} \tag{35}$$

where the Cartesian components of the spectral radii are given as:

$$\begin{aligned}
\Lambda_{c,x} &= \frac{1}{\gamma}(|u| + \gamma c_s)S_{i,x} \\
\Lambda_{c,y} &= \frac{1}{\gamma}(|v| + \gamma c_s)S_{i,y} \\
\Lambda_{c,z} &= \frac{1}{\gamma}(|w| + \gamma c_s)S_{i,z} \\
\Lambda_{v,x} &= \frac{2\mu}{\rho}\frac{(S_{i,x})^2}{\gamma V_i} \\
\Lambda_{v,y} &= \frac{2\mu}{\rho}\frac{(S_{i,y})^2}{\gamma V_i} \\
\Lambda_{v,z} &= \frac{2\mu}{\rho}\frac{(S_{i,z})^2}{\gamma V_i}
\end{aligned} \tag{36}$$

Table 2
A detailed summary of the PLBFS solution procedure.

	PLBFS Algorithm
1.	Calculate the local time step for each cell using Eq. (34) with $\mathbf{Q}_i^{(t-1)}$ as input.
2.	Calculate the gradients at the cell centres using Eq. (A.1).
3.	At the cell interface initialise a D3Q15 local latttice Boltzmann solution with δx chosen so that all lattice grid nodes lie within the two cells adjoining the shared surface.
4.	Find $\rho(\mathbf{r}-\mathbf{e}_\alpha\delta t, t-\delta t)$ and $\mathbf{V}(\mathbf{r}-\mathbf{e}_\alpha\delta t, t-\delta t)$ at the lattice grid nodes by interpolation using Eq. (13) and Eq. (14).
5.	Calculate the pre-streaming equilibrium density distribution function $f_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta t, t-\delta t)$ using Eq. (17).
6.	Calculate $\rho(\mathbf{r}, t)$ and $\mathbf{V}(\mathbf{r}, t)$, the post-streaming macroscopic variables at the cell interface, from Eq. (15).
7.	Use these values to calculate $f_\alpha^{eq}(\mathbf{r}, t)$ using Eq. (17).
8.	$f_\alpha^{neq}(\mathbf{r}, t)$ can be calculated from $f_\alpha^{eq}(\mathbf{r}, t)$ and $f_\alpha^{eq}(\mathbf{r}-\mathbf{e}_\alpha\delta t, t-\delta t)$ using Eq. (18).
9.	The inviscid and viscous fluxes can then be calculated from $f_\alpha^{eq}(\mathbf{r}, t)$ and $f_\alpha^{neq}(\mathbf{r}, t)$ using Eq. (16).
10.	The solution can then be advanced in time using Eq. (33).

where $S_{i,x}$, $S_{i,y}$ and $S_{i,z}$ represent projections of the cell onto *y-z, x-z* and *x-y* planes respectively, i.e.:

$$S_{i,x} = \frac{1}{2}\sum_{n=1}^{N_{faces}} |S_x|_{i,n}$$
$$S_{i,y} = \frac{1}{2}\sum_{n=1}^{N_{faces}} |S_y|_{i,n}$$
$$S_{i,z} = \frac{1}{2}\sum_{n=1}^{N_{faces}} |S_z|_{i,n} \tag{37}$$

where S_x, S_y and S_z denote the x, y and z components of the face vector $\mathbf{S}_{i,n} = \mathbf{n}_{i,n}\cdot S_{i,n}$, $S_{i,n}$ is the surface area and $\mathbf{n}_{i,n}$ is the outward normal of face *n* of cell *i*. The time step for the whole domain is then given as:

$$\Delta t = min(\Delta t_i) \tag{38}$$

For unstructured meshes, where the ratio of the largest cell to the smallest cell is large, this can result in extremely slow convergence of the solution to steady-state. However the transient behaviour of the flow in steady problems is of no interest. As a result local time-stepping can be employed and each cell is progressed at its maximum allowable stable time step calculated by Eq. (34). This results in significant convergence acceleration. The computational procedure employed in the PLBFS is summarised in Table 2.

3. Impact of preconditioning on unstructured grids

3.1. Overview

One advantage of using an unstructured grid with the LBFS is that it allows local refinement of the mesh in areas of rapidly changing flow. In practice, this involves having a very fine mesh close to surfaces in the flow and a much coarser mesh further away. This is in contrast to the traditional LBM where a uniform mesh density is used throughout the computational domain. When Guo [34] applied preconditioning to the LBM, there was no need to consider the impact of preconditioning on larger convection dominated cells or relatively finer cells in the boundary layer that are viscous dominated. As a result the purpose of that work was to reduce the stiffness of the inviscid Jacobians. In this section it will be shown that the PLBFS can improve the accuracy of flux calculations in larger cells compared to the LBFS but can also have an adverse effect on local time-stepping in viscous dominated cells. An optimal choice of γ will consider both these factors while trying to reduce the *CN* of the system, and a strategy for making this choice is proposed at the end of this section.

3.2. Influence of preconditioning on lattice streaming distance

When choosing a lattice streaming distance δx, the preferred strategy is to maximise the size of the lattice while keeping all lattice nodes within the adjoining cells to the common cell interface. This has the effect of reducing the error in interpolating the macroscopic variables to the lattice nodes using Eq. (13) and Eq. (14). This is due to the fact that as δx increases, the distance between the lattice nodes and the cell centre, from which the macroscopic variables are interpolated, is reduced. This increases the accuracy of the interpolation as the gradient is calculated at the cell centre and the lattice nodes are closer to the cell centre.

However this leads to issues in relatively large cells that are prevalent in unstructured grids. From Eq. (22), as δx increases τ_p decreases. This has an adverse affect on stability as the LBM becomes unstable as τ_p tends to its minimum value 0.5 especially in the range $0.50 < \tau_p < 0.55$ [48]. The special case of inviscid flow i.e. when $\tau_p = 0.5$ is not subject to this limitation as Eq. (22) is satisfied for an arbitrary lattice streaming distance δx. Preconditioning is most advantageous in scenarios when the stability limit of τ_p limits the size of δx. Decreasing γ allows an increase in δx for a constant τ_p. The benefits are illustrated by Fig. 3, where decreasing γ allows a much larger lattice reconstruction without introducing instability by lowering the value of τ_p.

As result of this property, preconditioning allows the use of coarser unstructured grids than the unpreconditioned LBFS with increased accuracy. This allows for either a decrease in runtimes for similar levels of accuracy or an increase in accuracy for the same runtime.

3.3. Influence of preconditioning on viscous dominated cells

Using preconditioning can have an adverse effect on time-stepping in viscous dominated cells, i.e. where the viscous spectral radii is larger than the convective spectral radii, or can cause convection dominated cells to become viscous dominated. To illustrate the impact of γ on the local time step of these relatively fine cells employed in viscous dominated regions, the spectral radii of a cell is calculated using Eq.s (34) to (37) and its assumed that $\mathbf{V} \to 0$ as the cell is assumed to neighbour a surface in the flow. For a uniform cell with edge length *l*, surface area *S* and volume *V* the convective and viscous spectral radii become:

$$\Lambda_c = \frac{1}{\gamma}3(0+\gamma c_s)S = 3c_sS$$
$$\Lambda_v = \frac{1}{\gamma V}\frac{2\mu}{\rho}\cdot 3S^2 = \frac{6\mu l}{\gamma\rho} \tag{39}$$

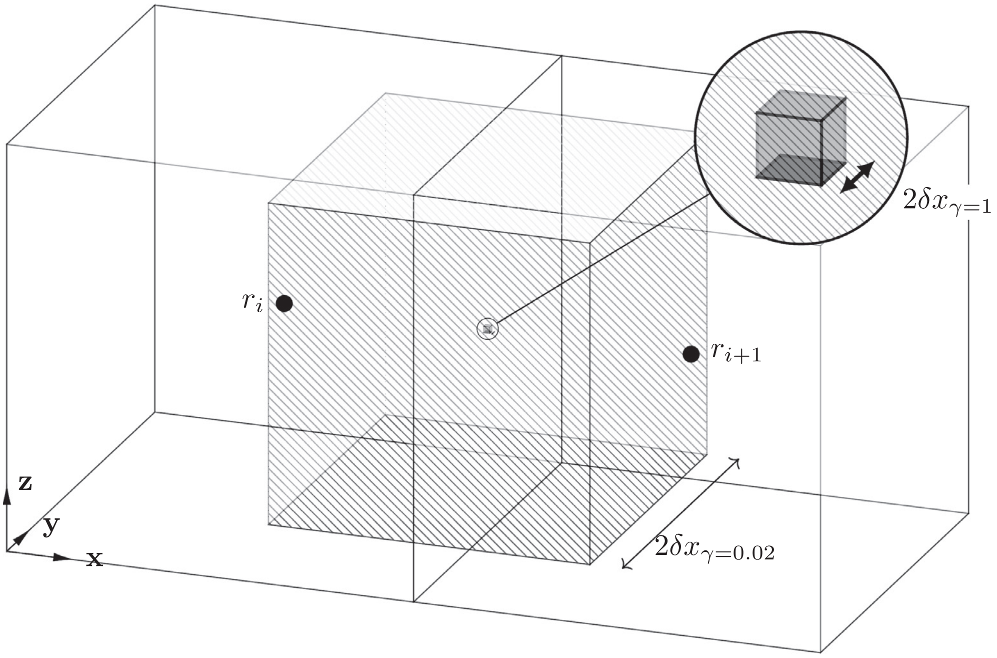


Fig. 3. Local reconstruction of the lattice Boltzmann lattice for $\gamma = 1$ and $\gamma = 0.02$ where $Re = 1000$, $Ma = 0.17$ and $\mu = 0.0001$.

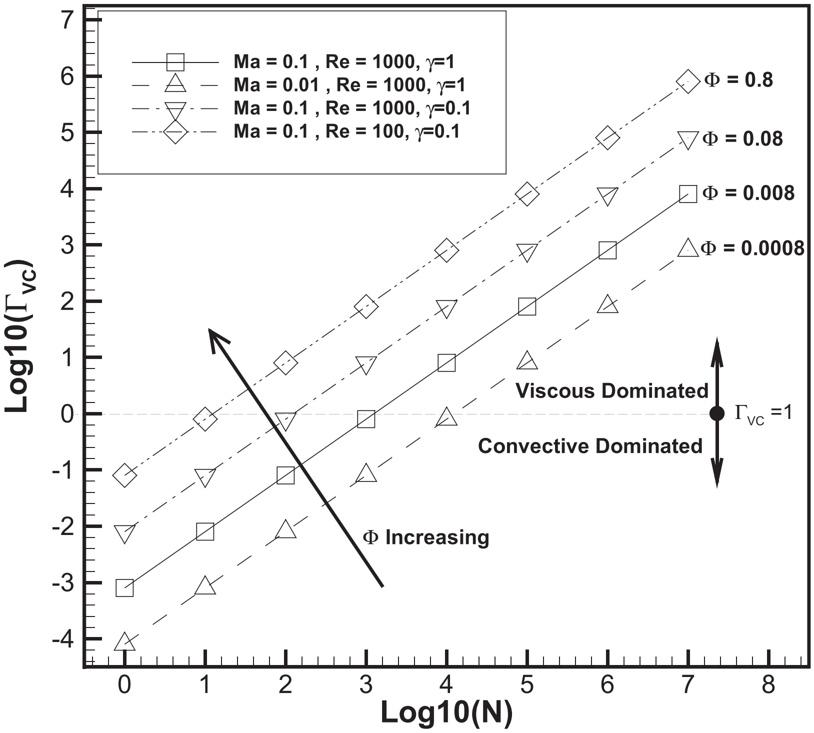


Fig. 4. Variation of Γ_{vc} with N for different values of Φ.

The ratio of the viscous spectral radius to the inviscid spectral radius, including the constant in Eq. (34), is given as:

$$\Gamma_{vc} = \frac{C\Lambda_v}{\Lambda_c} = 4\frac{2\mu}{\gamma c_s l \rho} \tag{40}$$

Re-writing in non-dimensional form and defining l relative to the reference length L gives:

$$\Gamma_{vc} = \frac{8MaN}{\gamma Re} = \Phi N \tag{41}$$

where Φ is a constant and $N = L/l$ is the number of uniform cells along the reference length L. Values of $\Gamma_{vc} > 1$ indicate that the flow in a cell is viscous dominated. If $\Gamma_{vc} > \; > 1$ this means that the time step in Eq. (34) is calculated based off the viscous spectral radius and is not optimised for the inviscid contributions. Fig. 4 illustrates how Φ and cell size influence if a cell is convection or viscous dominated. Cells becomes viscous dominated as the edge length of the cell becomes smaller and the viscous threshold of edge length, where the cell changes from convection domi-

Table 3
Strategy for choosing values for preconditioning on unstructured grids.

Preconditioning Parameter Selection
1. First attempt using the optimal precondition parameter provide by Izquierdo [37] where $\gamma = 2Ma$.
2. If no convergence acceleration is experienced, then for a given *Re* and mesh, investigate the histogram of cell sizes in the computational domain and identify the most common cell size.
3. Using this cell size calculate Γ_{vc} using Eq. (41) for uniform grids or Eq.s (35) to (37) for unstructured grids.
4. If $\Gamma_{vc} > 1$, then decrease *Ma*, increase γ or increase the size of the smallest cell in the mesh.
5. Repeat Step 3 until an acceleration in runtime is achieved or $\gamma = 1$

Table 4
Individual test case parameters for 3D lid-driven cavity flow.

Test Case No.	u_{lid}	Re	Ma	γ	CN	$\Gamma_{vc,\ max}$	δx_{max}	N_{cells}
1	0.1	100	0.173	1	6.77	0.622	0.60	90480
2	0.1	100	0.173	0.2	2.15	3.11	3.00	90480
3	0.01	100	0.017	1	58.73	0.06	0.06	90480
4	0.01	100	0.017	0.02	2.15	3.11	3.00	90480
5	0.1	1000	0.173	1	6.77	0.062	0.06	90480
6	0.1	1000	0.173	0.2	2.15	0.311	0.30	90480
7	0.01	1000	0.017	1	58.73	0.006	0.006	90480
8	0.01	1000	0.017	0.02	2.15	0.311	0.30	90480
9	0.1	100	0.173	1	6.77	0.854	0.6	226981
10	0.01	100	0.017	1	58.73	0.085	0.06	226981
11	0.1	1000	0.173	1	6.77	0.085	0.06	226981
12	0.01	1000	0.017	1	58.73	0.008	0.006	226981
13	0.01	1000	0.017	1	58.73	0.011	0.006	531411

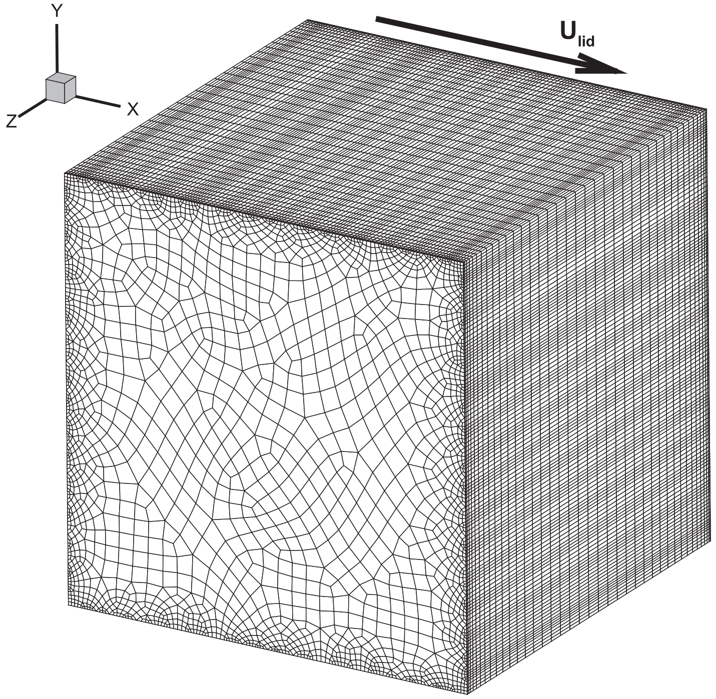


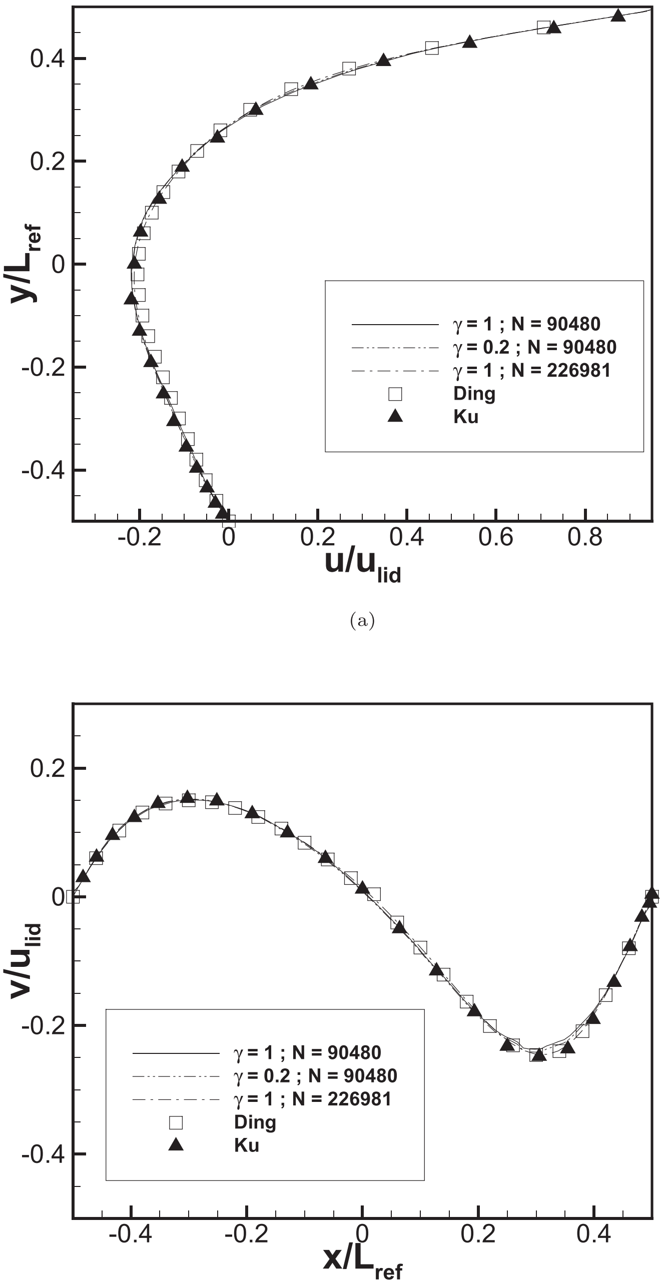
Fig. 5. 3D lid-driven cavity unstructured mesh (90480 cells) and problem setup.

nated to viscous dominated, is dependent on Φ. Smaller values of Φ allow finer cells to remain convection dominated. To reduce the value of Φ, the following choice of non-dimensional parameters is desired:

- Reduced values of *Ma*.
- Larger values of γ.
- Larger values of *Re*.

3.4. Preconditioning parameter selection strategy

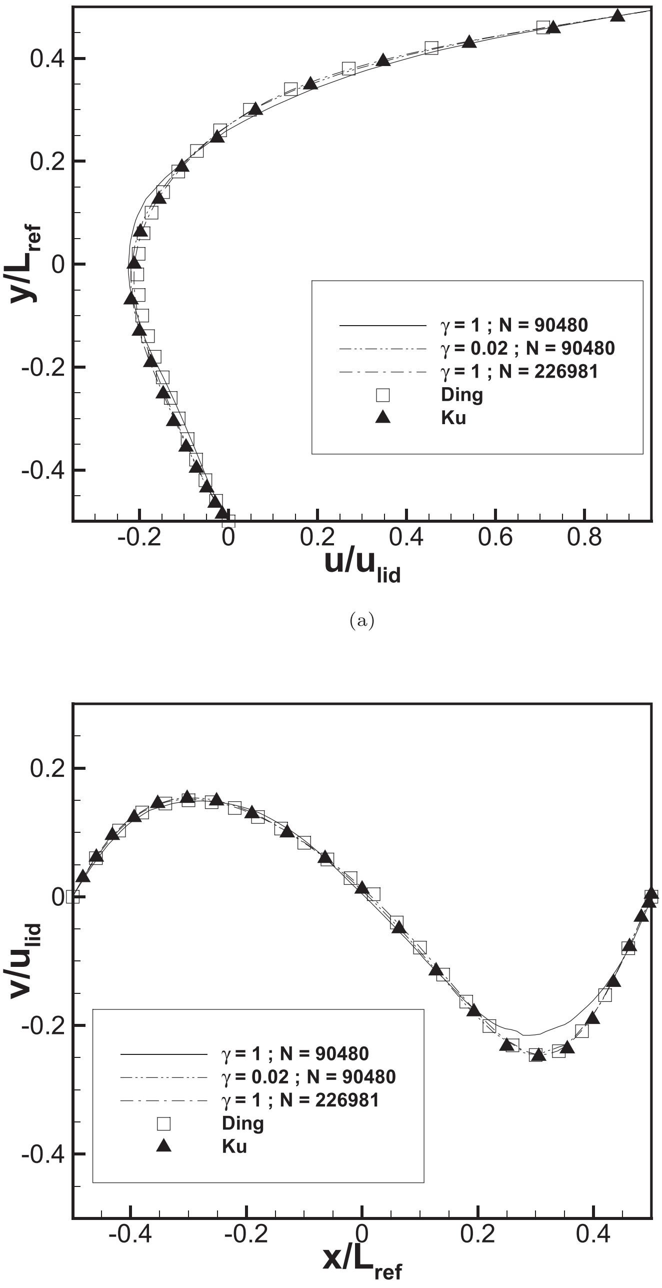
The purpose of preconditioning the NS equations is to reduce the stiffness of the inviscid Jacobians to accelerate convergence of steady flow calculations. However as shown in the previous sections, the use of preconditioning can have advantages and disadvantages when it comes to the use of unstructured grids. Preconditioning allows the use of larger cells in convection dominated flow



(a)

(b)

Fig. 6. 3D lid-driven cavity flow: normalised a) u and b) v velocity profiles along vertical and horizontal centrelines respectively in the $z = 0$ plane for $Re = 100$ and $u_{lid} = 0.1$.



(a)

(b)

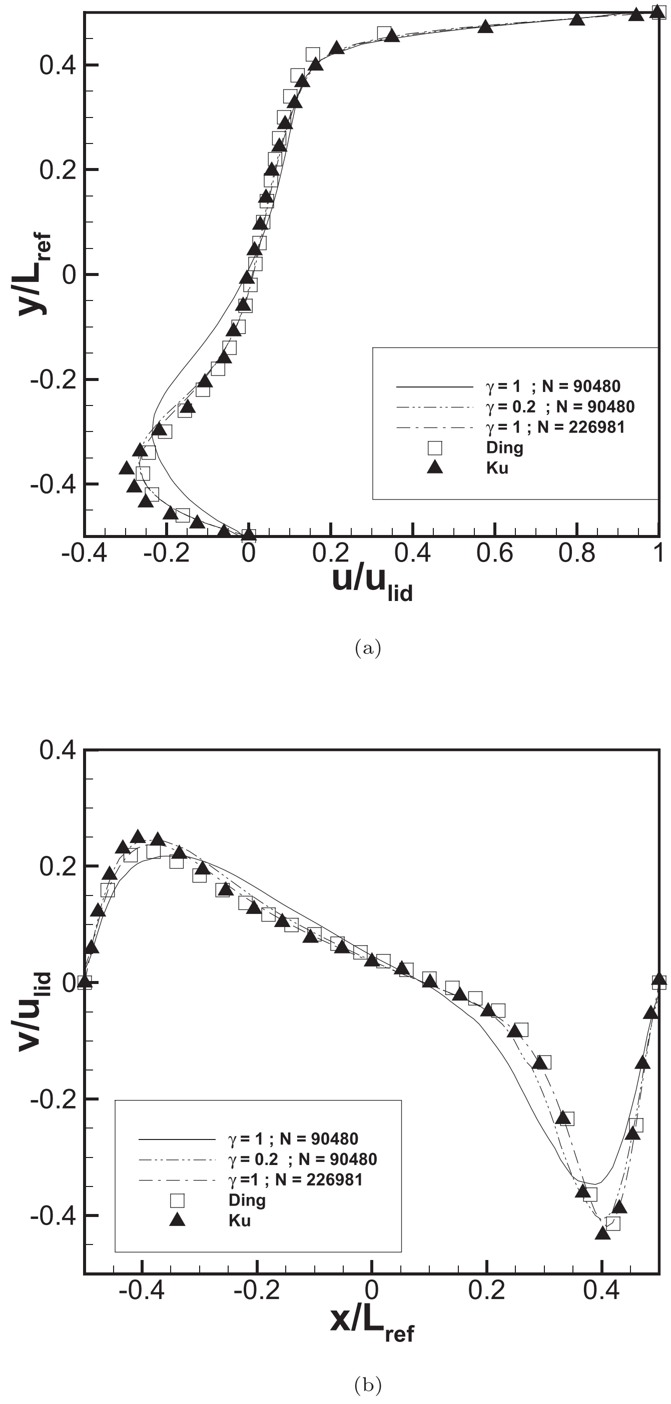
Fig. 7. 3D lid-driven cavity flow: normalised a) u and b) v velocity profiles along vertical and horizontal centrelines respectively in the $z = 0$ plane for $Re = 100$ and $u_{lid} = 0.01$.

which leads to reduced runtimes. However if the mesh contains a large number of viscous dominated cells, in particular cells in the boundary layer of a surface, this may have an adverse impact on the convergence rate. A good choice of preconditioning parameter would have $1 + \frac{\gamma}{Ma} \rightarrow 1$ whilst keeping $\Gamma_{vc} \leq 1$ for the vast majority of cells. From Eq. (41) and Fig. 4 it can be seen that smaller values of Ma and larger values of Re and γ allow smaller cells to be used while keeping $\Gamma_{vc} \leq 1$. Setting $\Gamma_{vc} = 1$ in Eq. (41) gives the following equation for setting γ:

$$\gamma > \frac{8MaN}{Re} \tag{42}$$

Note that Eq. (42) will only hold for every cell in uniform grids only. In grids where there is a large variation in cell size, there will be cells that will have $\Gamma_{vc} > 1$. Therefore it is recommended to investigate a histogram of cell size of the computational domain and optimise γ for the most common cell sizes.

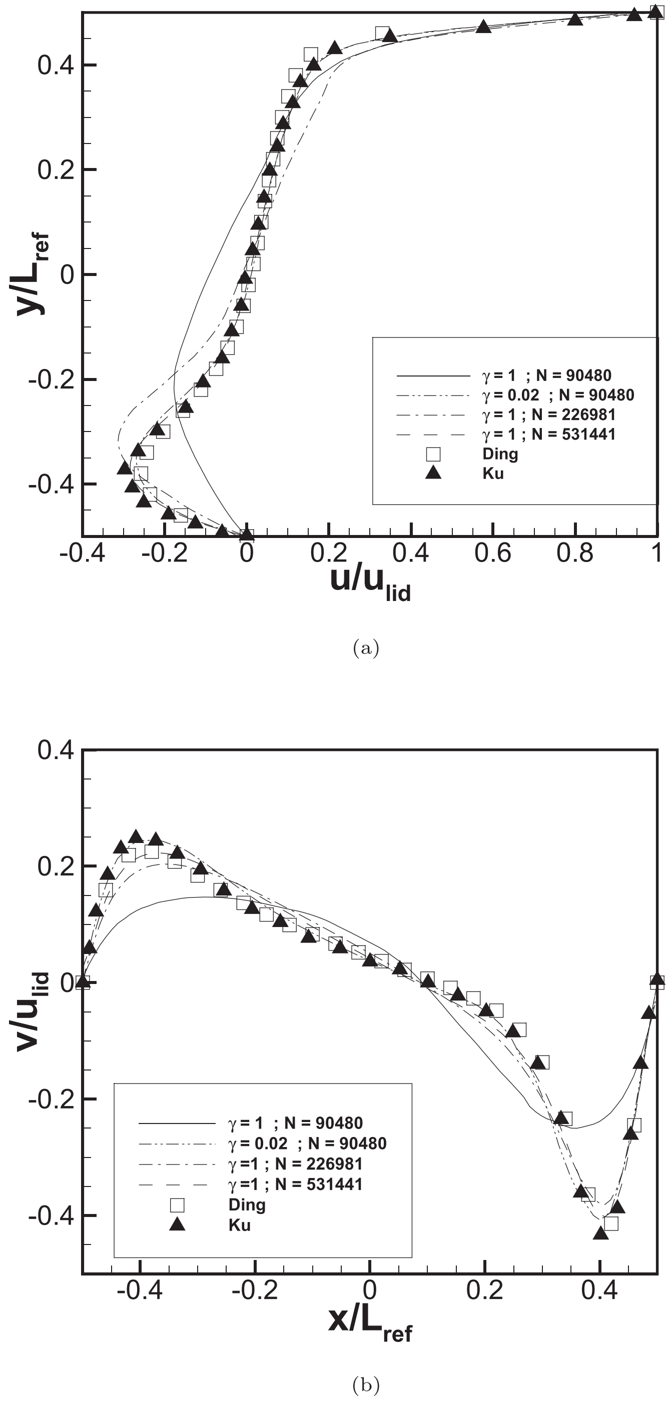
The impact of the choice of preconditioning parameter on stability should also be considered. Izquierdo suggested the optimal choice of $\gamma = 2Ma$ [37] as this value maximises speed up while offering robust stability. However it was provided with the caveat that the optimal value increases for lower Reynolds numbers. A strategy for choosing a value of γ is given in Table 3. It is preferable to decrease Ma rather than have $\Gamma_{vc} > 1$ as lower values of Ma reduces the compressibility error at no cost in runtime when local time-stepping is used. It should be noted that accurately calculating the real eigenvalues in a cell is quite difficult and that



(a)

(b)

Fig. 8. 3D lid-driven cavity flow: normalised a) u and b) v velocity profiles along vertical and horizontal centrelines respectively in the $z = 0$ plane for $Re = 1000$ and $u_{lid} = 0.1$.



(a)

(b)

Fig. 9. 3D lid-driven cavity flow: normalised a) u and b) v velocity profiles along vertical and horizontal centrelines respectively in the $z = 0$ plane for $Re = 1000$ and $u_{lid} = 0.01$.

Eq. (41) is based on an estimate and that $\Gamma_{vc} = 1$ may not be the exact threshold for cells becoming viscous dominated.

4. Numerical results and discussion

4.1. Overview

To demonstrate the capability of the PLBFS, two flow problems are considered: 3D lid-driven cavity flow and 3D flow over a circular cylinder. For both flow problems unstructured hexahedral meshes were used to demonstrate that the PLBFS can be effectively employed with an unstructured mesh topology. Flow problems were solved using a variety of Reynolds and Mach numbers and the impact of various values of the preconditioning parameter γ are investigated to show the impact of preconditioning on accuracy and convergence rates. All meshes were created with ANSYS ICEM meshing software (ANSYS Inc., Canonsburg, PA, USA).

The root mean square (RMS) of the ρu-momentum residual was used to monitor convergence to steady-state and is defined

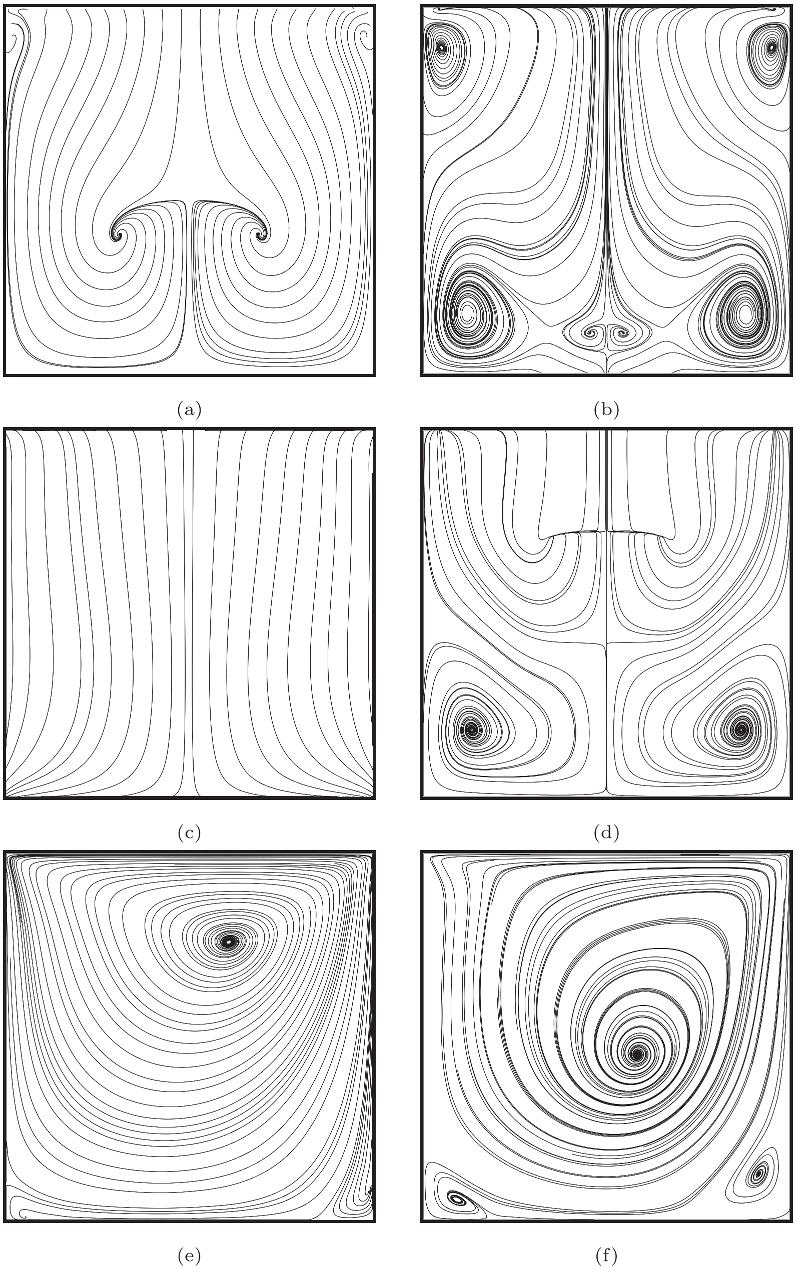


Fig. 10. 3D lid-driven cavity flow: predicted streamlines in the $x = 0$ plane for a) $Re = 100$ and b) $Re = 1000$, in the $y = 0$ plane for c) $Re = 100$ and d) $Re = 1000$ and in the $z = 0$ plane for a) $Re = 100$ and b) $Re = 1000$.

as follows:

$$R_{RMS}\left(\mathbf{Q}^{Iteration,t}\right) = \sqrt{\frac{\sum_{i=1}^{No\ of\ Cells}\mathbf{R}\left(\mathbf{Q}_{i,t}\right)^2}{No\ of\ Cells}} \tag{43}$$

where t is the iteration index and $\mathbf{R}(\mathbf{Q}_i)$ is the ρu residual calcualated for cell i using Eq. (31). The calculated RMS is normalised relative to the initial RMS value calculated at the end of the first iteration. i.e

$$R_{RMS,Norm} = \frac{R_{RMS}\left(\mathbf{Q}^{Iteration,t}\right)}{R_{RMS}\left(\mathbf{Q}^{Iteration,1}\right)} \tag{44}$$

The convergence criterion employed is a five order magnitude reduction in the normalised residual, i.e.

$$R_{RMS,Norm} < 1 * 10^{-5} \tag{45}$$

4.2. 3D Lid-driven cavity flow

Shear-driven flow in a square/cube cavity is a standard test case for validating predictions of incompressible viscous flow. It has historically been used as a benchmark flow problem to investigate the accuracy and the performance of CFD codes [49]. Lid-driven cavity flow is steady for a Reynolds number less than 10000, with a moving lid developing the flow. The steady-state predictions by the PLBFS are compared to existing numerical results produced by Ku et al. [50] and Ding et al. [51]. The set up of the lid-driven cavity problem is illustrated in Fig. 5. The top boundary moves with a velocity u_{lid}. This is implemented using a Dirichlet boundary condition specifying u_{lid}. A no-slip boundary condition is implemented at the remaining five boundaries. For density, a Neumann boundary condition, specifying zero change in density across the boundary, is implemented at all boundaries. The initial conditions are set as $\mathbf{V} = 0$ and $\rho = 1$. Multiple test cases were run for this flow problem with different combinations of Reynolds number, Mach number, preconditioning parameter γ and mesh density. The test case parameters are summarised in Table 4 and were chosen as per Table 3.

Test Cases 1-8 were run using the mesh shown in Fig. 5 consisting of 90480 hexahedral cells. Test Cases 9-13, performed without preconditioning, were run on finer meshes consisting of 226981 and 531411 cells. The origin of the coordinate system is located at the centroid of the cube and the edges on the cube have a reference length $L_{ref} = 10$. Predicted velocity profiles on vertical and horizontal centrelines respectively in the $z = 0$ plane for Test Cases 1-13 are shown in Figs. 6, 7, 8, 9. The results are compared to the predictions of Ku et al. [50] and Ding et al. [51]. Overall there is good agreement with the literature by the preconditioned test cases and the test cases performed on finer meshes. To show the flow patterns of the 3D lid-driven cavity flow, 2D streamlines are projected onto three centroidal planes at $x = 0$, $y = 0$ and $z = 0$. The flow patterns are shown for Test Cases 4 and 8 where $Re = 100$ and $Re = 1000$ respectively. These results are shown in Fig. 10 and show the development of stronger secondary vortices and a stronger 3D impact as the Reynolds number increases. This is in agreement with the predictions of previous studies by Ku et al. [50], Ding et al. [51] and Wang et al. [14]. Convergence histories are plotted in Fig. 11 and Fig. 12.

Using an unstructured hexahedral mesh produces excellent results that agree with predictions by other researchers in the literature. The previous study using a LBFS by Wang et al. [14] uses a $81X81X81$ (531441 cells) non-uniform structured mesh and it's predictions have excellent agreement with the studies of Ku et al. [50] and Ding et al. [51]. Using the PLBFS on a fully unstructured grid allows similar levels of accuracy to be attained while using a mesh with only 90480 cells. The results also confirm the theoretical analysis in Section 3.2. This analysis suggests that preconditioning should enable the use of large δx on lattices in relatively larger cells. This should reduce the interpolation error when applying Equation. Inspecting Table 4 shows that preconditioning allows the use of larger δx in the bigger cells in the mesh. Inspecting Figs. 6 to 9 show that the preconditioned case with the larger δx is more accurate than the unpreconditioned cases with smaller δx. Figs. 6 to 9 also show that increasing the mesh density results in more accurate comparisons with the benchmark solutions. There is also significant convergence acceleration with the use of preconditioning at lower Ma numbers. The reduction in iterations required to reach convergence compared to the non-preconditioned case is 9.72x at $Re = 1000$ and 5.62x at $Re = 100$ for $Ma = 0.017$. At the higher Mach number of $Ma = 0.17$, the decrease in CN is offset by disproportionately larger increase in $\Gamma_{vc,\,max}$ (see Table 4). This accounts for the lack of significant acceleration in convergence at the higher Mach number. An alternative to preconditioning to improve accuracy is to increase the mesh density. Figs. 6 to 9 show that the preconditioned test cases produce results that are of similar accuracy to those of the unpreconditioned test cases on finer meshes. This is also with the benefit of a reduction in the iterations required to reach convergence and a reduction in the flux calculations per iteration. As can be seen in Fig. 11 and Fig. 12, this effect is most prominent for high Re and low Ma. The preconditioned case of $Re = 1000$ and $Ma = 0.017$ uses 5.87x less cells and 20.67x less iterations than the unpreconditioned case on the finest mesh. In this case preconditioning enables a 121x reduction in computational effort while attaining similar levels of accuracy.

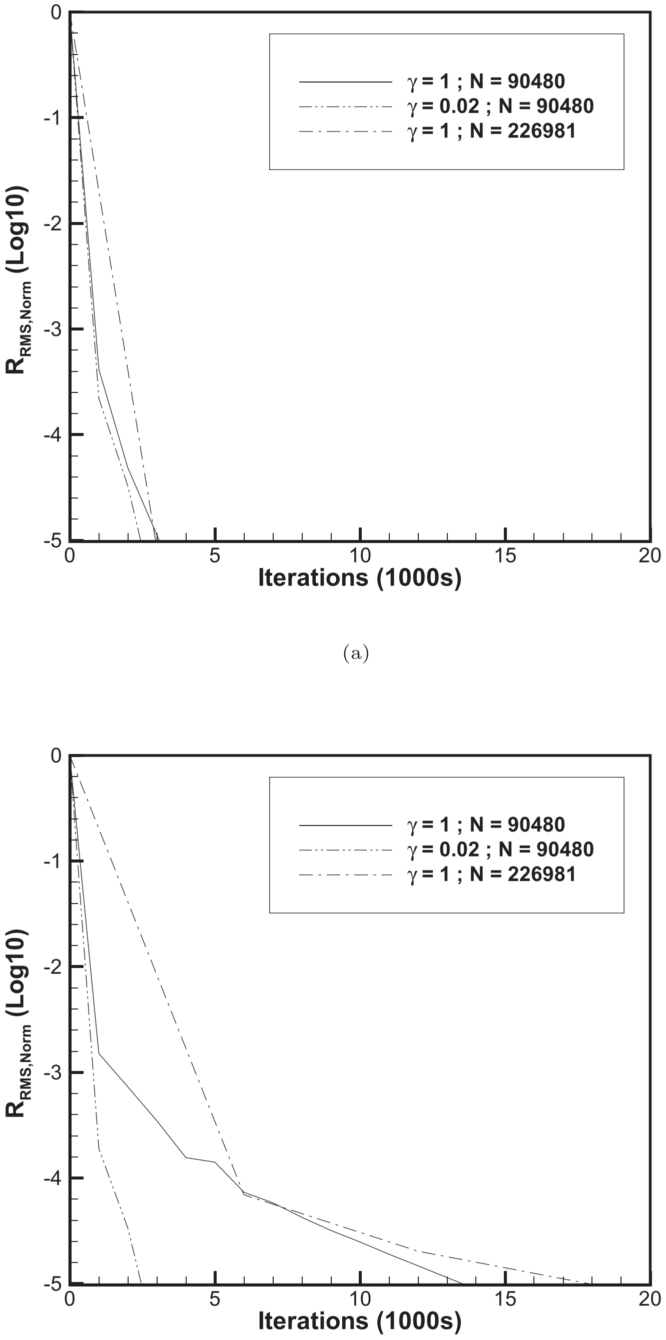
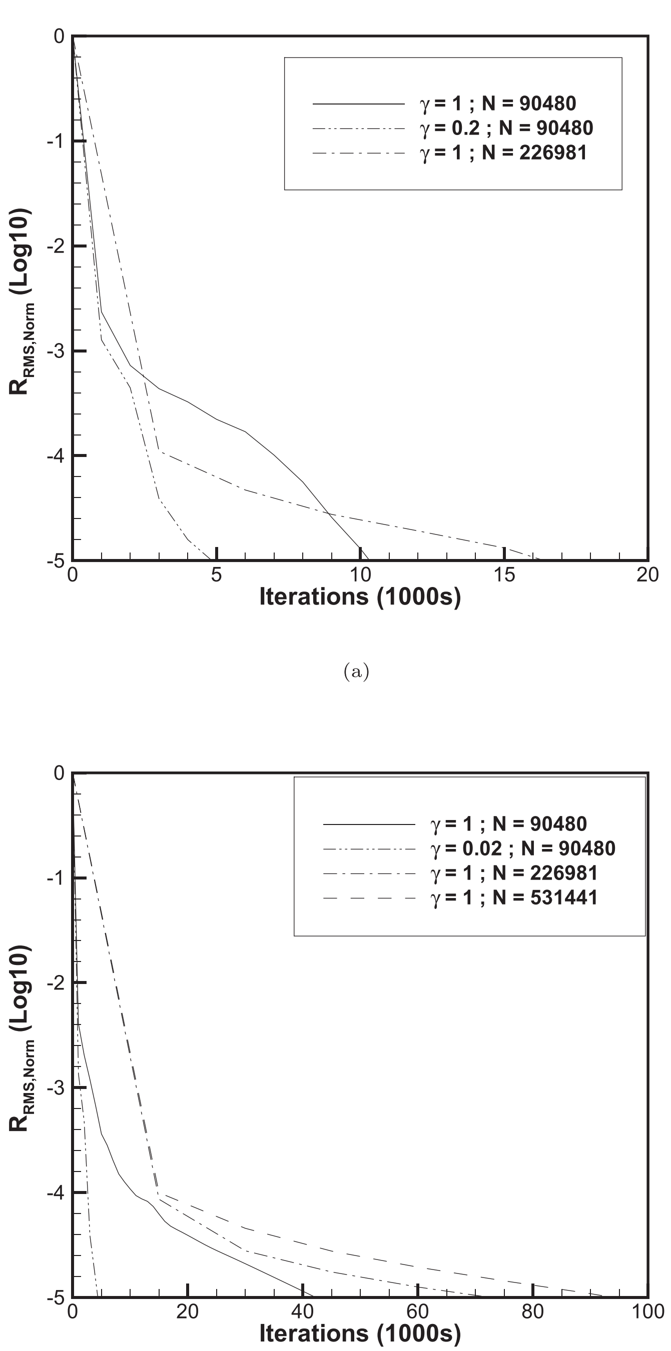


Fig. 11. 3D lid-driven cavity flow: convergence history for $Re = 100$ with a) $u_{lid} = 0.1$ and b) $u_{lid} = 0.01$.

4.3. 3D flow over a circular cylinder

As mentioned previously, one of the motivations behind using the LBFS is to avoid the staircase approximation of curved boundaries with the traditional LBM. To demonstrate the applicability of the LBFS to flow problems with curved boundaries, 3D flow over a circular cylinder was investigated. This is a steady flow for low Reynolds numbers and so the PLBFS can also be used to solve this flow problem. The problem is set up as in Fig. 13, with the cylinder immersed in a uniform freestream. The diameter of the cylinder is the reference length L_{ref} with a value of $L_{ref} = 1$



(a)

(b)

Fig. 12. 3D lid-driven cavity flow: convergence history for $Re = 1000$ with a) $u_{lid} = 0.1$ and b) $u_{lid} = 0.01$.

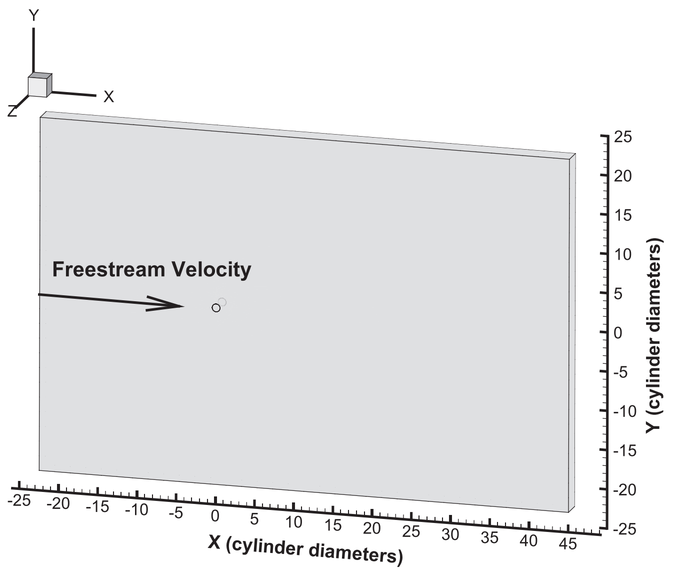


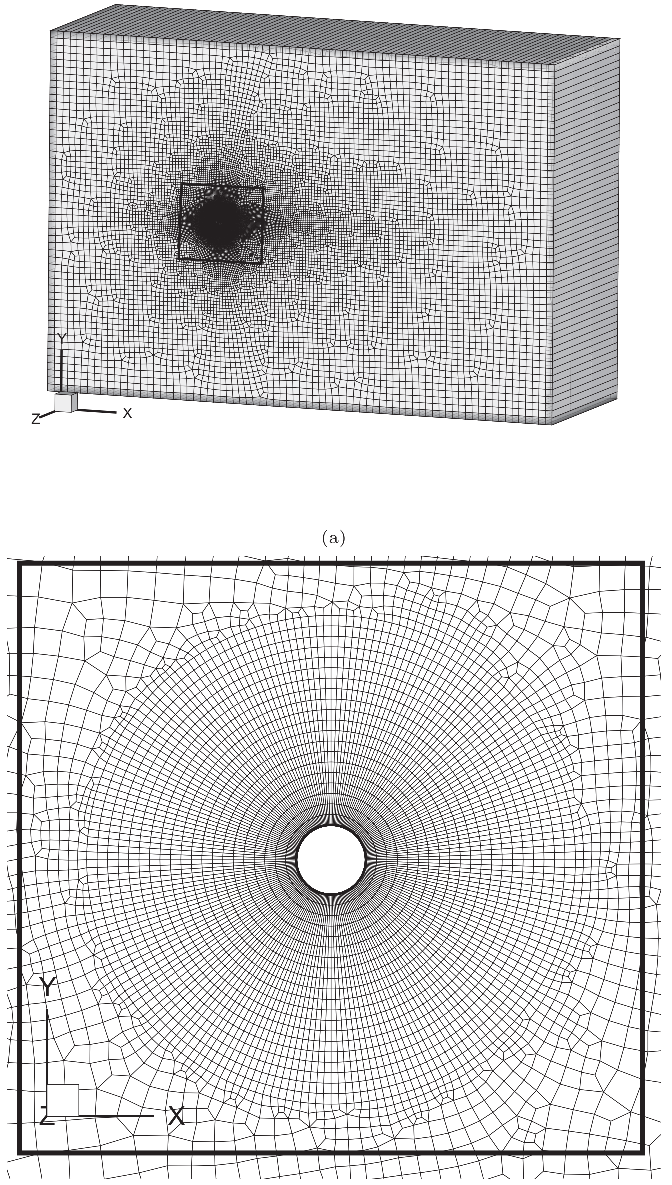
Fig. 13. 3D flow over a cylinder problem setup.

and spans the length of the z-domain. The boundary condition at the inlet boundary was set equal to the freestream density ρ_∞ and freestream velocity U_∞ using Dirichlet boundary conditions. At the outlet boundary condition the pressure was set equal to static pressure by a Dirichlet boundary condition and a zero change in velocity at the outlet was maintained by a Neumann boundary condition. A no-slip boundary condition was applied to the wall of the cylinder and the remaining boundaries all have a zero change in velocity and density which are maintained by Neumann boundary conditions. The initial conditions were set as $\mathbf{V} = 0$ and $\rho = 1$. The flow problem was run for different combinations of Reynolds number, Mach number and preconditioning parameter γ. The test cases parameters are summarised in Table 5. The values of γ were chosen for each test case as per the approach given by Table 3.

All test cases were run with a variety of meshes with densities of 7021, 13051, 18517, 23073 and 25877 cells. For each mesh, the cell height of the first cell adjacent to the cylinder wall was chosen as 0.05 and is one cell thick in the z-direction. The mesh containing 23073 cells is shown in Fig. 14. For $Re = 20$ and $Re = 40$, the flow is steady. The predicted streamlines and velocity contours Test Case 1 and 5 are shown in Fig. 15. The streamtraces show stationary recirculation regions on the lee side of the cylinder. The flow pattern shown is consistent with the existing studies in the literature; however more detailed measures are required to demonstrate the performance of the PLBFS. To do this, a variety of parameters can be employed including the drag coefficient of the cylinder, the recirculation length and separation angle. The drag coefficient C_d relates the force acting on a body by a fluid in the direction of the freestream velocity to the force of the freestream dynamic pressure acting on the frontal area of the body. This is calculated as follows:

Table 5
Individual test case parameters for 3D flow over a circular cylinder.

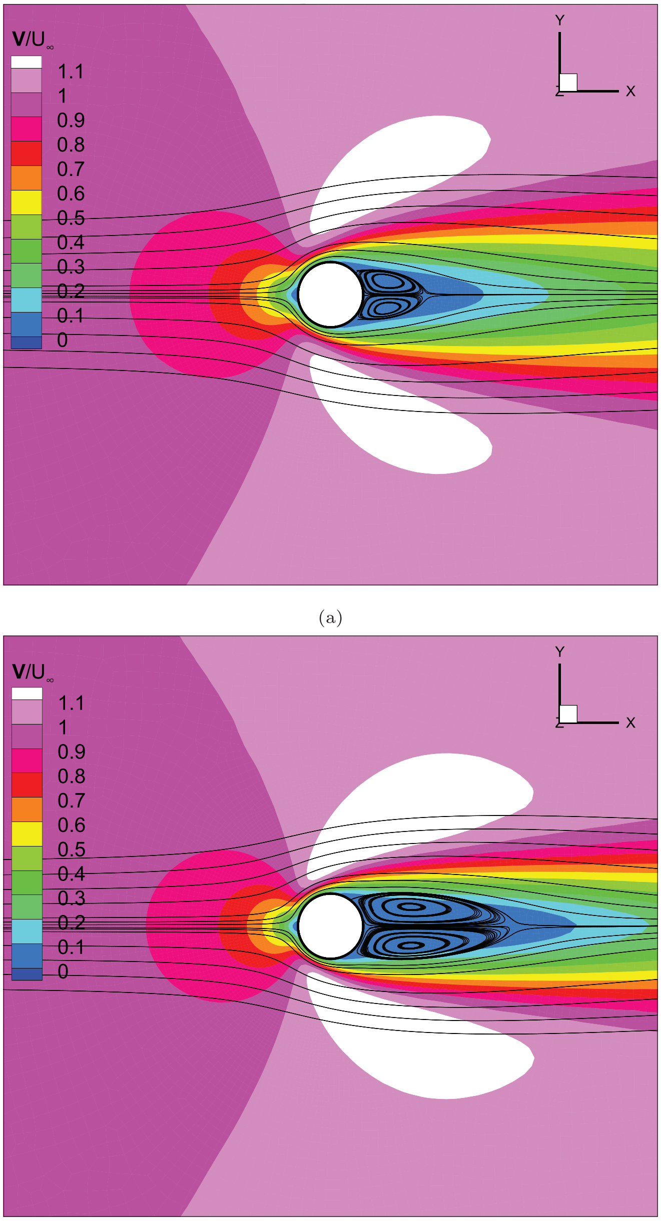
Test Case No.	U_∞	Re	Ma	γ	CN	$\Gamma_{vc,\,max}$	δx_{max}
1	0.1	20	0.17	1	6.77	4.41	0.173
2	0.1	20	0.17	0.2	2.15	22.05	0.866
3	0.01	20	0.017	1	58.73	0.44	0.017
4	0.01	20	0.017	0.05	3.88	8.82	0.346
5	0.1	40	0.17	1	6.77	2.20	0.086
6	0.1	40	0.17	0.2	2.15	11.04	0.433
7	0.01	40	0.017	1	58.73	0.22	0.008
8	0.01	40	0.017	0.05	3.88	4.41	0.173



(a)

(b)

Fig. 14. 3D flow over a circular cylinder: a) unstructured hexahedral mesh with 23073 cells and b) exploded view of the mesh close to cylinder surface.



(a)

(b)

Fig. 15. 3D flow over a circular cylinder: streamlines and velocity contour plots on the $z = 0$ plane for a) $Re = 20$ and b) $Re = 40$, $Ma = 0.17$, γ =1 and $N_{cells} = 23073$.

$$C_d = \frac{\gamma F_d}{0.5\rho_\infty U_\infty^2 A_f} \tag{46}$$

where A_f is the frontal area (the area projected onto a plane normal to the flow direction) of the body and F_d is the drag force which in this flow problem is calculated by integrating the pressure and viscous stresses acting in the x-direction over the surface of the cylinder, i.e.:

$$F_d = -\int_S \left(\left[p^* - \mu\left(2\frac{\partial u}{\partial x}\right) \quad -\mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \quad -\mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)\right] \cdot \mathbf{n}\right) dS \tag{47}$$

The pressure and viscous stresses in Eq. (47) were calculated using Eq. (16). The recirculation length L_s is defined as the distance between the trailing edge of the cylinder and the stagnation point in the wake i.e. where $\mathbf{V} = 0$ on the x-axis. The separation angle θ_s is the angle between the trailing edge and the point where the boundary layer separates from the cylinder. This point is indicated by the wall shear stress equalling zero.

To establish that the PLBFS is at least second order accurate, Test Case 5 has also been performed for a variety of mesh densities where the first cell height, number of divisions and maximum element length are scaled in the same manner from $0.2L_{ref}$ to $0.02L_{ref}$. The meshes used included:$0.2L_{ref}$, $0.1L_{ref}$, $0.0667L_{ref}$, $0.05L_{ref}$, $0.04L_{ref}$ and $0.02L_{ref}$. A further test case was performed at a scale of $0.01L_{ref}$ and this is used as the benchmark drag coefficient. The percentage error in the drag coefficient is calculated for each test case and is plotted in logarithmic form in Fig. 16. This shows that the PLBFS has an order of accuracy of approximately 2.3 for this test case and is at least second order accurate which is consistent with the work of Shu et al. [10].

The results from the PLBFS are compared to previous numerical studies (Shu et al. [10], Dennis and Chang [52], Shukla et al.

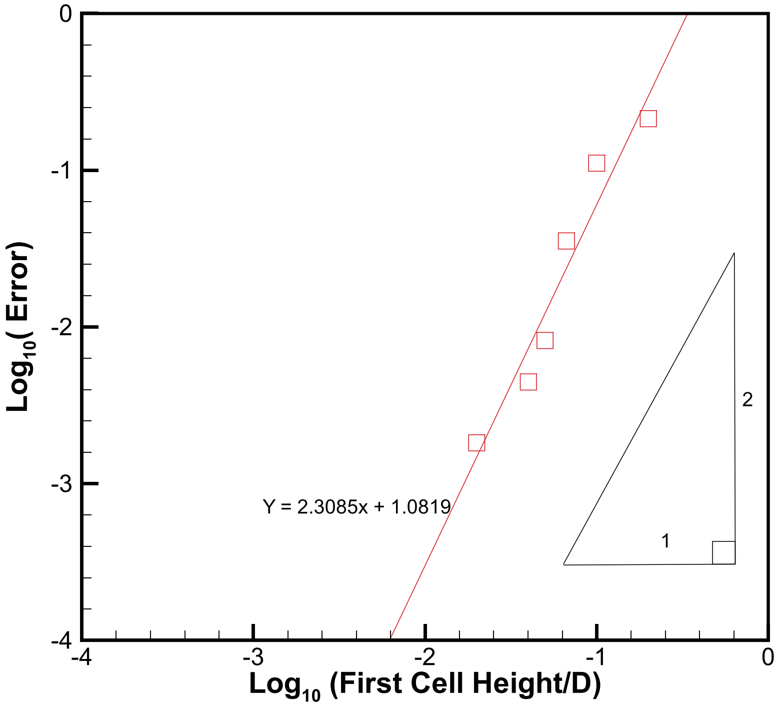


Fig. 16. Variation in percentage error of drag coefficient with mesh size. The percentage error is relative to a simulation with first cell height on the cylinder boundary equal $0.01D$. All simulations were run for $Re = 40$, $Ma = 0.17$ and γ =1.

Table 6
3D flow over a circular cylinder: comparison of predicted drag coefficient, recirculation length and separation angle with predictions in the literature for Re =20 and N_{cells} = 23073.

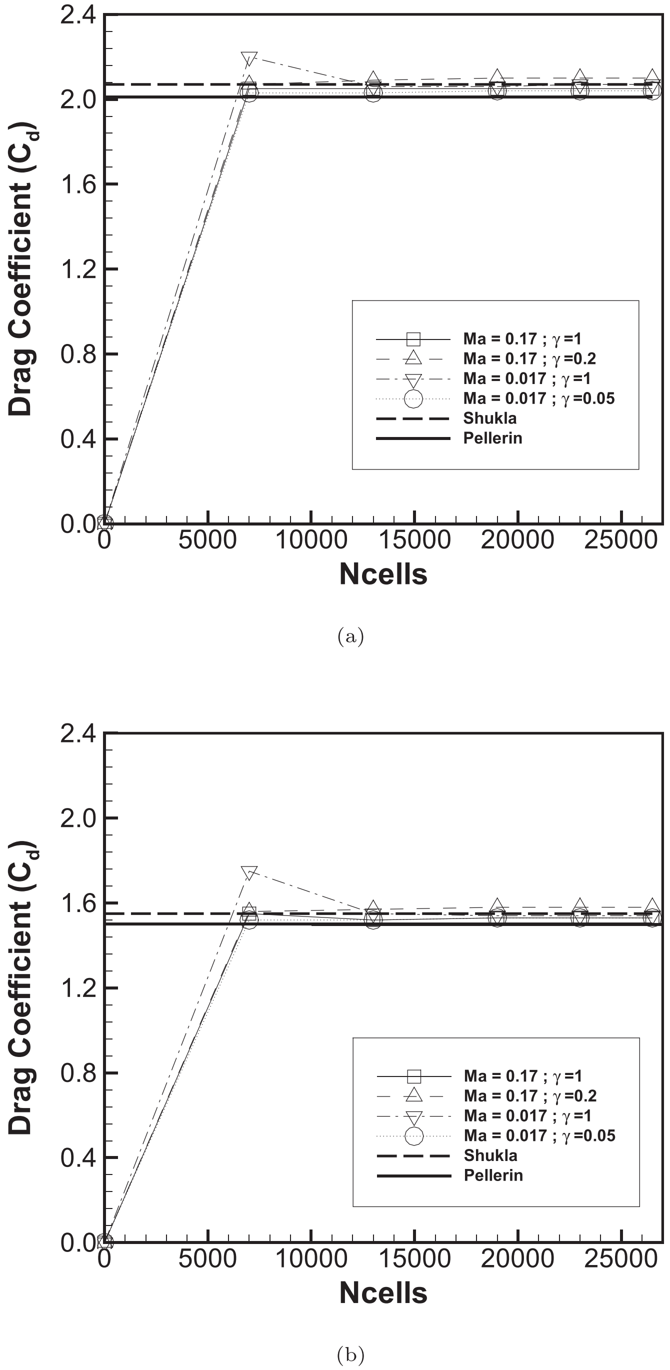
Reference	U_∞	γ	C_d	L_s/D	θ_s
Shu et al. [10]	0.1	-	2.06	0.94	42.94
Dennis and Chang [52]	-	-	2.05	0.94	43.70
Shukla et al. [53]	-	-	2.07	0.92	43.30
Pellerin et al.[11]	-	-	2.01	0.92	43.58
Test Case 1	0.1	1	2.05	0.91	42.97
Test Case 2	0.1	0.2	2.1	0.98	42.81
Test Case 3	0.01	1	2.07	0.86	40.86
Test Case 4	0.01	0.02	2.04	0.91	42.97

Table 7
3D flow over a circular cylinder: comparison of predicted drag coefficient, recirculation length and separation angle with predictions in the literature for Re = 40 and N_{cells} = 23073.

Reference	U_∞	γ	C_d	L_s/D	θ_s
Shu et al. [10]	-	-	1.53	2.24	52.69
Dennis and Chang [52]	-	-	1.52	2.35	53.80
Shukla et al. [53]	-	-	1.55	2.34	52.70
Pellerin et al. [11]	-	-	1.5	2.26	53.52
Test Case 5	0.1	1	1.53	2.23	53.13
Test Case 6	0.1	0.2	1.58	2.4	52.64
Test Case 7	0.01	1	1.54	2.00	49.26
Test Case 8	0.01	0.02	1.53	2.21	52.64

[53] and Pellerin et al. [11]) in Table 6 and Table 7. The variation of the various parameters with mesh density are also plotted and compared to the upperbound and lowerbound values from the literature in Figs. 17, 18, 19. These results show that for the vast majority of test cases, the parameters are converged or very near convergence on a mesh of 23073 cells. The main differentiator in accuracy are the parameters $\Gamma_{vc,\ max}$ and δx_{max}. The most accurate predictions are for those test cases which have reduced values of $\Gamma_{vc,\ max}$ while having δx_{max} greater than the first cell height in the boundary layer. These parameters are dependent on the choice of Ma and γ and a balance has to be struck between both to ensure maximum accuracy.

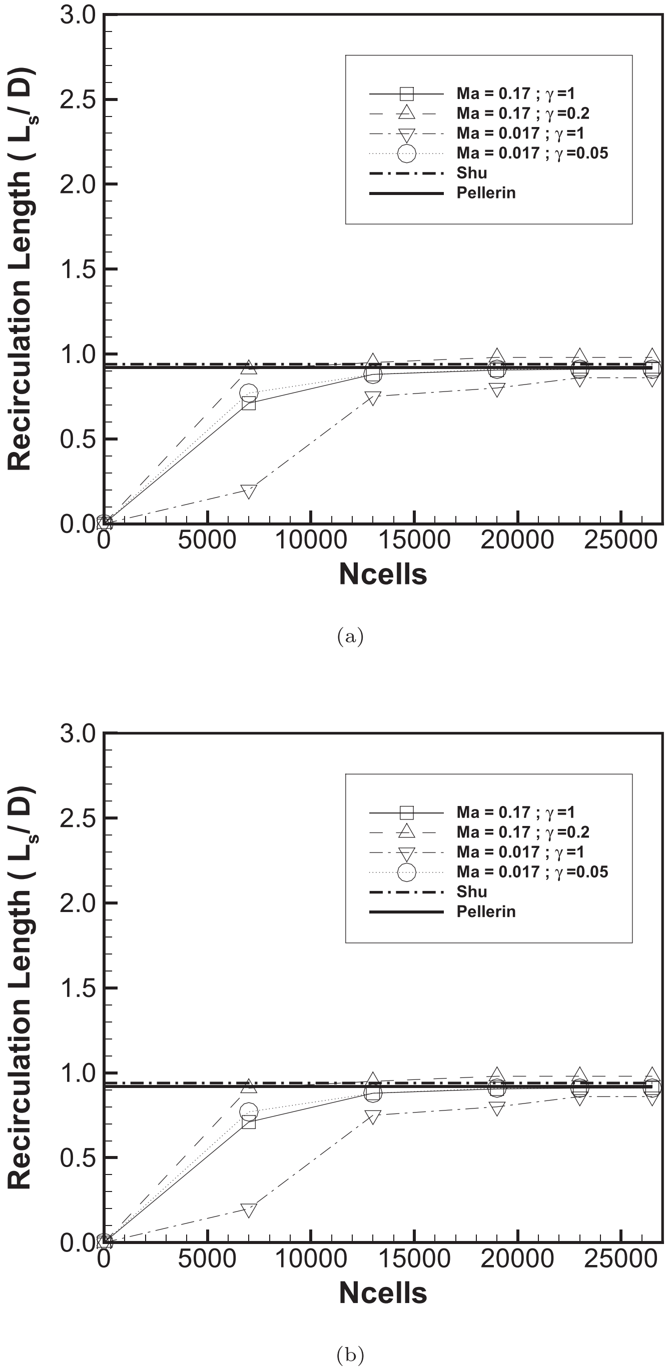
The convergence histories are plotted in Fig. 20. As shown in Fig. 20, the reduction in iterations required to reach convergence compared to the non-preconditioned case is 4.92x at $Re = 20$ and 9.625x at $Re = 40$ for $Ma = 0.017$. However there is an increase in iterations at the higher Mach number of 0.17. Due to the detrimental effect on convergence rates at $Ma = 0.17$, further test cases



(a)

(b)

Fig. 17. 3D flow over a circular cylinder: variation of drag coefficient with mesh density for varying U_∞ and γ for a) $Re = 20$ and b) $Re = 40$.



(a)

(b)

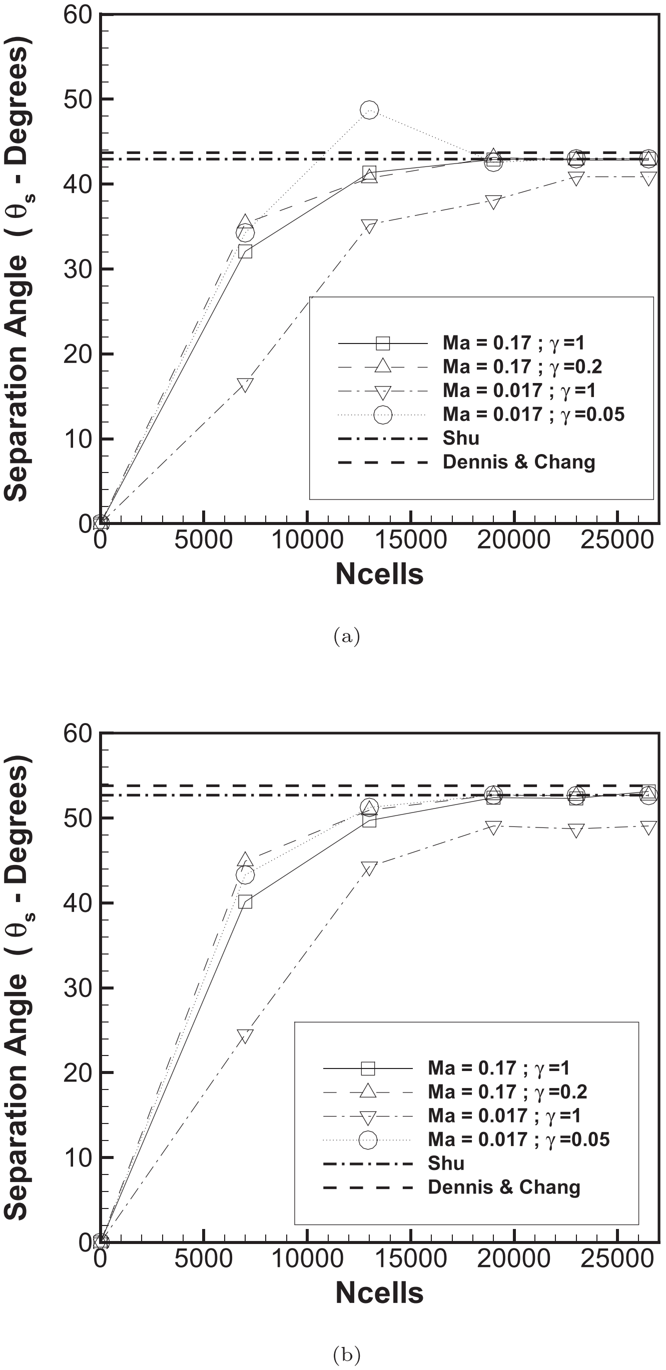
Fig. 18. 3D flow over a circular cylinder: variation of recirculation length with mesh density for varying U_∞ and γ for a) $Re = 20$ and b) $Re = 40$.

were run to investigate this issue. These additional test cases were chosen as per Table 3 and are shown in Table 8. Test Cases 9-12 used the same mesh with $N_{cells} = 23073$ as before but with a more moderate level of preconditioning. The convergence histories of these additional test cases can be seen in Fig. 21. These results show that any level of preconditioning has an adverse effect on convergence rates for $Ma = 0.17$.

Using an unstructured hexahedral mesh produces excellent results that agree with existing studies in the literature. The previous studies using the LBFS were performed by Shu et al. [10] with a cell-centred non-uniform O-grid mesh and by Pellerin et al. [11] with a vertex-centred tetrahedral mesh. They achieve excellent accuracy using 60501 cells and 63541 vertices respectively. Using the PLBFS on a fully unstructured hexahedral grid allows similar levels of accuracy to be attained while using a mesh with 23073 cells. Inspecting Table 5 shows that preconditioning allows the use of larger δx_{max} in larger cells in the mesh. This leads to more accurate results. It can be seen that Test Cases 3 and 7, which do not employ preconditioning, have δx_{max} values that are an order of magnitude lower than the preconditioned cases. In both cases, δx_{max} is less than the cell height of the first cell in the boundary layer. This results in drag coefficients, recirculation lengths and separation angles which are not as accurate as in the preconditioned cases. From the initial results in Fig. 20, preconditioning has

Table 8
Additional individual test case parameters for 3D flow over a circular cylinder problem.

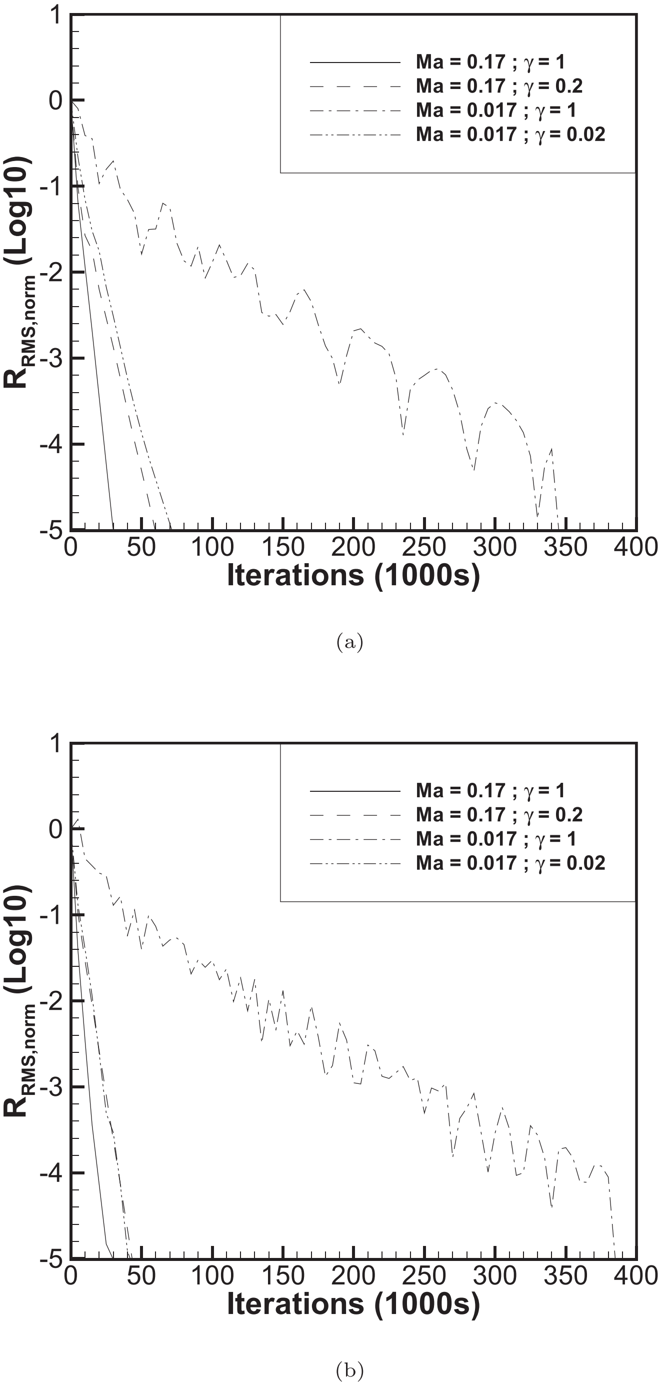
Test Case No.	U_∞	Re	Ma	γ	CN	$\Gamma_{vc,\ max}$	δx_{max}
9	0.1	20	0.17	0.5	3.88	8.21	0.346
10	0.1	20	0.17	0.75	5.33	5.88	0.230
11	0.1	40	0.17	0.5	3.88	4.41	0.173
12	0.1	40	0.17	0.75	5.33	2.94	0.115



(a)

(b)

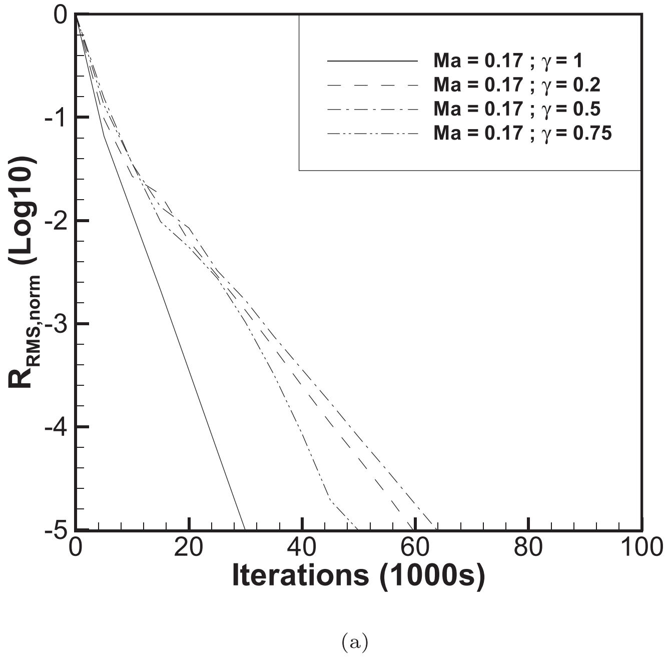
Fig. 19. 3D flow over a circular cylinder: variation of separation angle with mesh density for varying U_∞ and γ for a) $Re = 20$ and b) $Re = 40$.



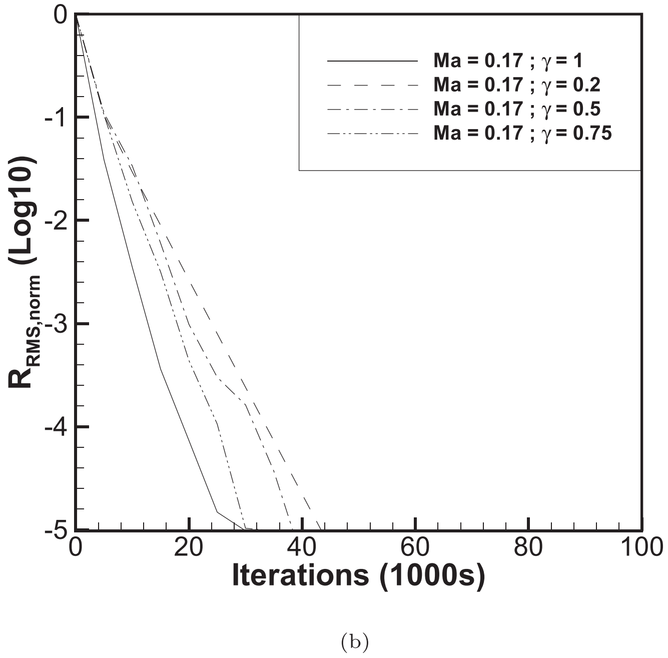
(a)

(b)

Fig. 20. 3D flow over a circular cylinder: convergence history for a) $Re = 20$ and b) $Re = 40$ on a mesh with $N_{cells} = 23073$.



(a)



(b)

Fig. 21. 3D flow over a circular cylinder: convergence history for a) $Re = 20$ and b) $Re = 40$ for a range of γ $N_{cells} = 23073$.

a significant acceleration effect on convergence at $Ma = 0.017$ but has a detrimental effect on the rate of convergence for $Ma = 0.17$. This can be attributed to the fact that the vast majority of the cells in the mesh can be considered viscous dominated for $\gamma = 1$ as the Reynolds number is very low in each test case. Any implementation of preconditioning for $Ma = 0.17$ would make these cells increasingly viscous dominated and increase runtimes.

5. Conclusion

In this paper a preconditioned lattice Boltzmann flux solver (PLBFS) has been successfully used to solve flow problems on 3D unstructured hexahedral meshes. This is demonstrated by successfully simulating 3D lid-driven cavity flow and 3D flow over a circular cylinder. The use of a unstructured hexahedral mesh topology is shown to enable the use of coarser meshes than those required using a structured mesh topology for similar levels of accuracy. This accuracy is also further increased by using preconditioning as it allows the use of a more optimal local lattice Boltzmann reconstruction in larger cells. This effect is demonstrated to be more significant at higher Reynolds numbers and lower Mach numbers. Preconditioning also has the benefit of accelerating convergence in the vast majority of cases. Again this effect is more beneficial at higher Reynolds numbers and lower Mach numbers as the unpreconditioned condition number is larger. It has also been shown that in certain cases, such as flow problems with low Reynolds numbers and very fine mesh refinement in the boundary layer as with flow over a circular cylinder, the acceleration due to preconditioning can be inimal. In such cases it is recommended to use moderate levels of preconditioning, so that the ratio of the viscous spectral radius to the convective spectral radius remains less than one, or to use a coarser grid where possible if sufficient accuracy can be achieved with such a mesh. The PLBFS has been shown to greatly enhance the use of unstructured meshes with the LBFS, particularly in flow problems with higher Reynolds numbers.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Brendan Walsh: Conceptualization, Methodology, Data curation, Writing - original draft, Visualization, Investigation, Software, Validation. **Fergal J. Boyle:** Supervision, Writing - review & editing, Project administration, Funding acquisition.

Acknowledgements and Source Code

The authors would like to gratefully acknowledge the Technological University Dublin Fiosraigh Scholarship programme which funded this research and also the support of NVIDIA Corporation for the donation of the Tesla K40 GPU used for this research. The source code that was developed during this research can be found at https://github.com/CHRG-Developer/3D-PLFBS-GPU.

Appendix A. Hybrid Green-Gauss/Weighted-Least-Squares Gradient Operator

The Hybrid Green-Gauss/Weighted-Least-Squares Gradient Operator is given as:

$$[\beta_i \mathbf{M}_i + 2(1-\beta_i)V_i\mathbf{I}]\nabla\mathbf{Q}_i = \sum_{j=1}^{N_{faces}} \left(\beta_i \alpha_{wj}\mathbf{x}_{ij} + 2(1-\beta_i)\alpha_{Gj}\mathbf{n}_j\right)S_j\Delta\mathbf{Q}_j \tag{A.1}$$

where $\mathbf{I}$ is the identity matrix, i is the index of the current cell where the gradient is being calculated, j is the face index of cell i, $\nabla\mathbf{Q}_i$ is the gradient vector of the macroscopic variables at the centroid of cell i, S_j is the surface area of face j, α_{Gj} is a constant assumed to be equal to 0.5 for all faces, $\mathbf{n}_j$ is the normal to face j, $\Delta\mathbf{Q}_j$ is the change in macroscopic variable between the the centroid of the current cell i and the centroid of the neighbouring cell with shared face j, β_i is the switching parameter for cell i and is given by:

$$\beta_i = min\left(1, \frac{2}{AR}\right) \tag{A.2}$$

where AR is the effective aspect ratio of the cell and is given by:

$$AR = \frac{2 \cdot max|\mathbf{r}_j - \mathbf{r}_i| \cdot max\left(S_j\right)}{V_i} \tag{A.3}$$

$\mathbf{r}_j$ is a position vector for the centroid of the neighbouring cell with shared face j, $\mathbf{r}_i$ is a position vector for the centroid of the current cell i and V_i is the cell volume of the current cell i. $\mathbf{M}_i$ is a matrix that weights the contribution of each neighbouring cell to the least-squares gradient calculation and is given by:

$$\mathbf{M}_i = \begin{bmatrix} \sum_{j=1}^{N_{faces}} \omega_j \Delta x \Delta x & \sum_{j=1}^{N_{faces}} \omega_j \Delta y \Delta x & \sum_{j=1}^{N_{faces}} \omega_j \Delta z \Delta x \\ \sum_{j=1}^{N_{faces}} \omega_j \Delta x \Delta y & \sum_{j=1}^{N_{faces}} \omega_j \Delta y \Delta y & \sum_{j=1}^{N_{faces}} \omega_j \Delta z \Delta y \\ \sum_{j=1}^{N_{faces}} \omega_j \Delta x \Delta z & \sum_{j=1}^{N_{faces}} \omega_j \Delta y \Delta z & \sum_{j=1}^{N_{faces}} \omega_j \Delta z \Delta z \end{bmatrix} \tag{A.4}$$

Δx, Δy and Δz are the changes in location along the Cartesian axes between the current cell's centroid and the j^{th} neighbouring cell's centroid. α_{wj} is an interpolation factor given by:

$$\alpha_{wj} = 4\left(\frac{|\mathbf{r}_k - \mathbf{r}_i| \cdot \mathbf{n}_j}{|\mathbf{r}_j - \mathbf{r}_i| \cdot \mathbf{n}_j}\right)^2 \tag{A.5}$$

where $\mathbf{r}_k$ is the centroid of the face j and ω_j is a weighting function given by:

$$\omega_j = \alpha_{wj} \frac{S_j}{|\mathbf{r}_j - \mathbf{r}_i|} \tag{A.6}$$

Finally $\mathbf{x}_{ij}$ is given by:

$$\mathbf{x}_{ij} = \frac{\mathbf{r}_j - \mathbf{r}_i}{|\mathbf{r}_j - \mathbf{r}_i|} \tag{A.7}$$

References

[1] Chapman S, Cowling TG. The mathematical theory of non-uniform gases. 3rd edition.. Cambridge University Press; 1991. ISBN 978-0521408448.

[2] Peng G, Xi H, Duncan C, Chou S-H. Finite volume scheme for the lattice Boltzmann method on unstructured meshes. Phys Rev E 1999;59(4):4675–82. doi:10.1103/PhysRevE.59.4675.

[3] Peng G, Xi H, Duncan C, Chou S-H. Lattice Boltzmann method on irregular meshes. Phys Rev E 1998;58(4):R4124–7. doi:10.1103/PhysRevE.58.R4124.

[4] Xi H, Peng G, Chou SH. Finite-volume lattice Boltzmann method. Phys Rev E 1999;59(5):6202–5. doi:10.1103/PhysRevE.59.6202.

[5] Stiebler M, Tölke J, Krafczyk M. An upwind discretization scheme for the finite volume lattice Boltzmann method. Comput Fluid 2006. doi:10.1016/j.compfluid.2005.09.002.

[6] Patil DV, Lakshmisha KN. Finite volume TVD formulation of lattice Boltzmann simulation on unstructured mesh. J Comput Phys 2009;228(14):5262–79. doi:10.1016/j.jcp.2009.04.008.

[7] Ubertini S, Bella G, Succi S. Unstructured lattice Boltzmann equation with memory. Math Comput Simulat 2006;72(2-6):237–41. doi:10.1016/j.matcom.2006.05.009.

[8] Zarghami A, Maghrebi MJ, Ghasemi J, Ubertini S. Lattice Boltzmann finite volume formulation with improved stability. Commun Comput Phys 2012;12(1):42–64. doi:10.4208/cicp.151210.140711a.

[9] Wang Y, Yang L, Shu C. From lattice Boltzmann method to lattice Boltzmann flux solver. Entropy 2015;17(11):7713–35. doi:10.3390/e17117713.

[10] Shu C, Wang Y, Teo CJ, Wu J. Development of lattice Boltzmann flux solver for simulation of incompressible flows. Adv Appl Math Mech 2014;6(4):436–60. doi:10.4208/aamm.2014.4.s2.

[11] Pellerin N, Leclaire S, Reggio M. Solving incompressible fluid flows on unstructured meshes with the lattice Boltzmann flux solver. Eng Appl Comput Fluid Mech 2017;11(1):310–27. doi:10.1080/19942060.2017.1292410.

[12] Wu Q-F, Shu C, Wang Y, Yang L-M. An effective lattice Boltzmann flux solver on arbitrarily unstructured meshes. Modern Phys Lett B 2018;32(12n13):1840012. doi:10.1142/S0217984918400122.

[13] Liu Y, Shu C, Zhang H, Yang L. A high order least square-based finite difference-finite volume method with lattice Boltzmann flux solver for simulation of incompressible flows on unstructured grids. J Comput Phys 2019:109019. doi:10.1016/J.JCP.2019.109019.

[14] Wang Y, Shu C, Teo CJ, Wu J, Yang L. Three-dimensional lattice Boltzmann flux solver and its applications to incompressible isothermal and thermal flows. Commun Comput Phys 2015;18(3):593–620. doi:10.4208/cicp.300514.160115a.

[15] Chorin A.J.. A numerical method for solving incompressible viscous flow problems. 1967.. 10.1006/jcph.1997.5716

[16] Turkel E. Preconditioned methods for solving the incompressible and low speed compressible equations. J Comput Phys 1987;72(2):277–98. doi:10.1016/0021-9991(87)90084-2.

[17] Malan AG, Lewis RW, Nithiarasu P. An improved unsteady, unstructured, artificial compressibility, finite volume scheme for viscous incompressible flows: Part I. Theory and implementation. Int J Numer Method Eng 2002;54(5):695–714. doi:10.1002/nme.447.

[18] Hejranfar K, Parseh K. Application of a preconditioned high-order accurate artificial compressibility-based incompressible flow solver in wide range of Reynolds numbers. Int J Numer Method Fluids 2018;86(1):46–77. doi:10.1002/fld.4407.

[19] He X, Doolen GD, Clark T. Comparison of the lattice Boltzmann method and the artificial compressibility method for Navier-Stokes equations. J Comput Phys 2002. doi:10.1006/jcph.2002.7064.

[20] Shah A, Yuan L, Khan A. Upwind compact finite difference scheme for time-accurate solution of the incompressible Navier-Stokes equations. Appl Math Comput 2010;215(9):3201–13. doi:10.1016/j.amc.2009.10.001.

[21] Kozel K., Kozel K., Louda P., Jarom, Íhoda Í. P. R.. Numerical solution of an unsteady flow using artificial compressibility method10.1.1.124.657.

[22] Hejranfar K, Parseh K. Preconditioned characteristic boundary conditions based on artificial compressibility method for solution of incompressible flows. J Comput Phys 2017. doi:10.1016/j.jcp.2017.05.014.

[23] He X, Luo L-S. Lattice Boltzmann model for the incompressible NavierStokes equation. J Stat Phys 1997;88(3/4):927–44. doi:10.1023/B:JOSS.0000015179.12689.e4.

[24] Krüger T, Kusumaatmaja H, Kuzmin A, Shardt O, Silva G, Viggen EM. The lattice Boltzmann method. Cham: Springer International Publishing; 2017. ISBN 978-3-319-44647-9. doi:10.1007/978-3-319-44649-3.

[25] Turkel E. Preconditioned methods for solving the incompressible and low speed compressible equations. J Comput Phys 1987;72(2):277–98. doi:10.1016/0021-9991(87)90084-2.

[26] Bernaschi M, Succi S, Chen H. Accelerated lattice Boltzmann schemes for steady-state flow simulations. J Sci Comput 2001;16(2):135–44. doi:10.1023/A:1012230722915.

[27] Verberg R, Ladd aJ. Simulation of low-Reynolds-number flow via a time-independent lattice-Boltzmann method.. Phys Rev E 1999;60(3):3366–73. doi:10.1103/PhysRevE.60.3366.

[28] Noble DR, Holdych DJ. Full Newton lattice Boltzmann method for time-steady flows using a direct linear solver. Int J Modern Phys C 2007;18(04):652–60. doi:10.1142/S0129183107010905.

[29] Tölke J, Krafczyk M, Rank E. A multigrid-solver for the discrete Boltzmann equation. J Stat Phys 2002;107(1/2):573–91. doi:10.1023/A:1014551813787.

[30] Lee T, Lin C-L. An eulerian description of the streaming process in the lattice Boltzmann equation. J Comput Phys 2003;185(2):445–71. doi:10.1016/S0021-9991(02)00065-7.

[31] Seta T, Takahashi R. Numerical Stability Analysis of FDLBM. J Stat Phys 2002;107(1/2):557–72. doi:10.1023/A:1014599729717.

[32] Tölke J, Krafczyk M, Schulz M, Rank E, Berrios R. Implicit discretization and nonuniform mesh refinement approaches for FD discretizations of LBGK Models. Int J Modern Phys C 1998;09(08):1143–57. doi:10.1142/S0129183198001059.

[33] Mavriplis DJ. Multigrid solution of the steady-state lattice Boltzmann equation. Comput Fluids 2006;35(8-9):793–804. doi:10.1016/j.compfluid.2005.07.020.

[34] Guo Z, Zhao TS, Shi Y. Preconditioned lattice-Boltzmann method for steady flows. Phys Rev E 2004;70(6):1–8. doi:10.1103/PhysRevE.70.066706.

[35] Premnath KN, Pattison MJ, Banerjee S. Steady state convergence acceleration of the generalized lattice Boltzmann equation with forcing term through preconditioning. J Comput Phys 2009;228(3):746–69. doi:10.1016/j.jcp.2008.09.028.

[36] Izquierdo S, Fueyo N. Preconditioned Navier-Stokes schemes from the generalised lattice Boltzmann equation. Progr Comput Fluid Dyn Int J 2008;8(1-4):189. doi:10.1504/PCFD.2008.018089.

[37] Izquierdo S, Fueyo N. Optimal preconditioning of lattice Boltzmann methods. J Comput Phys 2009;228(17):6479–95. doi:10.1016/j.jcp.2009.05.040.

[38] De Rosis A. Preconditioned lattice Boltzmann method for steady flows: a non-cascaded central-moments-based approach. Phys Rev E 2017;96(6):063308. doi:10.1103/PhysRevE.96.063308.

[39] Hajabdollahi F, Premnath KN. Improving the low Mach number steady state convergence of the cascaded lattice Boltzmann method by preconditioning. Comput Math Appl 2017. doi:10.1016/J.CAMWA.2016.12.034.

[40] Hajabdollahi F, Premnath KN. Galilean-invariant preconditioned central-moment lattice Boltzmann method without cubic velocity errors for efficient steady flow simulations. Phys Rev E 2018;97(5). doi:10.1103/PhysRevE.97.053303.

[41] Meng X, Wang L, Yang X, Guo Z. Preconditioned multiple-relaxation-time lattice Boltzmann equation model for incompressible flow in porous media. Phys Rev E 2018;98(5):1–12. doi:10.1103/PhysRevE.98.053309.

[42] Bhatnagar PL, Gross EP, Krook M. A model for collision processes in gases. I. small amplitude processes in charged and neutral one-component systems. Phys Rev 1954;94(3):511–25. doi:10.1103/PhysRev.94.511.

[43] Shima E, Kitamura K, Haga T. Greengauss/weighted-least-squares hybrid gradient reconstruction for arbitrary polyhedra unstructured grids. AIAA J 2013;51(11):2740–7. doi:10.2514/1.J052095.

[44] Jameson A, Schmidt W, Turkel E. Numerical solution of the Euler equations by finite volume methods using Runge Kutta time stepping schemes. 14th Fluid Plasma Dyn Conf 1981. doi:10.2514/6.1981-1259.

[45] Hirsch C. Numerical computation of internal and external flows : fundamentals of computational fluid dynamics. Elsevier/Butterworth-Heinemann; 2007. ISBN 9780750665940.

[46] Courant R, Friedrichs K, Lewy H. Über die partiellen differenzengleichungen der mathematischen physik. Math Annalen 1928;100(1):32–74. doi:10.1007/BF01448839.

[47] Mavriplis DJ, Jameson A. Multigrid solution of the Navier-Stokes equations on triangular meshes. AIAA J 1990;28(8):1415–25. doi:10.2514/3.25233.

[48] Niu XD, Shu C, Chew YT, Wang TG. Investigation of stability and hydrodynamics of different lattice Boltzmann models. J Stat Phys 2004;117(3-4):665–80. doi:10.1007/s10955-004-2264-x.

[49] Freitas CJ. Perspective - selected benchmarks from commercial CFD codes. J Fluid Eng 1995;117(2):208–18. doi:10.1115/1.2817132.

[50] Ku HC, Hirsh RS, Taylor TD. A pseudospectral method for solution of the three-dimensional incompressible Navier-Stokes equations. J Comput Phys 1987;70(2):439–62. doi:10.1016/0021-9991(87)90190-2.

[51] Ding H, Shu C, Yeo KS, Xu D. Numerical computation of three-dimensional incompressible viscous flows in the primitive variable form by local multiquadric differential quadrature method. Comput Method Appl Mech Eng 2006;195(7-8):516–33. doi:10.1016/j.cma.2005.02.006.

[52] Dennis S, Chang GZ. Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100. J Fluid Mech 1970;42(03):471–89. doi:10.1017/S0022112070001428.

[53] Shukla RK, Tatineni M, Zhong X. Very high-order compact finite difference schemes on non-uniform grids for incompressible Navier-Stokes equations. J Comput Phys 2007;224(2):1064–94. doi:10.1016/j.jcp.2006.11.007.