Simulation of Cavity Flows by an Implicit Domain Decomposition Algorithm for the Lattice Boltzmann Equations *

Jizu Huang[1,4], Chao Yang[1,2], and Xiao-Chuan Cai[3]
[1] Institute of Software, Chinese Academy of Sciences, Beijing 100190, P. R. China.
[2] State Key Laboratory of Computer Science, Chinese Academy of Sciences, Beijing 100190, P. R. China.
[3] Department of Computer Science, University of Colorado Boulder, Boulder, CO 80309, USA.
[4] Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China.

Abstract In this paper, we develop a fully implicit finite difference scheme for the lattice Boltzmann equations. A parallel, highly scalable Newton–Krylov–RAS algorithm is presented to solve the large sparse nonlinear system of equations arising at each time step. RAS is a restricted additive Schwarz preconditioner built with a cheaper discretization. The accuracy of the proposed method is carefully studied by comparing with other benchmark solutions. We show numerically that the nonlinearly implicit method is scalable on a supercomputer with more than ten thousand processors.

1 Introduction

The 2D steady state lid-driven cavity flow problem is a benchmark problem to test new numerical methods due to its simple geometry and interesting flow behaviors. There are several mathematical models available for simulating this flow, such as the Navier–Stokes (NS) equations and the Boltzmann equations among others. For problems satisfying the continuum assumption, the Boltzmann model and the NS model usually have the same solution in some sense, because the NS model can be derived from the Boltzmann model. But for problems that don't satisfy the continuum assumption, the NS model fails to provide a physically meaningful solution and the Boltzmann model can be viewed as a higher level model. In the past two decades, numerical methods based on the Boltzmann model, such as the lattice Boltzmann equations (LBEs) become increasingly popular [2, 10] in simulating the 2D lid-driven cavity flow. There are extensive numerical experiments carried out with the LBEs [10, 12, 13]. However, all existing approaches are explicit or semi-implicit and

* The work was supported in part by NSFC grants 61170075 and 973 grant 2011CB309701.

the time step size of these approaches is limited by the Courant–Friedrichs–Lewy (CFL) condition, and the numerical solutions obtained by using these methods are less accurate than those of the NS equations.

In this paper, we introduce a fully implicit and parallel Newton–Krylov–RAS algorithm for the LBEs, which is unconditionally stable and the time step size depends only on the accuracy requirement. The method is based on an inexact Newton method whose Jacobian systems are solved with an overlapping RAS preconditioned Krylov subspace method. To reduce the computational cost and improve the scalability of the RAS preconditioner, a first-order discretization is developed just for the preconditioner which is re-computed only once per time step. We report accuracy results and scalability studies on fine meshes and on a supercomputer with more than ten thousands processors.

2 Model problem, discretization, and domain decomposition preconditioning

In this paper, the LBEs [2] are considered

$$\frac{\partial f_\alpha}{\partial t}(\mathbf{x},t)+\mathbf{e}_\alpha\cdot\nabla f_\alpha(\mathbf{x},t)=\Theta_\alpha,\quad \alpha=0,1,\cdots,8,\quad \mathbf{x}\in\Omega,\quad t\in(0,T),\tag{1}$$

where f_α is the particle distribution function, $\mathbf{e}_\alpha=(e_{\alpha1},e_{\alpha2})$ is the discrete particle velocity, Θ_α is the collision operator, $\Omega=(0,1)^2\in R^2$ is the computational domain, and $(0,T)$ is the time interval. The macroscopic density ρ and the macroscopic velocity $\mathbf{u}=(u_1,u_2)$ of the fluid are respectively induced from the particle distribution function by

$$\rho=\sum_\alpha f_\alpha,\qquad \mathbf{u}=\frac{1}{\rho}\sum_\alpha f_\alpha\mathbf{e}_\alpha.\tag{2}$$

The collision operator is defined as $\Theta_\alpha=-\frac{1}{\tau}(f_\alpha(\mathbf{x},t)-f_\alpha^{(eq)}(\mathbf{x},t))$, where $\tau=c_s^{-2}\nu$ is the relaxation time of the fluid and $f_\alpha^{(eq)}$ is the local equilibrium distribution function (EDF) defined as

$$f_\alpha^{(eq)}=w_\alpha\rho\left[1+\frac{1}{c_s^2}\mathbf{e}_\alpha\cdot\mathbf{u}+\frac{1}{2c_s^4}(\mathbf{e}_\alpha\cdot\mathbf{u})^2-\frac{1}{2c_s^2}|\mathbf{u}|^2\right].\tag{3}$$

Here ν is the shear viscosity, $c_s=1/\sqrt{3}$ is the sound speed, $|\mathbf{u}|=(u_1^2+u_2^2)^{1/2}$, and the discrete velocities are given by $\mathbf{e}_0=(0,0)$, and $\mathbf{e}_\alpha=\lambda_\alpha(\cos\theta_\alpha,\sin\theta_\alpha)$, with $\lambda_\alpha=1$, $\theta_\alpha=(\alpha-1)\pi/2$ for $\alpha=1,2,3,4$ and $\lambda_\alpha=\sqrt{2}$, $\theta_\alpha=(\alpha-5)\pi/2+\pi/4$ for $\alpha=5,6,7,8$. The weighting factors are defined as $w_0=4/9$, $w_\alpha=1/9$ for $\alpha=1,2,3,4$ and $w_\alpha=1/36$ for $\alpha=5,6,7,8$.

Assume $(0,T)$ is divided into time intervals, where n is the time step index. A fully implicit backward Euler scheme is used to discretize the temporal derivative. Then we obtain a semi-discretized system for (1) as follows

$$\frac{f_\alpha^{n+1}-f_\alpha^n}{\Delta t_{n+1}}+\mathbf{e}_\alpha\cdot\nabla f_\alpha^{n+1}=\Theta_\alpha^{n+1}, \tag{4}$$

where the time step size is $\Delta t_{n+1}=t_{n+1}-t_n$, $f_\alpha^n(\mathbf{x})\approx f_\alpha(\mathbf{x},t^n)$, and $\Theta_\alpha^{n+1}\approx\Theta_\alpha(\mathbf{x},t^{n+1})$. If $e_{\alpha k}\neq 0$, we implement a family of fully implicit finite difference schemes originally proposed in [10] for an explicit method to discretize the spatial derivative $\frac{\partial f_\alpha}{\partial x_k}$. We partition the domain Ω to a uniform $N\times N$ mesh with mesh size $h=1/(N-1)$ and mesh points $(x_1^i,x_2^j), i,j=0,1,...,N-1$. Let us define a scheme $\frac{\partial f_\alpha}{\partial x_k^i}|_m$ in the family as

$$\frac{\partial f_\alpha}{\partial x_k^i}\Big|_m=\varepsilon\frac{\partial f_\alpha}{\partial x_k^i}\Big|_u+(1-\varepsilon)\frac{\partial f_\alpha}{\partial x_k^i}\Big|_c,\qquad k=1,2,\quad 1\le i\le N-2, \tag{5}$$

where $0\le\varepsilon\le 1$ is a control parameter that determines how much upwinding is added,

$$\frac{\partial f_\alpha}{\partial x_k^i}\Big|_c=\frac{1}{2h}\big[f_\alpha(x_k^{i+1},\cdot)-f_\alpha(x_k^{i-1},\cdot)\big],$$

and

$$\frac{\partial f_\alpha}{\partial x_k^i}\Big|_u=\begin{cases}\dfrac{e_{\alpha k}}{2h}\big[3f_\alpha(x_k^i,\cdot)-4f_\alpha(x_k^{i-e_{\alpha k}},\cdot)+f_\alpha(x_k^{i-2e_{\alpha k}},\cdot)\big] & \text{if}\quad 2\le i\le N-3,\\ \dfrac{e_{\alpha k}}{h}\big[f_\alpha(x_k^i,\cdot)-f_\alpha(x_k^{i-e_{\alpha k}},\cdot)\big] & \text{if}\quad i=1,\text{or } i=N-2.\end{cases}$$

Theoretically, the scheme is second-order in the interior of the domain and first-order near the boundary, but for our test cases, the scheme is effectively second-order. We also introduce a cheaper first-order upwinding scheme $\frac{\partial f_\alpha}{\partial x_k^i}=\frac{e_{\alpha k}}{h}\big[f_\alpha(x_k^i,\cdot)-f_\alpha(x_k^{i-e_{\alpha k}},\cdot)\big]$ to construct an efficient preconditioner for the scheme (5).

The initial condition is set to be the EDF, i.e. $f_\alpha(\mathbf{x},0)=f_\alpha^{(eq)}(\mathbf{x},0)$. The boundary conditions are obtained by a nonequilibrium extrapolation method [14]. Assume that $\mathbf{x}_b$ is a mesh point on the boundary of the domain, and $\mathbf{x}_{nb}$ is the nearest neighboring mesh point of $\mathbf{x}_b$ in the interior of the domain. According to the nonequilibrium extrapolation method, the particle distribution function at $\mathbf{x}_b$ is set to be

$$f_\alpha(\mathbf{x}_b)=f_\alpha^{(eq)}(\mathbf{x}_b)+[f_\alpha(\mathbf{x}_{nb})-f_\alpha^{(eq)}(\mathbf{x}_{nb})]. \tag{6}$$

After the discretization, a system of nonlinear algebraic equations

$$\mathscr{F}^{n+1}(\mathbf{X}^{n+1}):=\frac{\mathbf{X}^{n+1}-\mathbf{X}^n}{\Delta t_{n+1}}+\mathscr{G}^{n+1}(\mathbf{X}^{n+1})=0,\quad n=0,1,\dots \tag{7}$$

is obtained and needs to be solved at each time step. Here $\mathscr{G}^{n+1}$ is dependent on the spatial discretization and the collision term. We employ a Newton–Krylov–Schwarz (NKS) [5, 6] type algorithm to solve (7). At each Newton iteration, a Jacobian system is analytically computed and approximately solved by using a Krylov subspace

method

$$J^{n+1}\mathbf{S}^{n+1} = -\mathscr{F}^{n+1}(\mathbf{X}^{n+1}), \tag{8}$$

where the Jacobian matrix $J^{n+1} = (\mathscr{F}^{n+1})'(\mathbf{X}^{n+1})$ and $\mathbf{S}^{n+1}$ is the search direction of the Newton method. A restarted GMRES (20) method is applied to approximately solve the right-preconditioned system

$$J^{n+1}(M^{n+1})^{-1}(M^{n+1}\mathbf{S}^{n+1}) = -\mathscr{F}^{n+1}(\mathbf{X}^{n+1}), \tag{9}$$

where M^{n+1} is the restricted additive Schwarz (RAS) preconditioner defined in [7]. The initial guess for the Newton iteration is chosen as the final solution from the previous time step.

3 Numerical experiments

We implement the new algorithm described in the previous section based on PETSc [1]. A steady state driven cavity flow in 2D is carefully studied in this section. The numerical tests are carried out on a supercomputer Tianhe-2, which tops the Top-500 list as of June, 2013. The computing nodes of Tianhe-2 are interconnected via a proprietary high performance network. And there are two 12-core Intel Ivy Bridge Xeon CPUs and 24GB local memory in each node. In the numerical experiments we use all 24 CPU cores in each node and assign one subdomain to each core.

In the 2D driven cavity flow problem, we assume the top boundary of the cavity moves from right to left with a constant velocity $U_0 = -0.1$ while the other three boundaries are fixed. The initial condition of macroscopic variables $\rho = 1.0$ and $\mathbf{u} = (0,0)$ in the cavity. The Reynolds number is defined as $Re = U_0 H/\nu$ with $H = 1.0$. In our simulations, Re is chosen to be 100, 1000, 3200, 5000, 7500, and 10000.

Simulating this flow by solving the NS equations is a popular approach [8, 9, 11], in which the presence of singularities at the corners is a well-known difficulty. At the corners (0,1) and (1,1), the pressure and the vorticity are unbounded, and at the corners (1,0) and (0,0) the second derivatives of the pressure and vorticity are unbounded. To improve the accuracy of the solution at the corners, Deng *et al.* [8] perform a Richardson extrapolation of solutions obtained by a finite volume method. In [3], a spectral method is developed to remove the pollution of the singularities. To check the accuracy of the discretization, we simulate the flow at $Re = 100$ with different mesh sizes. The maximum of u_1 on the vertical line $x_1 = 0.5$ is denoted as $u_{1,\max}$ and its location $x_{2,\max}$. The minimum and maximum of u_2 on the horizontal line $x_2 = 0.5$ are, respectively, denoted as $u_{2,\min}$ and $u_{2,\max}$; their locations are, respectively, denoted as $x_{1,\min}$ and $x_{1,\max}$. Table 1 shows the values of these extremum and previously published results obtained by the NS equations. Our results are in agreement with those of [4, 9], but less accurate than the results of [3, 8]. In [4, 9], second-order schemes are used to solve the NS equations. In [3, 8], higher order schemes are given to remove the pollution from the corner singularities.

Table 1 $Re = 100$, extrema of the velocity through the centerlines of the cavity.

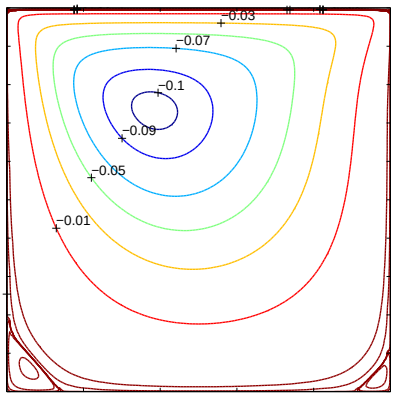
Reference	N	$u_{1,\max}$	$x_{2,\max}$	$u_{2,\max}$	$x_{1,\max}$	$u_{2,\min}$	$x_{1,\min}$
Present	65	0.2075	0.4531	0.1674	0.7656	-0.2471	0.1875
Present	97	0.2098	0.4583	0.1711	0.7604	-0.2492	0.1875
Present	129	0.2107	0.4609	0.1725	0.7656	-0.2500	0.1875
Present	161	0.2111	0.4562	0.1733	0.7625	-0.2504	0.1875
Present	257	0.2117	0.4609	0.1742	0.7617	-0.2510	0.1875
Ref. [3]	96	0.2140	0.4581	0.1796	0.7630	-0.2538	0.1896
Ref. [8]	64	0.2132	–	0.1790	–	-0.2534	–
Ref. [9]	129	0.2109	0.4531	0.1753	0.7656	-0.2453	0.1953
Ref. [4]	129	0.2106	0.4531	0.1786	0.7656	-0.2521	0.1875

The streamline contours for the cavity flow configurations with Re increasing from 100 to 10000 are shown in Fig. 1. We observe that the flow structures are in good agreement with the benchmark results obtained by Ghia *et al*. [9]. These plots show clearly the effect of Re on the flow pattern. For flows with $Re \leq 1000$, only three vortices appear in the cavity; a primary one near the center and a pair of secondary ones in the corners of the cavity. At $Re = 3200$, a third secondary vortex is seen in the upper right corner. At $Re = 5000$, a tertiary vortex appears in the lower left corner. Furthermore, another tertiary vortex appears in the lower right corner as $Re \geq 7500$.

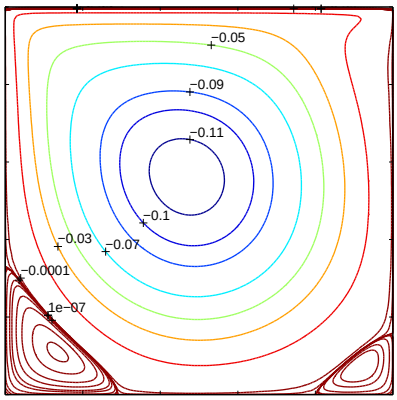
To show the parallel scalability of the implicit method, we consider a 4096×4096 uniform mesh. We use a fixed time step size $\Delta t = 0.0244$ and run the code for 10 time steps. We test two overlapping factors $\delta = h, 2h$ with different number of processors. We compare the point-block LU subdomain solver and the point-block ILU(l) solver. Here l is the fill-in level for the incomplete LU factorization. The point-block size is 9×9. We set the fill-in levels $l = 0, 1, 2, 3$. The numbers of linear and nonlinear iterations are reported in Table 2. The number of linear iterations grows slowly with the increase of the number of processors. Large overlap or larger fill-in helps reduce the total number of linear iterations. The compute time of both an explicit method [10] and the implicit method with different subdomain solvers is shown in Fig. 2. The optimal compute time can be obtained with fill-in level $l = 1$, which is less than that of the explicit method. Excellent speedup is obtained from 512 processors to 16384 processors. From the figure we see that ILU is faster in terms of the total compute time than LU.

We also do some weak scaling tests of proposed implicit method with local solvers (LU or ILU(1)). It is observed that the method does not reach the ideal performance, because the number of GMRES iterations increases as more processor cores are used. We believe that coarse level corrections in the additive Schwarz preconditioner can improve the weak scaling performance of the fully implicit solver and plan to study this issue in the future. But, due to the page limit, the results are not given in the paper.

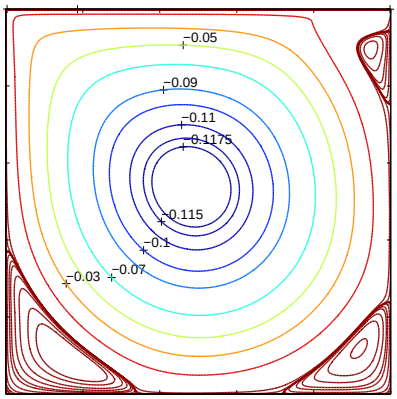
(a) $Re = 100$, 128×128 mesh



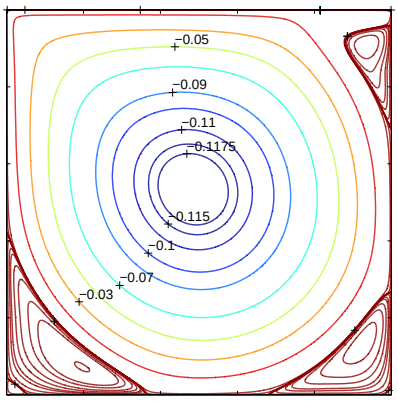
(b) $Re = 1000$, 128×128 mesh



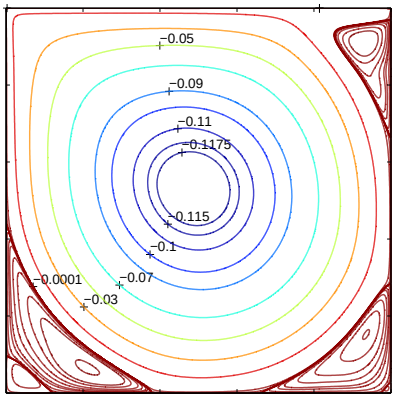
(c) $Re = 3200$, 256×256 mesh



(d) $Re = 5000$, 512×512 mesh



(e) $Re = 7500$, 768×768 mesh



(f) $Re = 10000$, 768×768 mesh

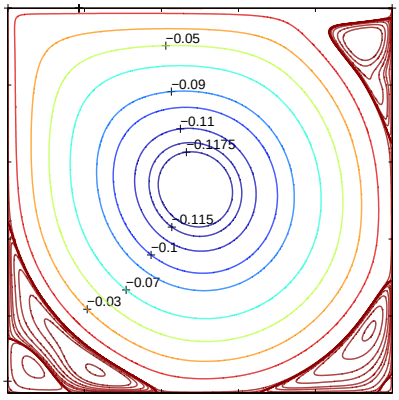


Fig. 1 Streamline patterns for the primary, secondary, and additional corner vortices.

Table 2 Test results using different overlapping factors and number of processors, 4096×4096 mesh (# of unknowns $= 150,994,944$), $t_0 = 0$, time step size $\Delta t = 0.0244$, CFL $= 100$, $Re = 3200$, 10 time steps.

δ	N_p	Newton(avg.)					GMRES/Newton				
		LU	ILU(0)	ILU(1)	ILU(2)	ILU(3)	LU	ILU(0)	ILU(1)	ILU(2)	ILU(3)
h	512	4.7	6.1	6	6	6	20.68	26.62	21.93	21.95	21.85
	1024	4.7	6.1	6	6	6	21.83	27.21	22.92	22.98	22.90
	2048	4.7	6.1	6	6	6	22.62	27.93	23.42	23.52	23.55
	4096	4.7	6.1	6	6	6	24.02	28.82	24.45	24.63	24.55
	8192	4.7	6.1	6	6	6	26.11	30.43	25.73	25.80	25.73
	16384	4.7	6.1	6	6	6	27.60	32.08	26.98	27.07	26.98
$2h$	512	6	6.1	6	6	6	20.65	26.16	20.85	20.67	20.63
	1024	6	6.1	6	6	6	21.78	26.82	21.50	21.60	21.60
	2048	6	6.1	6	6	6	22.35	27.43	22.03	22.07	22.12
	4096	6	6.1	6	6	6	23.85	28.10	22.83	22.97	23.10
	8192	6	6.1	6	6	6	25.47	29.43	23.78	23.77	23.78
	16384	6	6.1	6	6	6	26.85	30.97	24.90	24.68	25.20

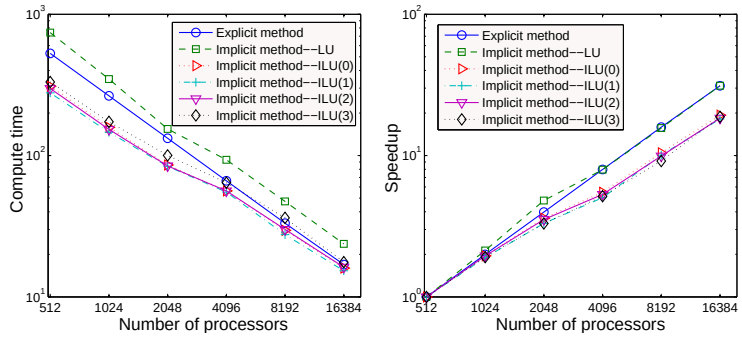
4 Conclusions

We developed a parallel, highly scalable fully implicit method for the LBEs. The accuracy of the method is comparable with that of the NS equations. The fully implicit method exhibits an excellent speedup with up to 150 million unknowns on a supercomputer with up to 16384 processors. Without the CFL limit, the fully implicit method can be used with a suitable adaptive time stepping method that increases the time step size as the solution approaches steady state. Because of the page limit, the discussion related to adaptive time stepping and comparisons with other methods will be presented in a separate report.

References

1. S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. Knepley, L. C. Mcinnes, B. F. Smith, and H. Zhang, *PETSc Users Manual*. Argonne National Laboratory, Argonne, 2013.
2. R. Benzi, S. Succi, and M. Vergassola, *The Lattice Boltzmann equation: theory and applications*. Phys. Rep., 222 (1992), 145-197.
3. O. Botella and R. Peyret, *Benchmark spectral results on the lid-driven cavity flow*. Comput. & Fluids, 27 (1998), 421-433.
4. C. H. Bruneau and C. Jouron, *An efficient scheme for solving steady incompressible Navier-Stokes equations*. J. Comput. Phys., 89 (1990), 389-413.
5. 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.
6. X.-C. Cai, W. D. Gropp, D. E. Keyes, and M. D. Tidriri, *Newton-Krylov-Schwarz methods in CFD*, in: R. Rannacher (Ed.), Notes in Numerical Fluid Mechanics: Proceedings of the In-

(a) $\delta = h$, compute time comparison (b) $\delta = h$, speedup curve



(a) $\delta = h$, compute time comparison (b) $\delta = h$, speedup curve

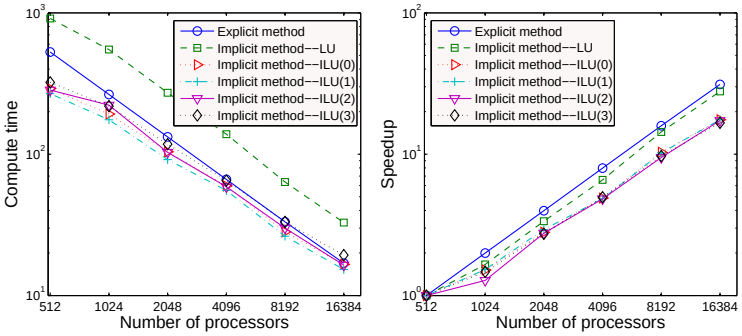


Fig. 2 Compute time and speedup comparison on a 4096×4096 mesh with 512, 1024, 2048, 4096, 8192, and 16384 processors. Implicit method with different subdomain solvers: 10 time steps with a fixed time step size $\Delta t = 0.0244$. Explicit method [10]: 20000 time steps with a fixed CFL=0.05.

ternational Workshop on the Navier-Stokes Equations, Vieweg Verlag, Braunschweig, 1994, 123-135.

7. X.-C. Cai and M. Sarkis, *A restricted additive Schwarz preconditioner for general sparse linear systems*. SIAM J. Sci. Comput., 21 (1999), 792-797.
8. G. B. Deng, J. Piquet, P. Queutey, and M. Visonneau, *Incompressible flow calculations with a consistent physical interpolation finite volume approach*. Comput. & Fluids, 23 (1994), 1029-1047.
9. U. Ghia, K. N. Ghia, and C. T. Shin, *High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method*. J. Comput. Phys., 48 (1982), 387-411.
10. Z. L. Guo and T. S. Zhao, *Explicit finite-difference lattice Boltzmann method for curvilinear coordinates*. Phys. Rev. E, 67 (2003), 066709(12p).
11. P. Luchini, *Higher-order difference approximations of the Navier-Stokes equations*. Int. J. Numer. Methods Fluids, 12 (1991), 491-506.
12. R. Mei and W. Shyy, *On the finite difference-based lattice Boltzmann method in curvilinear coordinates*. J. Comput. Phys., 143 (1998), 426-448.
13. H. W. Xi, G. W. Peng, and S.-H. Chou, *Finite-volume lattice Boltzmann method*. Phys. Rev. E, 59 (1999), 6202-6265.
14. Z. L. Guo, C. G. Zheng, and B. C. Shi, *Non-equilibrium extrapolation method for velocity and boundary conditions in the lattice Boltzmann method*. Chin. Phys., 11(4) (2002), 0366-0374.