Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2023-0019

Vol. **37**, No. 2, pp. 547-574
February 2025

A Parallel Fully Implicit Unstructured Finite Volume Lattice Boltzmann Method for Incompressible Flows

Lei Xu[1,2], Rongliang Chen[1,2] and Xiao-Chuan Cai[3,*]

[1] *Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, 518055, China.*
[2] *Shenzhen Key Laboratory for Exascale Engineering and Scientific Computing, Shenzhen 518055, China.*
[3] *Department of Mathematics, University of Macau, Macau, China.*

Received 13 January 2023; Accepted (in revised version) 31 July 2023

Abstract. Lattice Boltzmann method is a popular approach in computational fluid dynamics, and it can be used explicitly or implicitly. The explicit methods require small time step size which is not desirable. In the fully implicit case, existing approaches either lack a scalable and robust parallel nonlinear solver, or don't allow the mesh to be fully unstructured preventing the method to be used for the simulation of fluid flows in large domains with complex geometry. In this paper, a parallel fully implicit second-order finite volume lattice Boltzmann method for incompressible flows on unstructured grids is introduced. The lattice Boltzmann equation is discretized by a finite volume method in space and an implicit backward Euler scheme in time. The resulting large sparse nonlinear system of algebraic equations is solved by a highly parallel Schwarz type domain decomposition preconditioned Newton-Krylov algorithm. The proposed method is validated by three benchmark problems with a wide range of Reynolds number: (a) pressure driven Poiseuille flow, (b) lid-driven cavity flows, and (c) viscous flows passing a circular cylinder. The numerical results show that the proposed method is robust for all the test cases and a superlinear speedup is obtained for solving a problem with over ten million degree of freedoms using thousands of processor cores.

AMS subject classifications: 76M12, 76M28, 76D99, 76F99
Key words: Lattice Boltzmann equation, fully implicit method, unstructured grids, Newton-Krylov-Schwarz, parallel computing.

1 Introduction

In the past 30 years, the standard lattice Boltzmann method (SLBM) has become an effective and promising approach in computational fluid dynamics. In SLBM, the lattice

---

[*]Corresponding author. *Email addresses:* `lei.xu@siat.ac.cn` (L. Xu), `rl.chen@siat.ac.cn` (R. Chen), `xccai@um.edu.mo` (X.-C. Cai)

Boltzmann equation (LBE) is solved to simulate the flow on uniform Cartesian grids with different collision models [1–4] in which the grid spacing is equal to the time step, and the spatial grid has to be a uniform Cartesian gird in order to execute the stream operation along the discrete velocity set. The Courant-Friedrichs-Lewy (CFL) number for SLBM has to be 1. To simulate incompressible flow problems defined on geometrically complex domains or flows with strong local gradients, SLBM requires a uniformly fine grid in order to capture the details, but an unstructured grid method only requires a finer grid near the body or the walls, and the grid for the far field can be quite coarse. This results in a much smaller overall grid. Moreover in SLBM, the staircase patterns on some of the boundaries may introduce geometrical irregularities, even on smooth surface due to the restrictions of the uniform Cartesian grid, and the fitting accuracy on the curved boundary is lower than that of a body-fitted grid. In recent years, a few off-lattice Boltzmann methods were devised to avoid the Cartesian grid restriction including grid refinement and multi-block methods [5, 6], hybrid lattice Boltzmann method [7, 8], finite difference lattice Boltzmann methods (FD-LBMs) [9–11], continuous or discontinuous finite element lattice Boltzmann methods [12, 13], finite volume lattice Boltzmann methods (FV-LBMs), the lattice Boltzmann flux solver [14] and the discrete unified gas kinetic scheme in the continuum limits [15].

In this paper, we introduce and study a fully implicit and fully unstructured version of FV-LBM originally proposed by Nannelli and Succi [16, 17], in which a coarse-grained distribution is defined via a cell averaging operator over a control volume, and a first-order Euler time-marching scheme is employed for the time integration. Recently, FV-LBM has been extended to unstructured grids. Patil et al. presented a cell-centered FV-LBM with total-variation diminishing formulation on 2D unstructured grids and they provided the Chapman-Enskog (CE) analysis for their FV-LBM with a forward Euler time integration on a special grid comprised of equilateral triangles [18, 19]. Misztal et al. presented a FV-LBM with a vertex-centered scheme, and studied two temporal discretizations (forward Euler method and the operator splitting method) on 2D unstructured grids yielding the Navier-Stokes equations through the CE expansion [20]. Li and Luo proposed an explicit cell-centered FV-LBM with a multiple-relaxation time collision model on arbitrary unstructured grids [21], in which the fluxes on the cell interfaces are evaluated by a low-diffusion Roe scheme. Subsequently, they presented a gas kinetic BGK scheme to evaluate the fluxes based on the formal analytical solution of the lattice Boltzmann BGK equation [22]. Chen and Schaefer devised a simple unified Godunov-type upwind approach which does not need Riemann solvers for the face flux calculation on 2D unstructured cell-centered triangular grids [23]. In our previous work, we developed a scalable parallel explicit FV-LBM for 3D incompressible flows [24], and extended the method to thermal incompressible flows [25] and inviscid high-speed compressible flows [26].

Note that all the works mentioned above are explicit or semi-implicit schemes whose time step size is constrained by the CFL number. To enhance the stability with larger time step sizes, a few fully implicit LBMs were proposed. Cevik and Albayrak proposed a fully implicit FV-LBM with Spalart-Allmaras one equation turbulence model for turbulent flows on 2D structured grids [27]. The corresponding linear system is solved by an alternating direction implicit method. Huang, Yang and Cai presented a fully implicit second-order FD-LBM, and they devised a parallel Newton-Krylov-Schwarz (NKS) algorithm to solve the large sparse nonlinear system arising from the discretization at each time step [10]. NKS has been successfully applied to solve various problems, for example, PDE constrained optimization problems [28,29], fluid-structure interaction problems [30–33], inverse source problems [34], elasticity problems [35], shallow water equations [36] and so on. In particular, the restricted additive Schwarz preconditioners combined with different solving strategies, such as variational inequality [37,38], multilevel algorithms [39], field-split methods [40], and so on, can greatly improve and expand the scope of application of the domain decomposition algorithm for large-scale flow problems. For steady-state problems, Li et al. developed a fully implicit FV-LBM on arbitrary unstructured grids [41, 42]. The corresponding linear systems are solved by the block symmetric-Gauss-Seidel algorithm. Huang, Yang and Cai proposed and studied a two-level inexact Newton method and a nonlinear elimination preconditioner to accelerate the convergence of Newton iteration in the NKS algorithm for high Reynolds number flow problems [11].

About the development of parallel algorithms for LBM, we mention that several explicit methods were studied on both homogeneous [43,44] and heterogeneous computing environments [45–48]. The partitions are obtained by dividing the fluid domain along the coordinate axes, and each subdomain is mapped to a processor. Recently, we investigated the parallel performance of FV-LBMs on unstructured grids for incompressible flows [24], thermal incompressible flows [25] and compressible flows [26]. However, none of the method is "fully implicit" on "unstructured grids". In this paper, we present a parallel fully implicit finite volume lattice Boltzmann method with a cell-centered scheme for incompressible flows on unstructured grids, and we employ a parallel NKS algorithm to solve the large nonlinear system at each time step. Several aspects of the proposed method are investigated including the robustness, the stability and the scalability using up to 7680 processor cores.

The remainder of this paper is organized as follows. In Section 2, the LBE and a fully implicit second-order FV-LBM are presented, together with suitable boundary conditions. In Section 3, we briefly describe all components of the NKS algorithm, and some numerical results are reported in Section 4. Finally, some concluding remarks are given in Section 5.

2 A fully implicit finite volume lattice Boltzmann method

We consider the LBE with Bhatnagar-Gross-Krook (BGK) collision model and the D2Q9 lattice model for 2D incompressible flows [1,2]:

$$\frac{\partial f_\alpha(\mathbf{x},t)}{\partial t}+\mathbf{e}_\alpha\cdot\nabla f_\alpha(\mathbf{x},t)=-\frac{1}{\tau}\left[f_\alpha(\mathbf{x},t)-f_\alpha^{eq}(\mathbf{x},t)\right],\quad \mathbf{x}\in\Omega,\ \ t\in(0,T), \tag{2.1}$$

where $f_\alpha(\mathbf{x},t)$ is the particle distribution function (PDF) at position $\mathbf{x}$ and time t, $\Omega \in R^2$ is the computational domain, $(0,T)$ is the time interval, $\mathbf{e}_\alpha$ is the discrete particle velocity given as

$$\mathbf{e}_\alpha = c\begin{cases} (0,0), & \alpha=0, \\ (\cos[(\alpha-1)\frac{\pi}{4}],\sin[(\alpha-1)\frac{\pi}{4}]), & \alpha=1-4, \\ (\sqrt{2}\cos[(\alpha-1)\frac{\pi}{4}],\sqrt{2}\sin[(\alpha-1)\frac{\pi}{4}]), & \alpha=5-8. \end{cases}$$

Here, $c=\sqrt{3}c_s$ is the constant speed for an isothermal model, c_s is the speed of the sound. The right-hand side is the BGK collision operator, $\tau=\nu/c_s^2$ is the relaxation time related to the kinematic viscosity ν, f_α^{eq} is the local equilibrium distribution function (EDF) which is approximated by the Maxwellian distribution function for low Mach number, and the function can be calculated by

$$f_\alpha^{eq} = \rho\omega_\alpha\left[1+\frac{\mathbf{e}_\alpha\cdot\mathbf{u}}{c_s^2}+\frac{(\mathbf{e}_\alpha\cdot\mathbf{u})^2}{2c_s^4}-\frac{\mathbf{u}\cdot\mathbf{u}}{2c_s^2}\right],$$

where ρ is the macroscopic flow density, and $\mathbf{u}$ is the macroscopic flow velocity computed by

$$\rho=\sum_\alpha f_\alpha, \quad \rho\mathbf{u}=\sum_\alpha f_\alpha\mathbf{e}_\alpha,$$

ω_α is the weighting factor and has the values as:

$$\omega_\alpha=\begin{cases} \frac{4}{9}, & \alpha=0, \\ \frac{1}{9}, & \alpha=1-4, \\ \frac{1}{36}, & \alpha=5-8, \end{cases}$$

for the D2Q9 lattice model. Under the incompressible limit, the LBE (2.1) implies the Navier-Stokes equations through the CE expansion

$$\begin{cases} \frac{\partial\rho}{\partial t}\nabla\cdot(\rho\mathbf{u})=0, \\ \frac{\partial(\rho\mathbf{u})}{\partial t}+\nabla\cdot(\rho\mathbf{u}\mathbf{u})=-\nabla p+\nabla\cdot[\rho\nu(\nabla\mathbf{u}+\nabla\mathbf{u}^T)], \end{cases}$$

where p is the pressure and can be directly calculated as $p=\rho c_s^2$ in the LBM simulation.

Next we consider the spatial and temporal discretization of LBE on unstructured grids. To numerically solve (2.1), the computational domain Ω is divided into a number of nonoverlapping polygons V_i and the cell-centered finite volume method (See Fig. 1) is employed to discretize the spatial terms. Each polygon becomes naturally a control volume (CV), and each cell centroid of CV is chosen to be the data point which store the PDFs and macroscopic variables.

Integrating over the cell i with cell volume V_i and its boundary ∂V_i (See Fig. 1), and employing the Gauss theorem, we obtain

$$\frac{\partial}{\partial t}\int_{V_i} f_\alpha(\mathbf{x},t)dV+\oint_{\partial V_i}(\mathbf{e}_\alpha\cdot\mathbf{n})f_\alpha(\mathbf{x},t)ds=\int_{V_i}-\frac{1}{\tau}\left[f_\alpha(\mathbf{x},t)-f_\alpha^{eq}(\mathbf{x},t)\right]dV, \tag{2.2}$$

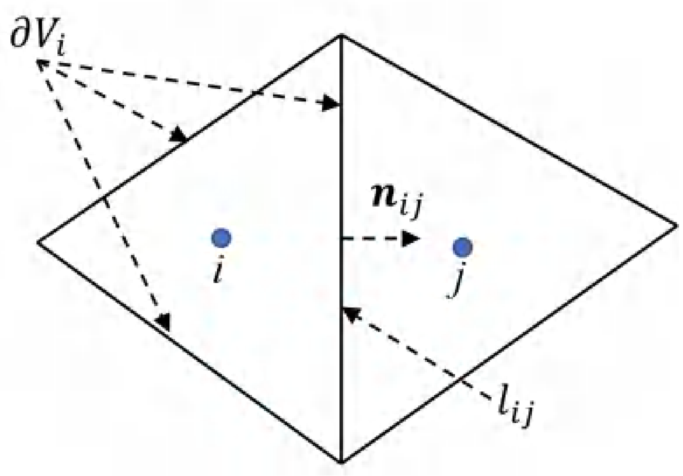


Figure 1: Cell-centered scheme on 2D unstructured grid, the blue solid circle is the cell centroid, $\mathbf{n}_{ij}$ and l_{ij} are the outward unit normal vector and the face length related to the interface of cell i and cell j, respectively, $\partial\Omega_i$ is all the boundary faces of cell i.

where $\mathbf{n}$ is the outward unit vector, locally normal to ∂V_i. The PDF at the centroid of the cell i is defined as $f_{\alpha i}$ denoting the cell-averaged value, and it is calculated by

$$f_{\alpha i}(t)=\frac{1}{V_i}\int_{V_i} f_\alpha(\mathbf{x},t)dV.$$

Then, (2.2) can be rewritten in a semi-discrete form as

$$\frac{d}{dt}f_{\alpha i}=A_{\alpha i}+C_{\alpha i}.$$

Here,

$$A_{\alpha i}=-\frac{1}{V_i}\oint_{\partial V_i} H(f_\alpha)dl, \tag{2.3}$$

and

$$C_{\alpha i}=-\frac{1}{V_i}\int_{V_i}\frac{1}{\tau}\left[f_\alpha(\mathbf{x},t)-f_\alpha^{eq}(\mathbf{x},t)\right]dV, \tag{2.4}$$

$A_{\alpha i}$ and $C_{\alpha i}$ are the advection and collision terms of particle, respectively. $H(f_\alpha)=(\mathbf{e}_\alpha\cdot\mathbf{n})f_\alpha(\mathbf{x},t)$ is the flux-density of f_α across the interface.

Because the PDFs and EDFs are assumed to be uniform over V_i, (2.4) can be calculated by

$$C_{\alpha i}=-\frac{1}{\tau}\left[f_{\alpha i}(\mathbf{x},t)-f_{\alpha i}^{eq}(\mathbf{x},t)\right],$$

and (2.3) can be discretized along the collection of faces owned by the cell boundary ∂V_i, we obtain

$$A_{\alpha i}=-\frac{1}{V_i}\sum_{m=1}^{N_s} H(f_{\alpha ij})l_{ij}, \tag{2.5}$$

where $f_{\alpha ij}$ is the PDF at the cell interface between cell i and cell j. It can be seen that, $f_{\alpha ij}$ in the right-hand side of (2.5) is needed to be reconstructed within V_i, and the flux-density H should be computed along the cell interface. To approximate the advective fluxes, a low-diffusion Roe scheme is applied. It evaluates the advective fluxes at a face of the CV from the left and right states by solving the Riemann problem [21] given as

$$H(f_{\alpha ij})=\frac{1}{2}\left[H(f_{\alpha ij}^{L})+H(f_{\alpha ij}^{R})-U|\mathbf{e}_{\alpha}\cdot\mathbf{n}_{ij}|_{\max}(f_{\alpha}^{R}-f_{\alpha}^{L})\right]), \tag{2.6}$$

where U is the scaled characteristic speed defined as

$$U=\max\left[\min(k\|\mathbf{u}\|),\frac{\nu}{\delta_x},\epsilon\right],$$

in which, ϵ and k are set to be 10^{-5} and 1, respectively. δ_x is the half distance between the centroid of cell i and cell j. $f_{\alpha ij}^{L}$ and $f_{\alpha ij}^{R}$ are the left and right states of the cell interface, respectively, reconstructed by piecewise linear reconstruction assuming the PDF is piecewise linearly distributed over the CV

$$\begin{cases} f_{\alpha ij}^{L}=f_{\alpha i}+\nabla f_{\alpha i}\cdot\mathbf{r}_L, \\ f_{\alpha ij}^{R}=f_{\alpha j}+\nabla f_{\alpha j}\cdot\mathbf{r}_R, \end{cases} \tag{2.7}$$

where $\mathbf{r}_L$ and $\mathbf{r}_R$ point from the cell centroid to the face-midpoint, $\nabla f_{\alpha i}$ and $\nabla f_{\alpha j}$ are the gradients of f_α at the cell centroid of cell i and cell j, respectively. The gradient of $f_{\alpha i}$ is computed by the inverse distance weighted least square method [49], because cell i has more neighbors than the number of components of $\nabla f_{\alpha i}$. In this method, $\nabla f_{\alpha i}$ can be computed by an optimization procedure to find the minimum of the function $G_{\alpha i}$ defined as

$$G_{\alpha i}=\min_{\nabla f_{\alpha i}}\left(\sum_{j=1}^{Ns}\frac{1}{r_{ij}}\|(f_{\alpha j}-f_{\alpha i})-\mathbf{r}_{ij}\cdot\nabla f_{\alpha i}\|^2\right),$$

where $\mathbf{r}_{ij}$ points from cell centroid i to j. $G_{\alpha i}$ is minimized by the conditions

$$\frac{\partial G_{\alpha i}}{\partial(\frac{\partial f_{\alpha i}}{\partial x})}=\frac{\partial G_{\alpha i}}{\partial(\frac{\partial f_{\alpha i}}{\partial y})}=0,$$

which leads to a $D\times D$ linear system, D is the dimensional number [21,50].

For the temporal discretization, a fully implicit first-order backward finite difference formula with a fixed time step size δ_t is employed. For a given semi-discrete system

$$\frac{d\mathbf{X}}{dt}=\mathbf{L}(\mathbf{X}),$$

the scheme is given as

$$\frac{\mathbf{X}^n-\mathbf{X}^{n-1}}{\delta t}=\mathbf{L}(\mathbf{X}^n), \tag{2.8}$$

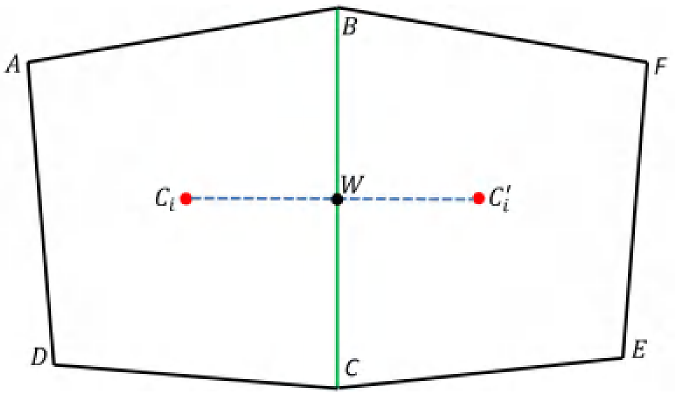


Figure 2: Boundary condition.

where $\mathbf{X}^n$ is the value of $\mathbf{X}$ at the nth time step. For simplicity, (2.8) can be rewritten as a sparse, nonlinear, algebraic system

$$F^n(\mathbf{X}^n)=0. \tag{2.9}$$

Based on the solution from the previous time step $\mathbf{X}^{n-1}$, the solution $\mathbf{X}^n$ can be obtained by solving the large nonlinear system of equations (2.9). In our algorithm, the solution vector $\mathbf{X}$ is arranged cell-by-cell in a field-coupling fashion, more precisely,

$$\mathbf{X}=(f_{00},f_{10},\cdots,f_{80},f_{01},f_{11},\cdots,f_{81},\cdots,f_{0N},f_{1N},\cdots,f_{8N})^T,$$

where N is the total number of cells. This ordering can improve the cache performance and the parallel efficiency in load and communication balance.

For comparison, we mention that in an explicit first-order forward Euler finite difference formula the CFL number is

$$\text{CFL}=\frac{\delta_t(\|\mathbf{e}_\alpha\|_{\max}+c)}{\frac{V_{\min}}{L^x_{\min}+L^y_{\min}}},$$

where $V_{\min}$ is the area of the smallest triangle, $L^x_{\min}$ and $L^y_{\min}$ are the projected lengths of the smallest triangle in the x and y directions, respectively.

Next we discuss the details of a strategy for boundary condition. The ghost cell method [21] and the non-equilibrium exploration method [51] are used to implement the boundary condition. In Fig. 2, $ABCD$ with its centroid C_i is an example of a boundary cell whose face BC lies on the boundary. The ghost cell $BCEF$ with centroid $C_i^{'}$ is the reflective image about the boundary face BC. W is the intersection point of $C_iC_i^{'}$ and face BC. According to the nonslip boundary condition, the relative velocity between the surface and the fluid directly at the surface is assumed to be 0. The macroscopic variables can be obtained by

$$\rho(C_i^{'})=\rho(C_i),\quad \mathbf{u}(W)=0,\quad \mathbf{u}(C_i^{'})=-\mathbf{u}(C_i).$$

The PDF $f_\alpha(W)$ can be divided into two parts: the equilibrium part $f_\alpha^{eq}(W)$ and the non-equilibrium part $f_\alpha^{neq}(W)$, that is,

$$f_\alpha(W)=f_\alpha^{eq}(W)+f_\alpha^{neq}(W).$$

$f_\alpha^{eq}(W)$ can be calculated based on the macroscopic variables at point W, $f_\alpha^{neq}(W)$ can be approximated by $f_\alpha^{neq}(C_i)$. Then the PDF at point $C_i^{'}$ can be calculated by

$$f_\alpha(C_i^{'})=2f_\alpha(W)-f_\alpha(C_i). \tag{2.10}$$

For other boundary conditions, the implementations are similar.

3 Monolithic Newton-Krylov-Schwarz algorithm

In this section, we introduce a monolithic Newton-Krylov-Schwarz algorithm to solve the nonlinear system (2.9). In the algorithm, an inexact Newton method is employed to solve the nonlinear system, in which a right-preconditioned Krylov subspace method (GMRES) is used to solve the Jacobian problem, and an overlapping restricted additive Schwarz method is adopted as the preconditioner. In the domain decomposition preconditioner, the subdomain problems are solved by a point-block LU or ILU factorization [52–54]. The main components of the algorithm are described as follows:

1. Take the solution of the previous time step $\mathbf{X}^{n-1}$ as the initial guess $\mathbf{X}_0^n$

$$\mathbf{X}_0^n=\mathbf{X}^{n-1};$$

2. For $k=0,1,2,\cdots$, until satisfying the stopping condition, do
   - **(a)** Compute the nonlinear function $F^n(\mathbf{X}_n^k)$, and form the corresponding Jacobian matrix J_k^n at $\mathbf{X}_n^k$;
   - **(b)** Inexactly solve the preconditioned Jacobian system to obtain the Newton direction $\mathbf{s}_k^n$ such that

$$\|J_k^n(M_k^n)^{-1}(M_k^n\mathbf{s}_k^n)+F^n(\mathbf{X}_{k-1}^n)\|\le\max\{\eta_r\|F^n(\mathbf{X}_{k-1}^n)\|,\eta_a\}; \tag{3.1}$$

   - **(c)** Find a step length λ_k^n by a cubic backtracking line search [55, 56], and update the approximation

$$\mathbf{X}_{k+1}^n=\mathbf{X}_k^n+\lambda_k^n\mathbf{s}_k^n.$$

3. Set $\mathbf{X}^n=\mathbf{X}_{k+1}^n$.

Here, $(M_k^n)^{-1}$ is a restricted additive Schwarz preconditioner to be described below. η_r and η_a are the relative and absolute tolerances for the linear solver, respectively.

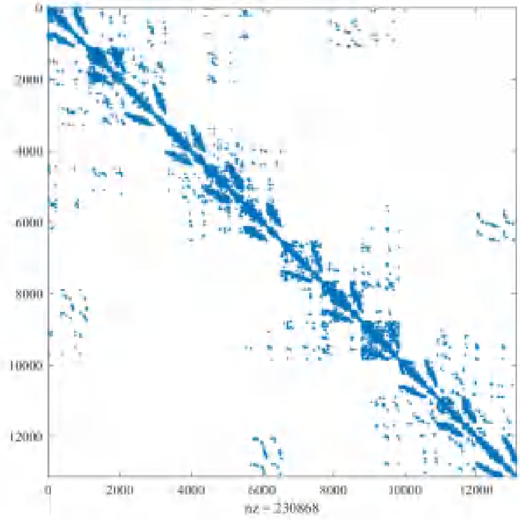


Figure 3: A 2D example of the non-zero structure of the Jacobian matrix. The non-zero elements are shown in blue.

Newton methods can be carried out with or without the explicit form of the Jacobian matrix. [57] provides an overview of the Jacobian-free Newton methods. In this paper, the explicit Jacobian matrix is computed since we need it to construct the preconditioner. To save the programmer's time, sometimes the Jacobian matrix is calculated approximately by a multicoloring finite difference method [58] or automatic differentiation method [59], but in this paper, the Jacobian matrix is calculated analytically. We also compare our analytical Jacobian with the approximate one proposed in [41], where the first order flux of the advective term is approximated by

$$\begin{aligned}
\frac{\partial H(f_{\alpha ij})}{\partial f_{\alpha i}} &\approx \frac{1}{2}\left[\frac{\partial H(f_{\alpha i})}{\partial f_{\alpha i}} + |\mathbf{e}_\alpha \cdot \mathbf{n}_{ij}|_{\max}\right],\\
\frac{\partial H(f_{\alpha ij})}{\partial f_{\alpha j}} &\approx \frac{1}{2}\left[\frac{\partial H(f_{\alpha j})}{\partial f_{\alpha j}} - |\mathbf{e}_\alpha \cdot \mathbf{n}_{ij}|_{\max}\right].
\end{aligned}$$

An example of the non-zero structure of the Jacobian matrix is shown in Fig. 3. In this work, the stopping condition for the Newton iteration (2) is determined by

$$\|F(\mathbf{X}^n)\| \le \min\{\widehat{\gamma}_a, \max\{\gamma_r \|F(\mathbf{X}^0)\|, \gamma_a\}\},$$

where $\widehat{\gamma}_a$, γ_r and γ_a are the safeguard, relative and absolute tolerance, respectively.

The most time-consuming step in NKS is to solve the large sparse nonsymmetric linear system (3.1) by a preconditioned Krylov subspace method (GMRES), and the most important component in the linear solver is the preconditioner, without which GMRES does not converge or converges slowly. A good choice of preconditioner can improve the convergence significantly. In this paper, an overlapping restricted additive Schwarz (RAS) preconditioner is employed to accelerate the convergence [53]. To introduce RAS, the grid in the computational domain Ω is divided into N_P nonoverlapping subdomains Ω_l $(l=1,2,\cdots,N_P)$, where N_P is the number of processor cores for parallel computing. Then the overlapping subdomains Ω_l^δ can be obtained by extending each subdomain Ω_l to include δ layers of elements belonging to its neighbors. Define the restriction operator $\mathbf{R}_l^\delta$ on each overlapping subdomain Ω_l^δ, which maps the global vector of unknowns in Ω to those belonging to Ω_l^δ, such that

$$\mathbf{X}_l^\delta=\mathbf{R}_l^\delta\mathbf{X}=\begin{pmatrix} \mathbf{I} & 0 \end{pmatrix}\begin{pmatrix} \mathbf{X}_l^\delta \\ \mathbf{X}\backslash\mathbf{X}_l^\delta \end{pmatrix},$$

where $\mathbf{X}\backslash\mathbf{X}_l^\delta$ denotes the unknowns outside the subdomain Ω_l^δ, $\mathbf{I}$ is the identity matrix. Then, the subdomain Jacobian matrix is constructed by

$$\mathbf{S}_l^\delta=\mathbf{R}_l^\delta\mathbf{J}_k^n(\mathbf{R}_l^\delta)^T, \quad l=1,2,\cdots,N_P,$$

which is the restriction of the global Jacobian matrix $\mathbf{J}_k^n$ to the subdomain Ω_l^δ. The RAS preconditioner is then defined as

$$(\mathbf{M}_k^n)^{-1}=\sum_{l=1}^{N_P}(\mathbf{R}_l^0)^T(\mathbf{B}_l^\delta)^{-1}\mathbf{R}_l^\delta,$$

where the restriction operators $\mathbf{R}_l^0$ are matrices similar to $\mathbf{R}_l^\delta$ which map the global vector of unknowns to those belonging to the non-overlapping subdomain Ω_l, $(\mathbf{B}_l^\delta)^{-1}$ is a preconditioner of the subdomain Jacobian $\mathbf{S}_l^\delta$. In this paper, $(\mathbf{B}_l^\delta)^{-1}$ is calculated by an incomplete LU (ILU) factorization with some levels of fill-ins [60].

4 Numerical results

In this section, we investigate the numerical behavior and parallel performance of the proposed method. The method is implemented using the Portable Extensible Toolkit for Scientific computing (PETSc) library of Argonne National Laboratory [61]. Grids are generated with ANSYS ICEM and partitioned with ParMETIS [62]. All Computations are performed on Tianhe-2A supercomputer, where the computing nodes are interconnected by a proprietary high performance network, with two 12-core Intel Ivy Bridge Xeon CPUs and 24 GB local memory. In the numerical experiments, all 24 CPU cores in each node are used and each subdomain is assigned to a processor core. The compute time is denoted as “Time (s)” in seconds. The stopping conditions for solving the nonlinear and linear systems are listed below

- The relative tolerance for nonlinear solver $\gamma_r=10^{-6}$;

- The absolute tolerance for nonlinear solver $\gamma_a = 10^{-10}$;
- The safeguard tolerance for nonlinear solver $\hat{\gamma}_a = 10^{-6}$;
- The relative tolerance for linear solver $\eta_r = 10^{-3}$;
- The absolute tolerance for linear solver $\eta_a = 10^{-11}$.

To test the accuracy, the convergence criterion for steady-state simulations is defined as follows

$$l_2 = \frac{\|\mathbf{X}^n - \mathbf{X}^{n-1}\|_2}{\|\mathbf{X}^n\|_2}. \tag{4.1}$$

4.1 Validation for the fully implicit finite volume lattice Boltzmann method

4.1.1 Pressure driven Poiseuille flow

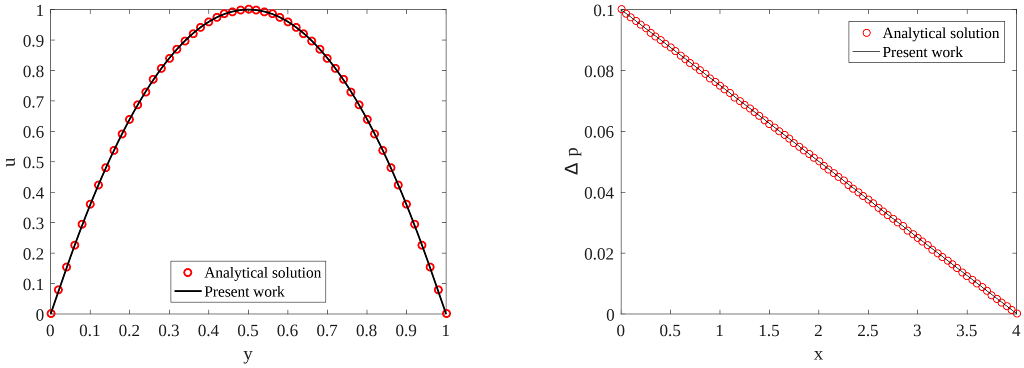
Numerical simulations for the pressure driven Poiseuille flow defined in a rectangular domain $[0,4]\times[0,1]$ is carried out to validate the proposed method. The exact velocity and pressure fields are expressed by

$$\begin{aligned} u(x,y,t) &= \frac{y(1-y)}{2\nu}\frac{dp}{dx}, \\ v(x,y) &= 0, \\ p(x,y) &= \frac{x}{4}\Delta p + p_{in}, \end{aligned}$$

where Δp is a constant pressure drop, ν is the dynamic viscosity, p_{in} is the inlet pressure, $p_{out} = p_{in} + \Delta p$ is the outlet pressure. We set $\Delta p = -0.1$ in this experiment. The top and bottom boundaries are nonslip walls. An unstructured grid with 40,000 quadrilateral cells is used for this experiment. Fig. 4 shows a comparison of the velocity values at $x = 2$ and the pressure values at $y = 0.5$ between the computed solution and the analytical solutions. As one can see the agreement between our results and the analytical results is good. Table 1 shows the errors of the velocity $u(y)$ and the order of convergence in L_2 norm. As given in Table 1, the observed order of accuracy for the proposed method is close to 2 as the formal order of accuracy suggests.

Table 1: The L_2 norm error of the velocity $u(y)$ and the order of convergence n.

Number of elements	L_2 error	n
40,000	2.04×10^{-5}	-
160,000	1.03×10^{-5}	1.98
640,000	5.19×10^{-6}	1.98



(a) (b)

Figure 4: A comparison of the analytical and numerical solutions for the Poiseuille flow. (a) The vertical velocity profiles at $x=2$, (b) The pressure drop at $y=0.5$.

4.1.2 Lid-driven cavity flow

Flow in a lid-driven cavity at different Reynolds numbers is solved to validate the present method. In the cavity flow, the top lid moves from left to right at a constant velocity $U_0=0.1$, and the other three walls are fixed as shown in Fig. 5. The Reynolds number Re of the flow is defined by U_0, the cavity width L, and the viscosity ν, that is, $Re=\frac{U_0L}{\nu}$.

To validate the method, a grid with 85,222 triangular elements (total degree of freedom DOF=766,998) is used for $Re=400$, 1000 and 3200, and a grid with 267,722 triangular elements (DOF=2,409,498) is adopted for $Re=5000$, 7500 and 10000. Fig. 6 shows the streamlines of the lid-driven cavity flows at six different Reynolds numbers. There are three vortices in the cavity when $Re\leq 1000$. There is a primary vortex almost occupies the

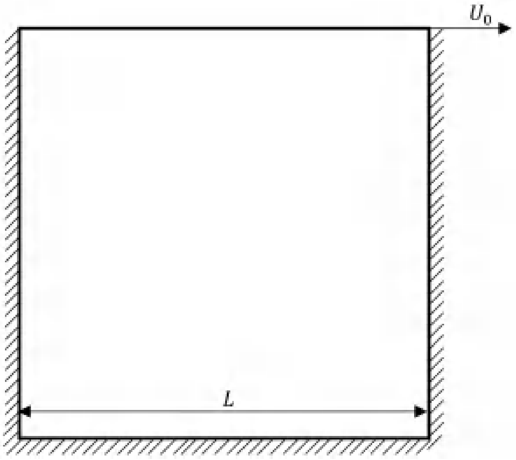
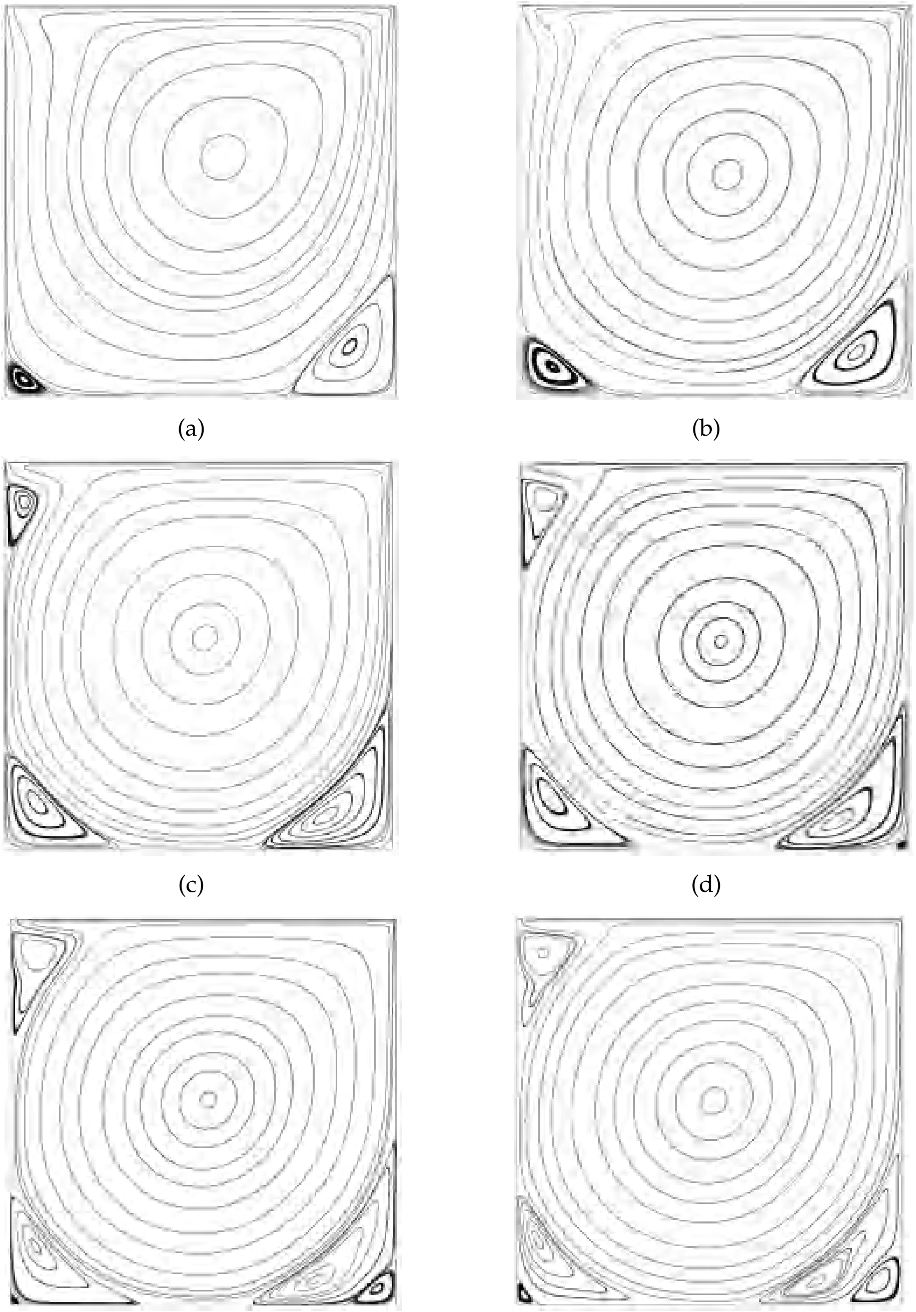


Figure 5: Sketch map of lid-driven cavity.



(a) (b)

(c) (d)

(e) (f)

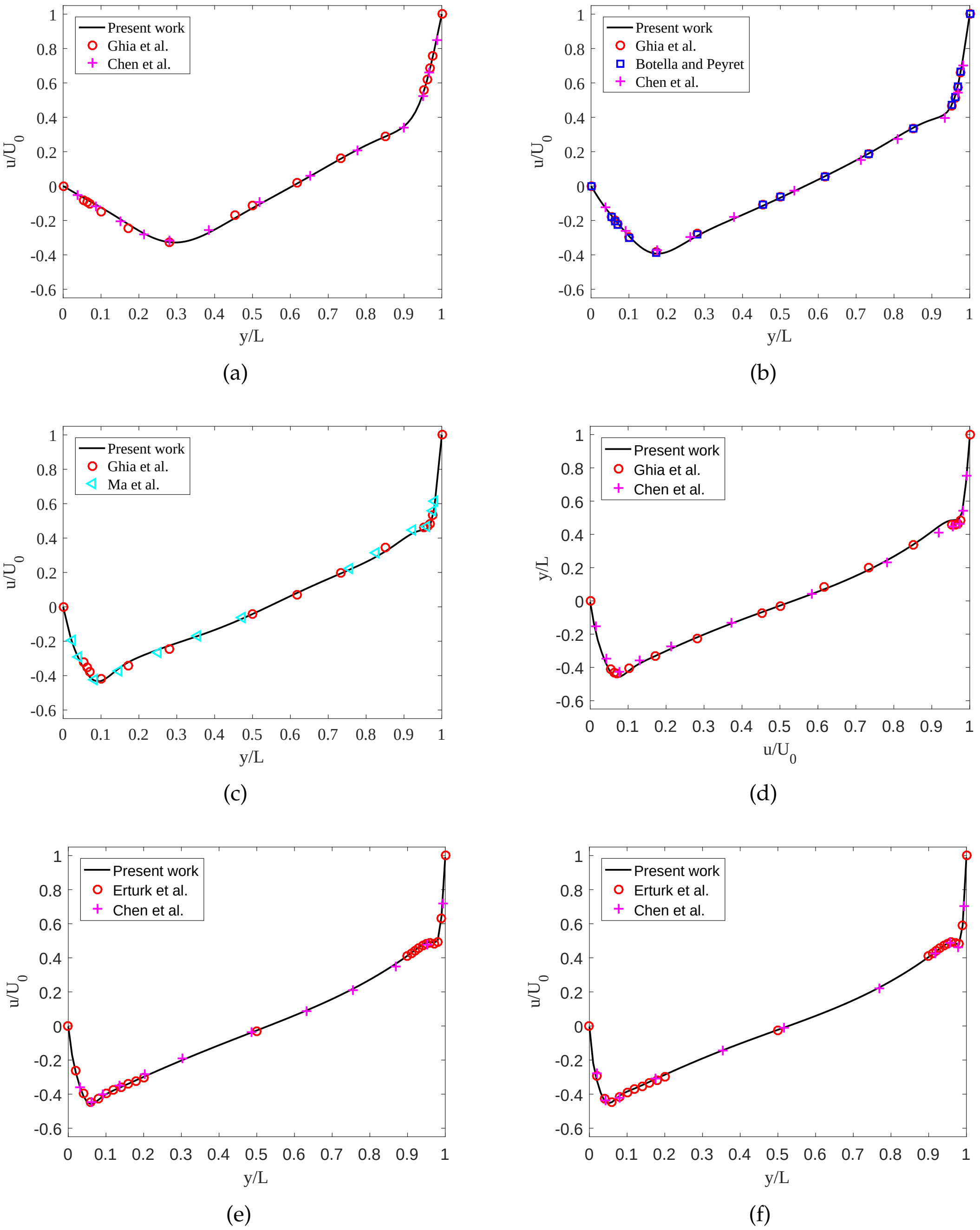
Figure 6: Streamlines of the lid-driven cavity flow computed with the present method for (a) $Re=400$, (b) $Re=1000$, (c) $Re=3200$, (d) $Re=5000$, (e) $Re=7500$ and (f) $Re=10000$.

Table 2: Comparison of the locations of the vortices at different Reynolds numbers for the lid-driven cavity flow.

Re	References	Primary vortex		Lower left vortex		Lower right vortex	
		x	y	x	y	x	y
400	Luo et al. [4]	0.554	0.605	0.049	0.049	0.885	0.123
	Patil et al. [18]	0.551	0.597	0.053	0.047	0.886	0.126
	Ghia et al. [63]	0.554	0.606	0.051	0.047	0.891	0.125
	Hou et al. [68]	0.561	0.608	0.055	0.051	0.890	0.126
	Vanka. [69]	0.556	0.600	0.050	0.050	0.888	0.119
	Present work	0.559	0.612	0.050	0.043	0.883	0.126
1000	Luo et al. [4]	0.531	0.562	0.084	0.080	0.862	0.110
	Ghia et al. [63]	0.531	0.563	0.086	0.078	0.859	0.109
	Erturk et al. [67]	0.530	0.565	0.083	0.078	0.863	0.112
	Hou et al. [68]	0.533	0.565	0.090	0.078	0.867	0.113
	Vanka. [69]	0.544	0.563	0.075	0.081	0.863	0.106
	Present work	0.532	0.568	0.085	0.076	0.861	0.113
3200	Patil et al. [18]	0.519	0.544	0.099	0.096	0.862	0.097
	Ghia et al. [63]	0.517	0.547	0.086	0.109	0.813	0.086
	Lin et al. [70]	0.518	0.540	0.081	0.120	0.825	0.084
	Present Work	0.518	0.542	0.083	0.121	0.836	0.085
5000	Ghia et al. [63]	0.512	0.535	0.070	0.137	0.809	0.074
	Erturk et al. [67]	0.515	0.535	0.073	0.137	0.805	0.073
	Hou et al. [68]	0.518	0.537	0.078	0.137	0.808	0.075
	Lin et al. [70]	0.515	0.535	0.073	0.137	0.805	0.073
	Present work	0.517	0.538	0.074	0.140	0.813	0.075
7500	Ghia et al. [63]	0.512	0.532	0.065	0.150	0.781	0.063
	Erturk et al. [67]	0.513	0.532	0.065	0.152	0.790	0.065
	Hou et al. [68]	0.518	0.533	0.071	0.153	0.792	0.067
	Lin et al. [70]	0.513	0.532	0.065	0.152	0.791	0.066
	Present work	0.512	0.532	0.064	0.152	0.789	0.065
10000	Ghia et al. [63]	0.512	0.535	0.070	0.136	0.808	0.074
	Erturk et al. [67]	0.512	0.530	0.058	0.163	0.777	0.060
	Chai et al. [71]	0.512	0.531	0.063	0.156	0.777	0.059
	Present work	0.512	0.530	0.057	0.162	0.782	0.061

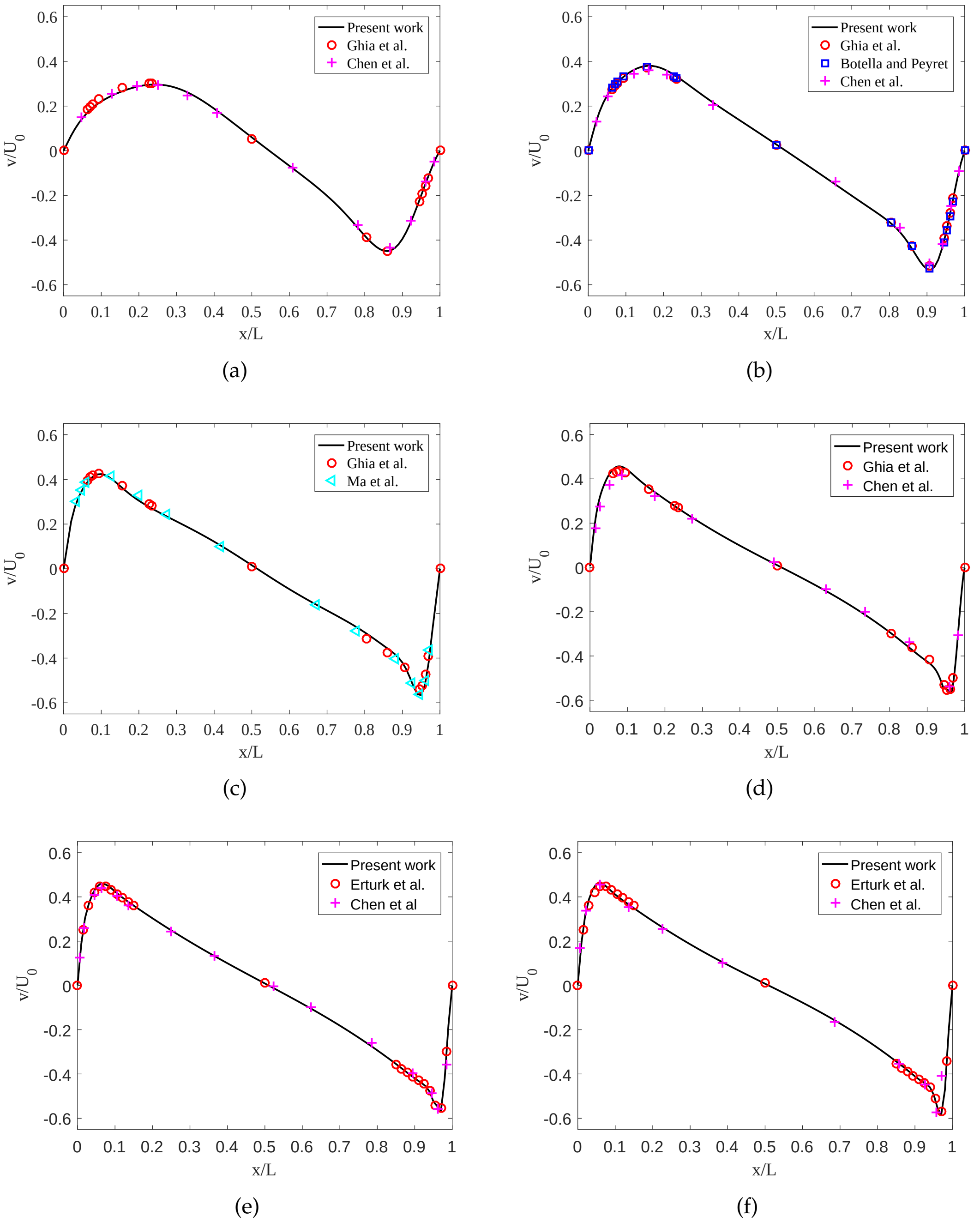
whole cavity and there are two secondary vortices locate at the lower left and lower right corners respectively. Another secondary vortex emerges at the upper left corner, when $Re=3200$. When Re increases to 5000, a third vortex arises. With the increase of Re, the center of the large vortex moves toward the center of the cavity.

To get more details about the flow, the horizonal and vertical velocity profiles along the centerlines of the cavity (see Figs. 7 and 8) are compared with the results from [63–67]. A comparison of the locations of the vortices from published results is listed in Table 2. It



(a) (b) (c) (d) (e) (f)

Figure 7: Vertical velocity profile along the centerlines of the cavity along with the results from Refs. [63–67], (a) $Re=400$, (b) $Re=1000$, (c) $Re=3200$, (d) $Re=5000$, (e) $Re=7500$ and (f)$Re=10000$.

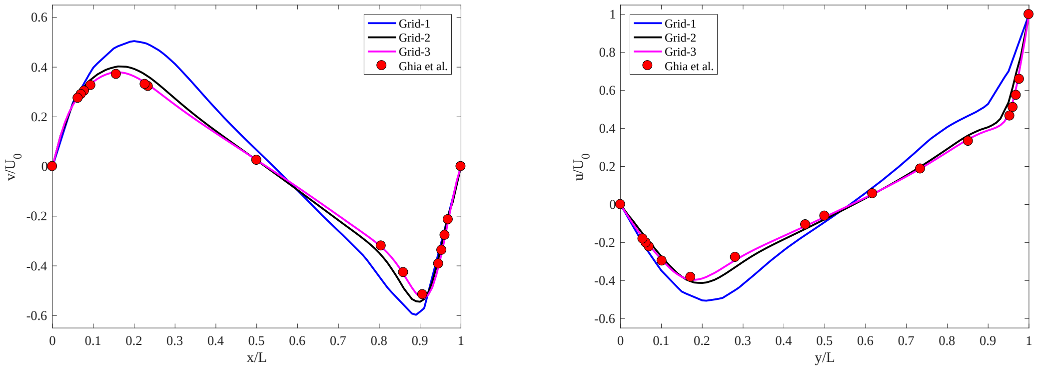


(a) (b)

(c) (d)

(e) (f)

Figure 8: Horizontal velocity profile along the centerlines of the cavity along with the results from Refs. [63–67], (a) $Re=400$, (b) $Re=1000$, (c) $Re=3200$, (d) $Re=5000$, (e) $Re=7500$ and (f)$Re=10000$.



(a) (b)

Figure 9: Horizontal (a) and Vertical (b) velocity profiles along the centerlines of the cavity with different grids, $Re=1000$.

can be seen that the results of the present work agree well with the references.

The grid convergence of the simulation is also investigated. We start the grid-refinement from a coarse grid with 1,250 triangular elements (DOF=11,250). Then, the grid is refined by dividing each triangular element into four triangles. Three different grids are employed to simulate the lid-driven cavity flow with $Re=1000$, which are denoted as “Grid-1”, “Grid-2” and “Grid-3”, respectively. Fig. 9 shows the normalized x-velocity u profiles along $x=0.5$ and the normalized y-velocity v profiles along $y=0.5$ for successively refined grids compared with the solutions of Ghia et al. [63]. It can be seen that the solutions become closer to the solutions of Ghia et al. with the refinement of the grid.

4.1.3 Flow past a circular cylinder

The incompressible laminar flow past a circular cylinder is a standard benchmark problem to test numerical schemes. In this subsection, the flows with three different Reynolds numbers $Re=10$, 20 and 40 are studied. The Reynolds number is defined by $Re=\frac{U_\infty D}{\nu}$, where the cylinder diameter and the initial velocity are set to be $D=1$ and $U_\infty=0.1$, respectively. The geometric configuration of the computational domain of size $56D\times 24D$ is shown in Fig. 10. The inlet of the flow is defined with a uniform x-velocity U_∞ and zero y-velocity. The outlet, bottom and top are considered as a fully developed boundary, and the wall on the circular cylinder is treated as a nonslip boundary. In this work, an unstructured grid with 174,504 (DOF=1,570,536) quadrilateral elements is employed. The streamlines at three different Reynolds numbers are shown in Fig. 11. It shows that the flow is separated and two counter-rotating vortices are formed behind the circular cylinder and there is no vortex shedding. The details of the length of the wakes L_s, separation angles θ_s and drag coefficients C_d are compared with the experimental and numerical

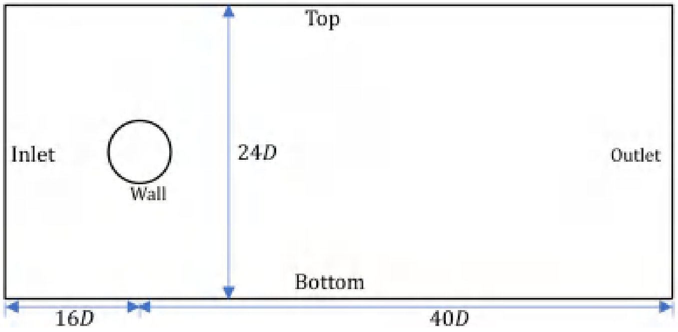
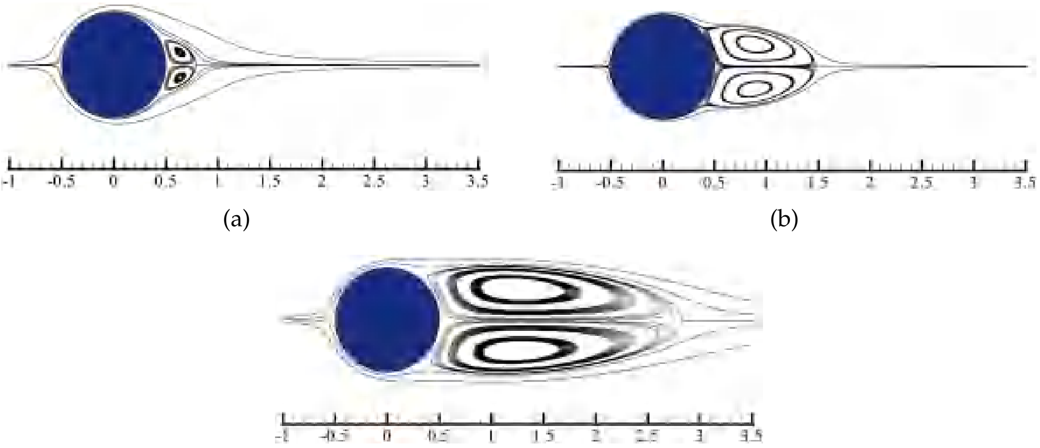


Figure 10: The configuration of flow past a circular cylinder.



(a)

(b)

(c)

Figure 11: Streamlines of flows past a circular cylinder at different Reynolds numbers, (a) $Re=10$, (b) $Re=20$ and (c) $Re=40$.

data from the references [9,11,21,72–76] in Table 3. The drag coefficient C_d is defined as

$$C_d = \frac{F_d}{\rho_\infty U_\infty^2 R},$$

where F_d is the drag force, which is the component of the aerodynamic force parallel to the flow direction. It can be seen that the results of the present work are in agreement with the published results in the literature.

4.2 Performance of the solver

In this subsection, parallel performance of the proposed solver is investigated including the influence of different overlapping size, subdomain solvers, the strong scalability, and

Table 3: The flow past a circular cylinder. A comparison of the wake length L_s/D, separation angles θ_s and drag coefficients C_d for flows past a circular cylinder.

Case	References	$2L_s/D$	θ_s	C_d
$Re=10$	Hejranfar and Ezzatneshan [9]	0.49	29.8°	2.801
	Huang, Yang and Cai [11]	0.50	29.5°	2.893
	Li and Luo [21]	0.66	31.8°	3.003
	Coutanceau and Bouard [72]	0.68	32.5°	–
	Dennis and Chang [73]	0.53	29.6°	2.846
	Imamura et al. [74]	0.49	29.8°	2.807
	Mei and Shyy [75]	0.49	30.0°	–
	Present work	0.58	32.2°	2.874
$Re=20$	Hejranfar and Ezzatneshan [9]	1.85	43.6°	2.021
	Huang, Yang and Cai [11]	1.87	44.5°	2.050
	Patil and Lakshmisha [18]	1.88	42.8°	1.949
	Li and Luo [21]	2.04	44.7°	2.118
	Dennis and Chang [73]	1.88	–	2.045
	Imamura et al. [74]	1.85	43.3°	2.051
	Mei and Shyy [75]	1.80	42.1°	–
	Present work	1.94	44.3°	2.099
$Re=40$	Hejranfar and Ezzatneshan [9]	4.51	51.9°	1.515
	Huang, Yang and Cai [11]	4.59	53.9°	1.528
	Li and Luo [21]	4.70	54.7°	1.568
	Dennis and Chang [73]	4.69	53.8°	1.522
	Imamura et al. [74]	4.45	52.4°	1.538
	Mei and Shyy [75]	4.38	50.1°	–
	Wu and Shu [76]	4.62	–	1.565
	Present work	4.65	53.7°	1.562

a comparison of an explicit method and the fully implicit method based on approximate and analytical Jacobian matrix. In the tables below, "GMRES" and "Newton" refer to the number of GMRES iterations and the number of Newton iterations, respectively. "Time (s)" represents the compute time in second.

The choice of subdomain solver and overlapping size have a remarkable influence on the performance of RAS preconditioner. To study the impact, the lid-driven cavity flow with $Re=1000$ is investigated. An unstructured grid with 570,902 (DOF=5,138,118) quadrilateral elements is used. The time step size is $\delta_t=0.0032$. Table 4 illustrates the numerical performance in terms of different subdomain solvers, in which different level of fill-ins is used. It can be seen that the number of Newton iterations (Newton) does not change with respect to the level of fill-ins and the number of parallel processors, and the number of GMRES iterations (GMRES) decreases as the level of fill-in increases, that

Table 4: Performance of NKS with different subdomain solvers. The lid-driven cavity flow with $Re=1000$ and a grid with 570,902 (DOF=5,138,118) quadrilateral elements is tested. The time step size is fixed to $\delta_t=0.0032$, and the overlapping size is $\delta=1$. The results are averaged over the first 10 time steps.

	Subdomain solver	N_p				
		240	480	960	1920	3840
Newton	ILU(0)	3.00	3.00	3.00	3.00	3.00
	ILU(1)	3.00	3.00	3.00	3.00	3.00
	ILU(2)	3.00	3.00	3.00	3.00	3.00
GMRES	ILU(0)	9.03	9.03	9.03	9.00	8.73
	ILU(1)	6.03	6.03	6.03	6.03	6.03
	ILU(2)	5.83	5.83	5.83	5.87	5.90
Time (s)	ILU(0)	43.35	21.25	10.97	6.18	3.76
	ILU(1)	78.65	38.98	19.54	9.76	3.39
	ILU(2)	158.80	79.97	40.31	18.02	4.54

is because a higher level of fill-ins generates a better subdomain preconditioner, which makes the global preconditioner stronger. However, as is shown in Table 4, when we use a higher level of fill-ins, more time is needed on the ILU factorization in the preconditioner step, which make the total compute time larger. In this test, the ideal level of fill-ins is 0, for which the computing time per time step reaches the minimum.

To study the influence of the overlapping size in the preconditioner, the lid-driven cavity flow with $Re=1000$ is tested on 1920 processor cores. A grid with 930,337 quadrilateral elements (DOF$=$8,373,033) is used. The time step size is fixed to $\delta_t=0.0032$, and the subdomain solver is ILU(1). The results are averaged over the first 10 time steps. The numerical results with different overlapping sizes are shown in Table 5. According to the theory of domain decomposition method, larger δ means stronger preconditioner and the preconditioned GMRES convergences faster. At the same time, the size of the subdomain problem grows with the increase of the overlapping size δ, which leads to more computing time for the subdomain problem and more communication time. In this test, the best result is obtained with $\delta=0$ in terms of the total compute time.

To investigate the effect of Reynolds numbers, we compare the numbers of nonlinear and linear iterations with a fixed initial velocity $U_0=0.1$. This test is executed on 240 processor cores, the time step size is fixed to $\delta_t=0.0032$, and the subdomain solver and overlapping size are ILU(0) and 0, respectively. An unstructured grid with 208,302 triangle elements (DOF$=$874,718) is used. Table 6 lists the number of nonlinear and linear iterations, and the compute time at different Reynolds numbers and viscosities. It may be seen that the number of nonlinear iterations does not change a lot, and the number of linear iterations increases as the Reynolds number increases, which indicates that the solver is quite robust with respect to the Reynolds numbers.

Next, a comparison between the fully implicit method and an explicit finite volume lattice Boltzmann method [24] is conducted on 480 processor cores. An unstructured grid

Table 5: Performance of NKS with different overlapping size. The lid-driven cavity flow with $Re=1000$ and a grid with 930,337 quadrilateral elements (DOF=8,373,033) is tested on 1920 processor cores. The time step size is fixed to $\delta_t=0.0032$, and the subdomain solver is ILU(1). The results are averaged over the first 10 time steps.

δ	Newton	GMRES	Time (s)
0	3.00	14.67	9.49
1	3.00	11.90	12.51
2	3.00	11.83	16.06
3	3.00	11.07	20.32
4	3.00	11.17	25.06

Table 6: The lid-driven cavity flow. A comparison of NKS with different Reynolds numbers. An unstructured grid with 208,302 triangular elements (DOF=874,718) is used. This test is executed on 240 processor cores, the time step size is fixed to $\delta_t=0.0032$, the subdomain solver is ILU(0), and the overlapping size is $\delta=0$. The results are averaged over the first ten time steps.

Case	viscosity	Newton	GMRES	Time (s)
$Re=400$	$\nu=2.5\times10^{-4}$	2.6	15.73	8.29
$Re=1000$	$\nu=1.0\times10^{-4}$	2.6	15.07	8.24
$Re=3200$	$\nu=3.125\times10^{-5}$	2.6	20.38	9.02
$Re=5000$	$\nu=2.0\times10^{-5}$	2.5	25.00	9.44
$Re=7500$	$\nu=1.333\times10^{-5}$	2.4	29.54	9.78
$Re=10000$	$\nu=1.0\times10^{-5}$	2.4	34.29	10.48

Table 7: The flow past a circular cylinder. A comparison of NKS with explicit finite volume lattice Boltzmann method is investigated on 480 processor cores. An unstructured grid with 174,504 quadrilateral elements (DOF=1,570,536) is employed, the l_2 relative error is set to be 5×10^{-8}, the subdomain solver is ILU(0), and the overlapping size is $\delta=0$, the time step size is 0.012 and 4×10^{-4} for implicit and explicit method, respectively.

Case	Explicit method		Implicit method	
	Iteration	Time (s)	Iteration	Time (s)
$Re=10$	2.7×10^{7}	2.5×10^{4}	8.9×10^{4}	1.1×10^{4}
$Re=20$	9.9×10^{5}	9.3×10^{3}	3.3×10^{3}	5.8×10^{3}
$Re=40$	1.2×10^{6}	1.2×10^{4}	4.0×10^{4}	6.9×10^{3}

with 174,504 quadrilateral elements (DOF= 1,570,536) is adopted, the l_2 relative error (4.1) is set to be 5×10^{-8}, the subdomain solver is ILU(0), and the overlapping size is $\delta=0$, the time step size is 0.012 and 4×10^{-4} for implicit and explicit method, respectively. The numerical results shown in Table 7 indicates that the fully implicit method runs faster than the explicit method. "Iteration" in Table 7 refers to the number of time steps.

In our method, the Jacobian matrix is computed analytically including all terms. The nonlinear solver convergences well with only a small number of iterations. Here, we

Table 8: Performance of NKS with different subdomain solvers. The lid-driven cavity flow with $Re=1000$ and a grid with 752,121 quadrilateral elements (DOF=6,769,089) is tested. The time step size is fixed to $\delta_t=0.0032$, the subdomain solver is ILU(0), and the overlapping size is $\delta=0$. The results are averaged over the first 10 time steps.

	Newton	GMRES	Time (s)
Approximate Jacobian	8.3	16.92	53.64
Multicoloring FD Jacobian	47.7	1.00	30.64
Our analytical Jacobian	2.6	8.88	12.60

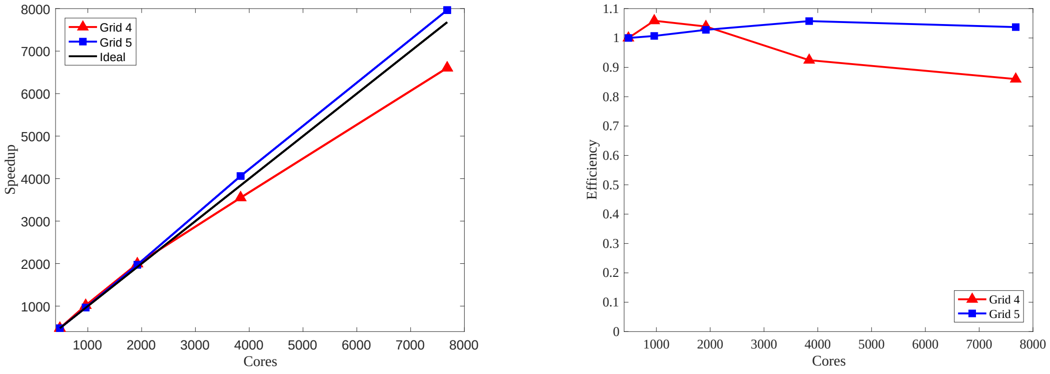
compare the approximate Jacobian matrix proposed in [41] where some gradient terms are ignored, and multicoloring finite difference (FD) method with our method. The comparison is tested for the lid-driven flow problem with $Re=1000$ on 960 processor cores. An unstructured grid with 752,121 quadrilateral elements (DOF= 6,769,089) is adopted. Table 8 gives the numerical results and shows that the implicit method with analytical Jacobian matrix is better than the two others with respect to the number of Newton iterations and the compute time. In the rest of this subsection, the strong scalability of the proposed algorithm is investigated for the lid-driven cavity flow problem with $Re=1000$. The strong scalability of a parallel algorithm measures its parallel speedup and efficiency as the number of processor cores increases and the total size of the grid keeps constant. The speedup and efficiency are defined as

$$\text{speedup}=\frac{t_M}{t_N} \quad \text{and} \quad \text{efficiency}=\frac{M\times t_M}{N\times t_N},$$

where t_M and t_N are the total compute times measured with M and N processor cores ($M \leq N$), respectively. To study the influence of the grid size on the parallel performance, two different grids are tested. One consists of 1,289,065 quadrilateral elements (DOF= 11,601,585), and the other 2,143,970 quadrilateral elements (DOF= 19,295,730), which are denoted as "Grid 4" and "Grid 5", respectively. We run all the tests with a fixed time step $\delta_t=0.0019$ for 10 time steps. ILU(1) is taken as the subdomain solver, and the overlapping size is 1. The number of linear and nonlinear iterations and the compute time are reported in Table 9. "Ideal" refers to the ideal parallel speedup. All the documented results are averaged values over the first 10 time steps. It can be concluded that the number of Newton iterations is independent of the number of processor cores and subdomain solvers, and the number of GMRES iterations increases slightly as the number of processor cores increases. Fig. 12 shows the parallel speedup and efficiency. We can see that the case with "Grid 5" has a superlinear speedup with up to 7680 processor cores. The parallel efficiency is 86.0% for "Grid 4" and 103.7% for "Grid5" when the number of processor cores increases from 480 to 7680. One interpretation of the superlinear speedup is much better cache performance when the number of processors is large.

Table 9: Parallel performance of the proposed fully implicit FV-LBM. The lid-driven cavity flow problem with $Re=1000$ is tested. Grid 4 has 1,289,065 quadrilateral elements (DOF=11,601,585), Grid 5 consists of 2,143,970 quadrilateral elements (DOF=19,295,730). The time step size is fixed to $\delta_t=0.0019$, ILU(1) is employed, and the overlapping size is $\delta=1$. The results are averaged over the first 10 time steps.

Grid	N_p	Newton	GMRES	Time (s)	Speedup	Ideal	Efficiency
Grid 4	480	3.20	3.94	107.61	1.00	1	100.0%
	960	3.20	3.94	50.82	2.12	2	105.9%
	1920	3.20	3.94	25.89	4.16	4	103.9%
	3840	3.20	3.94	14.55	7.40	8	92.5%
	7680	3.00	4.28	7.82	13.76	16	86.0%
Grid 5	480	3.30	3.70	201.27	1.0	1	100.0%
	960	3.30	3.94	99.94	2.01	2	100.7%
	1920	3.30	4.36	48.96	4.11	4	102.8%
	3840	3.30	5.45	23.79	8.46	8	105.8%
	7680	3.30	5.21	12.13	16.59	16	103.7%



(a) (b)

Figure 12: Parallel speedup (a) and efficiency (b) for solving the driven cavity flow as in Table 9. The time step size is fixed to $\delta t=0.0019$, ILU(1) is employed, and the overlapping size $\delta=1$. The results are averaged over the first 10 time steps.

5 Conclusions

In this paper, a fully implicit cell-centered finite volume lattice Boltzmann method on unstructured grids is presented for solving lattice Boltzmann equation, and a highly parallel scalable Newton-Krylov-Schwarz algorithm for solving the large sparse nonlinear system arising from the spatial and temporal discretizations at each time step is developed. The method is validated by two benchmark problems: (a) lid-driven flow problems with Re=400, 1000, 3200, 5000, 7500 and 10000, and (b) viscous flows past a circular cylinder with $Re=10$, 20, and 40. Different subdomain solver and overlapping size are compared, which shows the ideal subdomain solver is ILU(0) and the best overlapping size is 0 in our numerical results. A comparison of approximate, multicoloring FD and analytical Jacobian matrix is investigated, and we found that the analytical Jacobian matrix has a better performance in terms of the compute times the number of nonlinear and linear iterations. The test of different Reynolds numbers shows that our solver is robust with respect to a wide range of Reynolds numbers. The parallel performance exhibits a superlinear speedup on a supercomputer with up to 7680 processor cores.

In the Boltzmann equation, the nonlocal term (advection) is linear and the local term (collision) is nonlinear, but the nonlocal and nonlinear terms appear in the same term $\mathbf{u}\cdot\nabla\mathbf{u}$ in the Navier-Stokes equations, therefore, the method based on Boltzmann equation is often more feasible to simulate some flow problems. However, compared to the classical FVM solvers for the incompressible Navier-Stokes equations, the proposed solver has more independent variables, which requires more memory and communication. The proposed method can be used for other lattice models such as D2Q5 and D2Q13 since they have similar lattice structure and particle equilibrium distribution functions, a similar parallel scalability is expected. It is convenient to move to 3D incompressible flows. The fully implicit lattice Boltzmann method on unstructured grids and the parallel Schwarz type domain decomposition preconditioned Newton-Krylov algorithm do not change. The main effort is to incorporate a 3D lattice velocity model and its corresponding particle equilibrium distribution functions.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (NSFC) (Nos. 62161160312, 12101588 and 12071461), FDCT0141/2020/A3,0079/2021/AFJ, MYRG-GRG2023-00102-FST-UMDF, Guangdong Basic and Applied Basic Research Foundation (No. 2023B1515250005) and Shenzhen Science and Technology Program (No. JCYJ20241202124911016). The debug of numerical simulations in this study were partially carried out on the ORISE Supercomputer.

References

[1] S. Chen, H. Chen, D. Martnez and W. Matthaeus, Lattice Boltzmann model for simulation of magnetohydrodynamics, Phys. Rev. Lett., 67 (1991), 3776-3779.

[2] Y. H. Qian, D. D'Humieres and P. Lallemand, Lattice BGK model for Navier-Stokes equation, EPL, 17 (1992), 479-484.

[3] P. Lallemand and L.-S. Luo, Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability, Phys. Rev. E, 61 (2000), 6546-6562.

[4] L.-S. Luo, W. Liao, X. Chen, Y. Peng and W. Zhang, Numerics of the lattice Boltzmann method: Effects of collision models on the lattice Boltzmann simulations, Phys. Rev. E, 83 (2011), 056710.

[5] F. Schornbaum and U. Rüde, Massively parallel algorithms for the lattice Boltzmann method on nonuniform grids, SIAM J. Sci. Comput., 38 (2016), C96-C126.

[6] T.-M. Liou and C.-S. Wang, Three-dimensional multidomain lattice Boltzmann grid refinement for passive scalar transport, Phys. Rev. E, 98 (2018), 013306.

[7] G. Di Ilio, D. Chiappini, S. Ubertini, G. Bella and S. Succi, Hybrid lattice Boltzmann method on overlapping grids, Phys. Rev. E, 95 (2017), 013309.

[8] G. Di Ilio, D. Chiappini, S. Ubertini, G. Bella and S. Succi, Fluid flow around NACA 0012 airfoil at low-Reynolds numbers with hybrid lattice Boltzmann method, Comput. Fluids, 166 (2018), 200-208.

[9] K. Hejranfar and E. Ezzatneshan, Implementation of a high-order compact finite-difference lattice Boltzmann method in generalized curvilinear coordinates, J. Comput. Phys., 267 (2014), 28-49.

[10] J. Huang, X.-C. Cai and C. Yang, A fully implicit method for lattice Boltzmann equations, SIAM J. Sci. Comput., 37 (2015), S291-S313.

[11] J. Huang, C. Yao and X.-C. Cai, A nonlinearly preconditioned inexact Newton algorithm for steady state lattice Boltzmann equations, SIAM J. Sci. Comput., 38 (2015), A1701–A1724.

[12] R. Matin, M. K. Misztal, A. Hernandez-Garcia and J. Mathiesen, Finite-element lattice Boltzmann simulations of contact line dynamics, Phys. Rev. E, 97 (2018), 013307.

[13] J. Wu, M. Shen and C. Liu, Study of flow over object problems by a nodal discontinuous Galerkin-lattice Boltzmann method, Phys. Fluids, 30 (2018), 040903.

[14] C. Shu, Y. Wang, C. J. Teo and J. Wu, Development of lattice Boltzmann flux solver for simulation of incompressible flows, Adv. Appl. Math. Mech., 6 (2014), 436-460.

[15] Z. Guo, K. Xu and R. Wang, Discrete unified gas kinetic scheme for all Knudsen number flows: Low-speed isothermal case, Phys. Rev. E, 88 (2013), 033305.

[16] F. Nannelli and S. Succi, The lattice Boltzmann equation on irregular lattices, J. Stat. Phys., 68 (1992), 401-407.

[17] S. Ubertini, G. Bella and S. Succi, Lattice Boltzmann method on unstructured grids: Further developments, Phys. Rev. E, 68 (2003), 016701.

[18] D. V. Patil and K. N. Lakshmisha, Finite volume TVD formulation of lattice Boltzmann simulation on unstructured mesh, J. Comput. Phys., 228 (2009), 5262-5279.

[19] D. V. Patil, Finite volume TVD formulation of lattice Boltzmann simulation on unstructured mesh, Physica A, 392 (2013), 2701-2712.

[20] M. K. Misztal, A. Hernandez-Garcia, R. Matin, H. O. Sorensen and J. Mathiesen, Detailed analysis of the lattice Boltzmann method on unstructured grids, J. Comput. Phys., 297 (2015), 316-339.

[21] W. Li and L.-S. Luo, Finite volume lattice Boltzmann method for nearly incompressible flows on arbitrary unstructured meshes, Commun. Comput. Phys., 20 (2016), 301-324.

[22] W. Li and W. Li, A gas-kinetic BGK scheme for the finite volume lattice Boltzmann method for nearly incompressible flows, Comput. Fluids, 162 (2018), 126-138.

[23] L. Chen and L. Schaefer, Godunov-type upwind flux schemes of the two dimensional finite volume discrete Boltzmann method, Comput. Math. Appl., 75 (2018), 3105-3126.

[24] L. Xu, J. Li and R. Chen, A scalable parallel unstructured finite volume lattice Boltzmann method for three-dimensional incompressible flow simulations, Int. J. Numer. Methods Fluids, 93 (2021), 2744-2762.

[25] L. Xu and R. Chen, Scalable parallel finite volume lattice Boltzmann method for thermal incompressible flows on unstructured grids, Int. J. Heat Mass Transf., 160 (2020), 120156.

[26] L. Xu, R. Chen and X.-C. Cai, Parallel finite-volume discrete Boltzmann method for inviscid

compressible flows on unstructured grids, Phys. Rev. E, 103 (2021), 023306.

[27] F. Cevik and K. Albayrak, A fully implicit finite volume lattice Boltzmann method for turbulent flow, Commun. Comput. Phys., 22 (2017), 393-421.

[28] R. Chen and X.-C. Cai, Parallel one-shot Lagrange-Newton-Krylov-Schwarz algorithms for shape optimization of steady incompressible flows, SIAM J. Sci. Comput., 34 (2012), B584-B605.

[29] H. Yang, E. Prudencio and X.-C. Cai, Fully implicit Lagrange-Newton-Krylov-Schwarz algorithms for boundary control of unsteady incompressible flows, Int. J. Numer. Methods Engng., 91 (2012), 644-665.

[30] A. Barker and X.-C. Cai, Scalable parallel methods for monolithic coupling in fluid-structure interaction with application to blood flow modeling, J. Comput. Phys., 229 (2010), 642-659.

[31] Y. Wu and X.-C. Cai, A fully implicit domain decomposition based ALE framework for three-dimensional fluid-structure interaction with application in blood flow computation, J. Comput. Phys., 258 (2014), 524-537.

[32] F. Kong and X.-C. Cai, Scalability study of an implicit solver for coupled fluid-structure interaction problems on unstructured meshes in 3D, Int. J. High Perform. Comput. Appl., 32 (2016), 207-219.

[33] F. Kong and X.-C. Cai, A scalable nonlinear fluid-structure interaction solver based on a Schwarz preconditioner with isogeometric unstructured coarse spaces in 3D, J. Comput. Phys., 340 (2017), 498-518.

[34] X. Deng, X.-C. Cai and J. Zou, Two-level space-time domain decomposition methods for three-dimensional unsteady inverse source problems, J. Sci. Comput., 67 (2016), 860-882.

[35] F. Kong and X.-C. Cai, A highly scalable multilevel Schwarz method with boundary geometry preserving coarse spaces for 3D elasticity problems on domains with complex geometry, SIAM J. Sci. Comput., 38 (2016), C73-C95.

[36] C. Yang, J. Cao and X.-C. Cai, A fully implicit domain decomposition algorithm for shallow water equations on the cubed-sphere, SIAM J. Sci. Comput., 32 (2010), 418-438.

[37] H. Yang, C. Yang and J. Huang, Variational inequality transport model on the sphere by the active-set reduced-space algorithm, Comput. Phys. Commun., 260 (2021), 107739.

[38] H. Yang, Z. Zhu and J. Kou, A minimum-type nonlinear complementarity simulator with constrained pressure residual (CPR) methods for wormhole propagation in carbonate acidization, J. Comput. Phys., 473 (2023), 111732.

[39] R. Li, H. Yang and C. Yang, Parallel multilevel restricted Schwarz preconditioners for implicit simulation of subsurface flows with Peng-Robinson equation of state, J. Comput. Phys., 422 (2020), 109745.

[40] N. Yang, H. Yang, C. Yang, Multilevel field-split preconditioners with domain decomposition for steady and unsteady flow problems, Comput. Phys. Commun., 280 (2022), 108496.

[41] W. Li and L.-S. Luo, An implicit block LU-SGS finite-volume lattice Boltzmann scheme for steady flows on arbitrary unstructured meshes, J. Comput. Phys., 327 (2016), 503-518.

[42] W. Li, W. Li, P. Song and H. Ji, A conservation-moment-based implicit finite-volume lattice Boltzmann method for steady nearly incompressible flows, J. Comput. Phys., 398 (2019), 108882.

[43] C. Schepke, N. Maillard and P. O. A. Navaux, Parallel lattice Boltzmann method with blocked partitioning, Int. J. Parallel Program., 37 (2009), 593-611.

[44] D. Vidal, R. Roy and F. Bertrand, A parallel workload balanced and memory efficient lattice-Boltzmann algorithm with single unit BGK relaxation time for laminar Newtonian flows, Comput. Fluids, 39 (2010), 1411-1423.

[45] Y. Yu, K. Li, Y. Wang and T. Deng, Parallel computation of entropic lattice Boltzmann method on hybrid CPU-GPU accelerated system, Comput. Fluids, 110 (2015), 114-121.

[46] Z. Liu, X. Chu, X. Lv, H. Meng, S. Shi, W. Han, J. Xu, H. Fu, G. Yang, SunwayLB: Enabling extreme-scale lattice Boltzmann method based computing fluid dynamics simualtions on Sunway TaihuLight, 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS), IEEE (2019), pp. 557-566.

[47] T. Tomczak and R. G. Szafran, Sparse geometries handling in lattice Boltzmann method implementation for graphic processors, IEEE Trans. Parallel Distrib. Syst., 29 (2018), 1865-1878.

[48] T. Tomczak and R. G. Szafran, Propagation pattern for moment representation of the lattice Boltzmann method, IEEE Trans. Parallel Distrib. Syst., 33 (2022), 842-853.

[49] M. Stiebler, J. Tölke and M. Krafczyk, An upwind discretization scheme for the finite volume lattice Boltzmann method, Comput. Fluids, 35 (2006), 814-819.

[50] F. Moukalled, L. Mangani and M. Darwish, The finite volume method in computational fluid dynamics: An advanced introduction with OpenFOAM and Matlab, Springer, Cham, 2016.

[51] Z. Guo, C. Zheng and B. Shi, Non-equilibrium extrapolation method for velocity and pressure boundary conditions in the lattice Boltzmann method, Chinese Phys., 11 (2002), 366-375.

[52] X.-C. Cai, W. D. Gropp, D. E. Keyes, R. G. Melvin and D. P. Young, Parallel Newton-Krylov-Schwarz algorithms for the transonic full potential equation, SIAM J. Sci. Comput., 19 (1998), 246-265.

[53] X.-C. Cai and M. Sarkis, A restricted additive Schwarz preconditioner for general sparse linear systems, SIAM J. Sci. Comput., 21 (1999), 792-797.

[54] F.-N. Hwang and X.-C. Cai, A parallel nonlinear additive Schwarz preconditioned inexact Newton algorithm for incompressible Navier-Stokes equations, J. Comput. Phys., 204 (2005), 666-691.

[55] S. C. Eisenstat and H. F. Walker, Choosing the forcing terms in an inexact Newton method, SIAM J. Sci. Comput., 17 (1996), 16-32.

[56] J. Nocedal and S. Wright, Numerical Optimization, Springer, New York, 2006.

[57] D. Knoll and D. E. Keyes, Jacobian-free Newton-Krylov methods: A survey of approaches and applications, J. Comput. Phys., 193 (2004), 357-397.

[58] T. F. Coleman and J J. Morë, Estimation of sparse Jacobian matrices and graph coloring problems, SIAM J. Numer. Anal., 20 (1983), 187-209.

[59] A. Griewank and A. Walther, Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, SIAM, 2008.

[60] Y. Saad, Iterative Methods for Sparse Linear Systems, second ed., SIAM, 2003.

[61] S. Balay, S. Abhyankar, M. F. Adams, S. Benson, J. Brown, P. Brune, K. Buschelman, E. Constantinescu, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp, V. Hapla, T. Isaac, P. Jolivet, D. Karpeev, D. Kaushik, M. G. Knepley, F. Kong, S. Kruger, D. A. May, L. C. McInnes, R. T. Mills, L. Mitchell, T. Munson, J. E. Roman, K. Rupp, P. Sanan, J. Sarich, B. F. Smith, S. Zampini, H. Zhang, H. Zhang and J. Zhang, PETSc/TAO users manual, Argonne National Laboratory, ANL-21/39 - Revision 3.17, 2022.

[62] G. Karypis and K. Schloegel, ParMETIS: Parallel graph partitioning and sparse matrix ordering library version 4.0, University of Minnesota, 2013.

[63] U. Ghia, K. N. Ghia and C. T. Shin, High-Resolutions for incompressible flow using the Navier-Stokes equations and a multigrid method, J. Comput. Phys., 48 (1982), 387-411.

[64] Z. Chen, C. Shu and D. Tan, Highly accurate simplified lattice Boltzmann method, Phys. Flu-

ids, 30 (2018), 103605.

[65] O. Botella and R. Peyret, Benchmark spectral results on the lid-driven cavity flow, Comput. Fluids, 27 (1998), 421-433.

[66] C. Ma, J. Wu, H. Yu and L. Yang, A high-order implicit-explicit flux reconstruction lattice Boltzmann method for viscous incompressible flows, Comput. Math. Appl., 105 (2022), 13-28.

[67] E. Erturk, T. C. Corke and C. Gökçöl, Numerical soulutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers, Int. J. Numer. Meth. Fluids, 48 (2005), 747-774.

[68] S. Hou, Q. Zou, S. Chen, G. Doolen and A. C. Cogley, Simulation of cavity flow by the lattice Boltzmann method, J. Comput. Phys., 118 (1995), 329-347.

[69] S. P. Vanka, Block-implicit multigrid solution of Navier-Stokes equations in primitive variables, J. Comput. Phys., 65 (1986), 138-158.

[70] L.-S. Lin, Y.-C. Chen and C.-A. Lin, Multi relaxation time lattice Boltzmann simulations of deep lid driven cavity flows at different aspect ratios, Comput. Fluids, 45 (2011), 233-240.

[71] Z.-H. Chai, B.-C. Shi and L. Zheng, Simulating of high Reynolds number flow in two-dimensional lid-driven cavity by multi-relaxation-time lattice Boltzmann method, Chin. Phys., 15 (2006), 1855-1863.

[72] M. Coutanceau and R. Bouard, Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation part 1: Steady flow, J. Fluid Mech., 79 (1977), 231-256.

[73] S. C. R. Dennis and G. Z. Chang, Numerical solutions for steady flow past a circular cylinder at Reynolds number up to 100, J. Fluids Mech., 42 (1970), 471-489.

[74] T. Imamura, K. Suzuki, T. Nakamura and M. Yoshida, Acceleration of steady-state lattice Boltzmann simulations on non-uniform mesh using local time step method, J. Comput. Phys., 202 (2005), 645-663.

[75] R. W. Mei and W. Shyy, On the finite difference-based lattice Boltzmann method in curvilinear coordinates, J. Comput. Phys., 143 (1998), 426-448.

[76] J. Wu and C. Shu, Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications, J. Comput. Phys., 228 (2009), 1963-1979.