Lattice Boltzmann method on composite grids

Ching-Long Lin* and Yong G. Lai†
Department of Mechanical Engineering and Iowa Institute of Hydraulic Research, The University of Iowa, Iowa City, Iowa 52242-1527
(Received 10 March 2000)

A composite block-structured lattice Boltzmann method is proposed for the simulation of two-dimensional incompressible fluid flows. The grid structure is composed of a coarse base grid and one or several fine grid(s). The former covers the entire physical domain; the latter are placed at regions where local grid refinement is desirable. The simulation is first carried out on the base grid level at a smaller relaxation time, allowing a rapid propagation of boundary information throughout the entire domain. Thus large-scale flow features can be resolved efficiently at a relatively low cost. At a later time, fine grid variables are initiated. The dependent variables on both grid levels are, then, advanced in time simultaneously with the fine grid boundary conditions obtained from the base grid solution at the grid interface. As a demonstration, the lid-driven cavity flow is selected for study. The results show good agreement with benchmark numerical data and those calculated from the finite-volume U²RANS model. The proposed method is able to produce accurate solutions on fine grids, with a considerable saving in CPU time.

PACS number(s): 47.10.+g, 47.11.+j, 02.70.−c

I. INTRODUCTION

The lattice Boltzmann equation (LBE) method has received great attention from the community of computational fluid dynamics because of its capability of simulating multiphase flow, interfacial flow phenomena, and flow through porous media, among others [1]. Several issues, however, remain to be addressed. One of them is the extension of the LBE method from regular grids to irregular grids. With irregular grids, grid refinement can be applied to important flow regions, and flow structures along curved boundaries can be better resolved. Several efforts have been made to overcome this issue, and significant progresses have been achieved in recent years.

For instance, He and Luo [2,3] demonstrated that the LBE is a discretized form of the continuous Boltzmann equation, leading to the important concept that the discretization of physical space is not coupled to the discretization of momentum space. An immediate consequence is that arbitrary grids can be used in the LBE method. Based on the above concept, He *et al.* [4] proposed an interpolation-supplemented LBE (ISLBE) method to simulate a two-dimensional channel flow with sudden expansion on a nonuniform grid. Later, He and Doolen [5,6] applied the ISLBE method to simulate vortex shedding behind a circular cylinder on a curvilinear coordinate system. This method adds a new interpolation step between the streaming and relaxation steps in the conventional LBE method, and retains the locality property of the two steps. At the same time, Cao *et al.* [7] indicated that the LBE method is a special finite-difference discretization of the kinetic equation of the discrete velocity distribution function. The implication is that by introducing other standard finite-difference discretization methods, the application of nonuniform grids and semi-implicit collision schemes to the conventional LBE method is possible. On this basis, Mei and Shyy [8] solved the LBE in a generalized body-fitted coordinate system, demonstrating that the LBE method is capable of solving fluid flow problems in complex geometries. The finite-volume LBE (FVLBE) method was also proposed by Succi *et al.* [9] for a simulation of fluid flows in complex geometries. Xi *et al.* [10,11] proposed a cell-vertex FVLBE method for simulation of two- and three-dimensional flows. No numerical diffusion resulted from their method.

The regular grid structure, however, tends to have a better numerical stability property. Thus an alternative strategy for dealing with curved boundaries is to devise a boundary-fitted condition for nodes adjacent to these boundaries. Based on the work of Filippova and Hänel [12], Mei *et al.* [13] proposed a second-order accurate treatment, which has been tested on several benchmark cases. Along this line of thought, the capability of local grid refinement has to be developed to make a simulation on regular lattice computationally economical. Filippova and Hänel [12] proposed a local grid refinement method which allows two-way interaction at the post-collision stage after rescaling the discrete distribution functions on both grid levels. In this paper, we propose a composite LBE method, which allows one-way interaction at the post-streaming stage without rescaling the discrete distribution functions. The proposed method uses a multigrid architecture. The coarse grid covers the entire physical domain, and is nested with one or several fine grid(s) as in typical grid-nesting techniques, such as that of Sullivan *et al.* [14]. Different grid levels operate at different length and time scales. We will demonstrate that more accurate solutions are obtained on fine grids, with a considerable saving in CPU time.

The paper is organized as follows. In Sec. II, a two-dimensional square lattice model is briefly reviewed. The effect of characteristic length and time scales of the square lattice on numerical solutions and CPU time is discussed in Sec. III. In Sec. IV, a composite block-structured lattice and a solution procedure are presented. The results are shown

*Email address: ching-long-lin@uiowa.edu
†Email address: yong-gen-lai@uiowa.edu

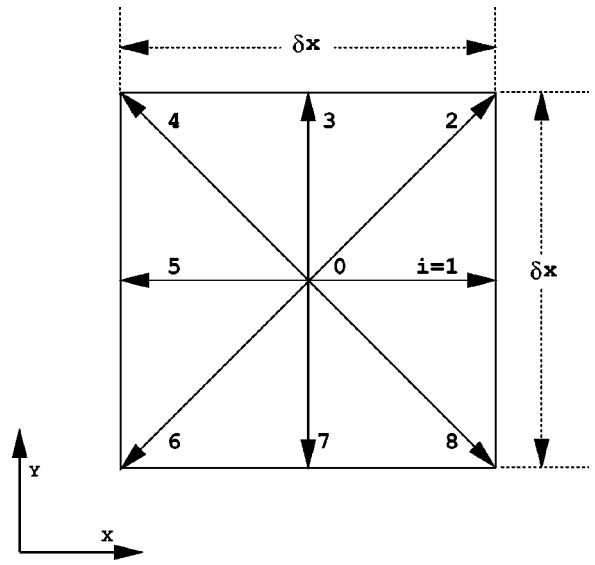

FIG. 1. Two-dimensional square lattice model with a length scale δx.

and discussed in Sec. V. Concluding remarks are given in Sec. VI.

II. SQUARE LATTICE MODEL

The Boltzmann equation with the Bhatnagar-Gross-Krook (BGK) collision operator [15] reads

$$\frac{\partial f}{\partial t}+\vec{e}\cdot\vec{\nabla}f=-\frac{1}{\tau}(f-f^{eq}), \tag{1}$$

where f is the single-particle distribution function, $\vec{e}$ is the microscopic velocity, $\vec{\nabla}f$ is the gradient of the function f, τ is the relaxation time due to collision, and f^{eq} is the Boltzmann-Maxwellian distribution function. Using the lattice Boltzmann nine-bit model in two dimensions (Fig. 1), the above equation can be discretized as

$$f_i(\vec{x}+\vec{e}_i\delta t,t+\delta t)=f_i(\vec{x},t)-\frac{1}{\tau}[f_i(\vec{x},t)-f_i^{eq}(\vec{x},t)]. \tag{2}$$

Here the right-hand side of the equation is known as the collision process, and the left-hand side as the streaming process. The discrete velocities $\vec{e}_i$ are expressed as (refer to Fig. 1 for the direction represented by the subscript i)

$$\vec{e}_i=\begin{cases} c(0,0), & i=0 \\ c(\cos\theta_i,\sin\theta_i), & \theta_i=(i-1)\pi/4, \quad i=1,3,5,7 \\ \sqrt{2}c(\cos\theta_i,\sin\theta_i), & \theta_i=(i-1)\pi/4, \quad i=2,4,6,8, \end{cases} \tag{3}$$

where $c\equiv\delta x/\delta t$, δx is the adjustable lattice size, and δt the time step. The equilibrium distribution function f_i^{eq} is defined as

$$f_i^{eq}=w_i\rho\left[1+3\frac{(\vec{e}_i\cdot\vec{u})}{c^2}+\frac{9}{2}\frac{(\vec{e}_i\cdot\vec{u})^2}{c^4}-\frac{3}{2}\frac{(\vec{u}\cdot\vec{u})}{c^2}\right], \tag{4}$$

with the weights $w_0=4/9$, $w_1=w_3=w_5=w_7=1/9$, and $w_2=w_4=w_6=w_8=1/36$ [2]. The macroscopic density ρ and velocity vector $\vec{u}$ are related to the distribution function by

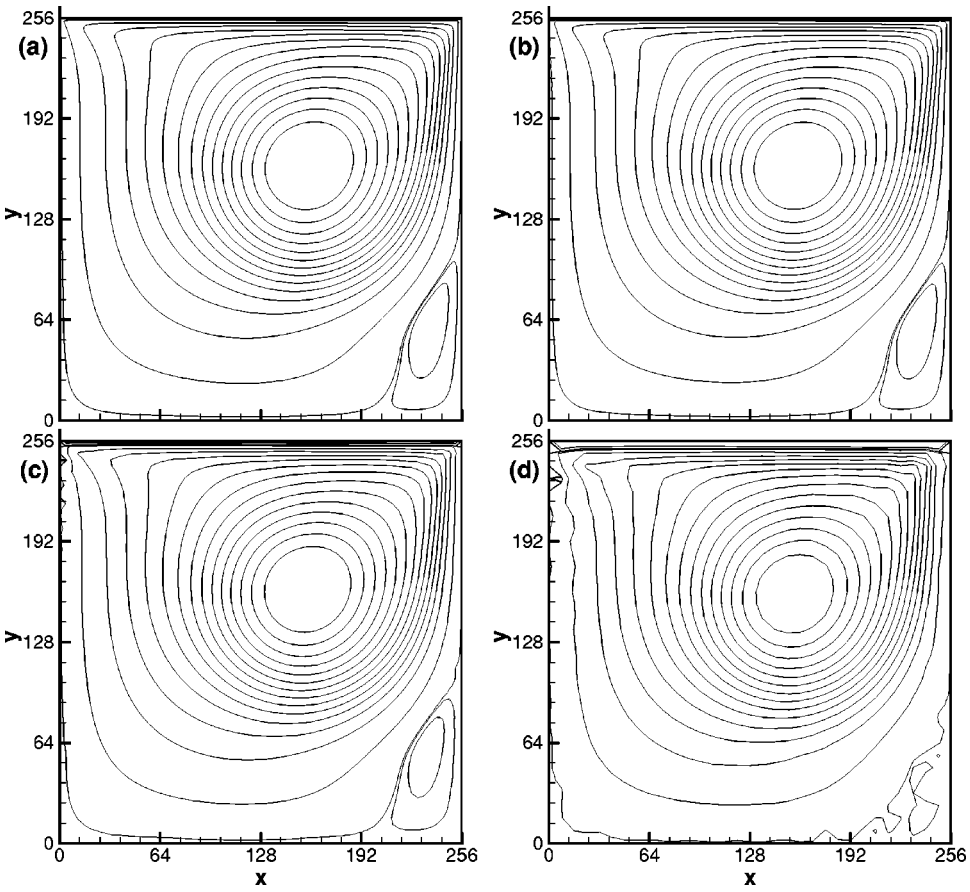

FIG. 2. Contours of the stream function at a time 20 000 obtained with grid sizes (a) 257×257, (b) 129×129, (c) 65×65, and (d) 33×33.

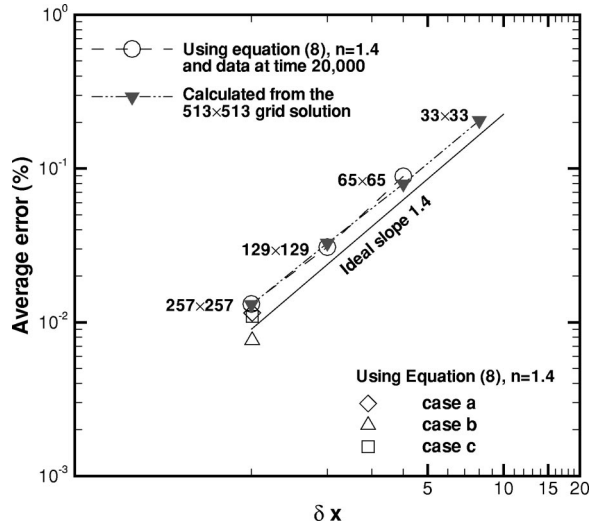

FIG. 3. Normalized average errors for grids of 65×65, 129×129, and 257×257 shown in Fig. 2, and for cases *a*, *b*, and *c* listed in Table I. The solid line is the ideal slope 1.4.

$$\sum_{i=0}^{8} f_i=\rho, \quad \sum_{i=1}^{8} f_i\vec{e}_i=\rho\vec{u}. \tag{5}$$

Using the Chapman-Enskog expansion [17,18], the above equations can recover the Navier-Stokes equations to the second order of accuracy, with the kinematic viscosity ν given by

$$\nu=\frac{2\tau-1}{6}\frac{\delta x^2}{\delta t} \tag{6}$$

The pressure can be calculated from $p=c_s^2\rho$, with the speed of sound $c_s=c/\sqrt{3}$. The above two-dimensional nine-bit square lattice model is adopted in this study.

III. EFFECT OF LATTICE LENGTH AND TIME SCALES

In the classical BGK model, the square lattice length (δx) and time (δt) scales are taken to be unity. For instance, Fig. 2(a) exhibits contours of the stream function for a developing lid-driven cavity flow at a Reynolds number $\mathrm{Re}=1000$ and time 20 000 (i.e. 20 000 iterations); the grid points are 257×257, with $\delta x=\delta t=1$. The lid velocity at the top wall is 0.1, which is the same value as used by Hou *et al.* [17] on a 256×256 grid. Note that the flow is not fully developed in order to illustrate the time accurate aspect of the method.

By increasing the lattice characteristic scales to $\delta x=\delta t=2$, the number of iterations required to reach the same time as above is reduced by a factor of 2. In addition, since only 129×129 grid points with $\delta x=2$ are needed to cover the same physical domain, the number of grid points is reduced by a factor of 4 (in three dimensions, the factor becomes 8), and so are the numbers of streaming and collision operations per time step. Figure 2(b) shows contours of stream function for this grid at time 20 000, using the same contour levels as before. Note that this case requires only 10 000 iterations since $\delta t=2$. Both grids produce almost the same results. With $\delta x=\delta t=4$, on a 65×65 grid, most of large- and small-scale flow structures are still discernible, as shown in Fig. 2(c). On a very coarse lattice 33×33, Fig. 2(d) shows that large-scale flow features are captured, while the weak small vortex at the lower right corner is not resolved. The lattice model fails to represent macroscopic motions of a fluid whose characteristic length scale is of the order of the lattice unit size. That is, the continuum of fluid motions holds only when the hydrodynamic length scale in a flow is much larger than the lattice scale δx. In Fig. 2(d), vortices exist on two different scales, but the scale of the lattice unit is only small enough for the central vortex.

Two approaches can be applied to estimate the overall order of accuracy of the method, when the exact solution is not available. One approach uses the following formula (Ferziger and Perić [16]), which requires numerical solutions on three grid levels with grid spacings δx, $\delta x/2$, and $\delta x/4$:

$$n\approx\frac{\ln\left[\dfrac{\sum|\phi_{\delta x/2}-\phi_{\delta x}|/N}{\sum|\phi_{\delta x/4}-\phi_{\delta x/2}|/N}\right]}{\ln 2} \tag{7}$$

$\phi_{\delta x}$ represents any dependent variable, such as velocity components u and v, with grid spacing δx. N is the number of grid points, with which ϕ values are compared. Here we use u and v data obtained with grids 65×65, 129×129, and 257×257, which resolve the lower right corner vortex. The

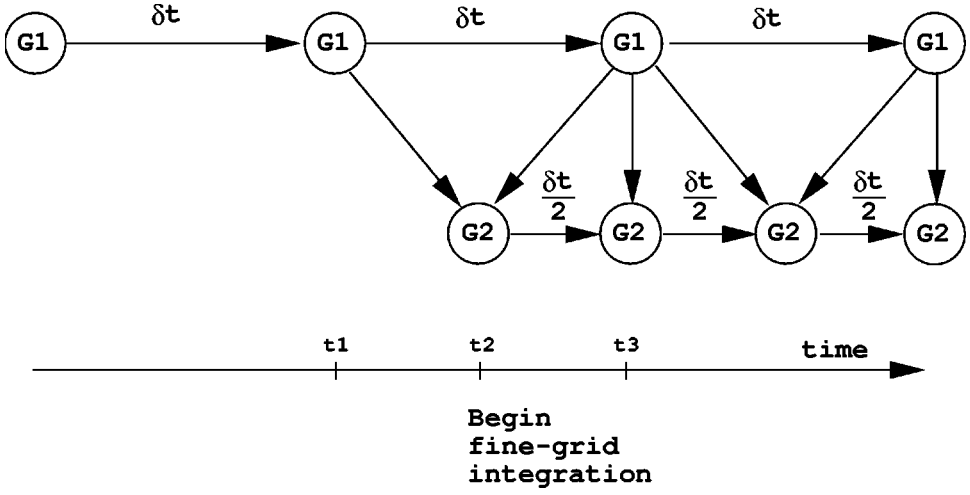

FIG. 4. Integration sequence of the composite grid.

order of accuracy estimated is $n=1.4$. With the assumption of the 1.4-order convergence of the method, we can further estimate the average discretization error ϵ for $\phi_{\delta x/2}$ via Richardson extrapolation [16]. That is,

$$\epsilon \approx \frac{\sum |\phi_{\delta x/2} - \phi_{\delta x}|/N}{2^n - 1}. \tag{8}$$

The estimated error is further normalized using the formula $\epsilon/(\Sigma|\phi_{\delta x/2}|/N+\epsilon)$, where the denominator represents the estimated ''exact'' solution. Figure 3 shows the normalized average errors on three different grid levels. The average errors are 1.3%, 3.1%, and 8.9% of the exact solution for grids 257×257, 129×129, and 65×65, respectively. They closely follow the ideal slope 1.4.

The other approach adopted by Hou *et al.* [17] uses a very fine grid solution as the ''exact'' solution for the calculation of errors in other coarser grid solutions. To verify the above estimation using this approach, we repeat the simulation with a very fine grid 513×513 and $\delta x=\delta t=0.5$. Figure 3 shows that the error curve generated by this approach nearly parallels the curve of an ideal slope 1.4, and almost overlaps the curve generated by the previous method. Therefore, by doubling the grid spacing, the discretization error roughly increases by a factor of $2^{1.4}=2.64$. On the other hand, the number of arithmetic operations is reduced by a factor of 8 in two dimensions (16 in three dimensions).

The above consideration has an immediate consequence: a considerable saving in CPU time can be achieved if coarser grids are used in regions where flow structures are smooth or not important. For instance, all the above cases are run on HP-UX 9000/785 workstations. The Fortran library routine DTIME is used to record the elapsed execution time for each case. The above four cases (with lattice sizes of 257×257, 129×129, 65×65, and 33×33) take 98.02, 4.961, 0.4643, and 0.0658 CPU seconds, respectively. Normalization by 0.0658 yields 1490, 75, 7, and 1, respectively. It verifies that an appreciable amount of CPU time can be saved at a moderate loss of accuracy of numerical results.

In the above calculation, the use of $\delta x=\delta t$ gives a constant $c=\delta x/\delta t=1$, and, as a result, the same discrete velocities $\vec{e}_i$ [Eq. (3)] and equilibrium distribution function [Eq. (4)] are used for all of the above cases. Since kinematic viscosity ν is determined by Reynolds number $\mathrm{Re}=U_{lid}L_{lid}/\nu$, where U_{lid} and L_{lid} are the lid velocity and length, ν remains constant for all cases. The only difference among these cases lies in the relaxation time τ, which appears in Eqs. (2) and (6). With increasing δx and δt, the relaxation time τ decreases, leading to a larger change of f_i for each collision step. An observation of interest is that f_i is interchangeable after the streaming step among the four cases, in spite of their different characteristic scales and accuracy.

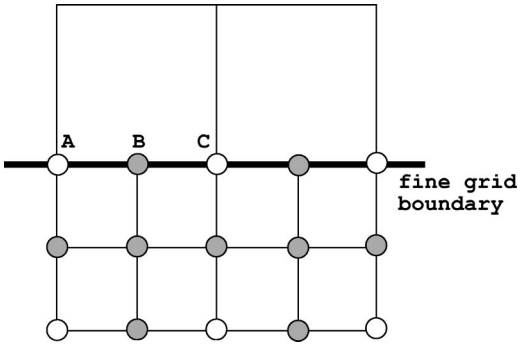

FIG. 5. Schematic of the intergrid communication.

IV. COMPOSITE BLOCK-STRUCTURED LATTICE

With the above observation, it is possible to construct a composite block-structured lattice method so that accurate numerical solutions can be obtained at designated areas with much less CPU time. The objective of the composite lattice is to place lattices of smaller characteristic length and time scales at locations where fine-scale eddies exist, akin to the concept of grid nesting [14]. The grid structure is composed of a coarse base grid and one or several fine grid(s). The former covers the entire physical domain; the latter covers only subdomains. The characteristic length and time scales (δx and δt) of the base grid are twice those of the fine grid.

The integration sequence of the composite grid is illustrated as in Fig. 4. Before time $t3$, flow structures on the base grid level (denoted by $G1$) are solved at a faster rate with a larger time step δt. The simulation on the fine grid (denoted by $G2$) starts at a later time $t2$ with a smaller time step $\delta t/2$. At time $t2$, the fine grid boundary conditions are obtained from the base grid solutions at times $t1$ and $t3$ through linear interpolation. Whereas at time $t3$, the fine grid boundary conditions are provided by the base grid solution at time $t3$ alone, because solutions on both grids are at the same time levels, and no interpolation in time is needed.

When the fine grid solution procedure is activated at time

TABLE I. Simulation parameters. Case *a* is a single fine grid system. Case *b* is a two-grid system. Case *c* is a four-grid system. Case *d* is a two-grid system. $G1$ represents a base grid, and $G2$, $G3$, and $G4$ represent composite fine grids. (i,j) designates the coordinates of the lower left corner of the composite grid relative to the $G1$ grid. The Reynolds number for these cases is 1000.

Case	$G1$	$G2$		$G3$		$G4$	
	Size	Size	(i,j)	Size	(i,j)	Size	(i,j)
a	255×255						
b	128×128	255×137	(1,1)				
c	128×128	57×57	(1,1)	111×99	(73,1)	101×101	(1,78)
d	46×46					43×37	(1,28)

TABLE II. Locations and values of maximum and minimum stream functions for three vortices of the lid-driven cavity flow at a Reynolds number of 1000.

Case	Primary vortex			Lower left vortex			Lower right vortex		
	ψ_{max}	x/L_{lid}	y/L_{lid}	ψ_{min}	x/L_{lid}	y/L_{lid}	ψ_{min}	x/L_{lid}	y/L_{lid}
Ghia *et al.*	0.1179	0.5313	0.5625	−0.000231	0.0859	0.0781	−0.00175	0.8594	0.1094
Hou *et al.*	0.1178	0.5333	0.5647	−0.000222	0.0902	0.0784	−0.00169	0.8667	0.1137
U²RANS	0.1171	0.5315	0.5669	−0.000231	0.0866	0.0748	−0.00173	0.8661	0.1102
Case *a*	0.1140	0.5315	0.5669	−0.000201	0.0827	0.0787	−0.00160	0.8622	0.1142
Case *b*, *G*1	0.1140	0.5276	0.5669	−0.000195	0.0866	0.0787	−0.00158	0.8661	0.1181
Case *b*, *G*2				−0.000204	0.0827	0.0787	−0.00171	0.8583	0.1142
Case *c*, *G*1	0.1140	0.5276	0.5669	−0.000195	0.0866	0.0787	−0.00158	0.8661	0.1181
Case *c*, *G*2				−0.000239	0.0866	0.0827			
Case *c*, *G*3							−0.00176	0.8583	0.1142

$t2$, f_i values in this grid are initialized through the following two steps. The first step is to insert the time-averaged f_i values on the base grid into the lattice nodes that are common to both grids, and are denoted by the open circles in Fig. 5. The second step is to use linear interpolation for the calculation of f_i values at lattice nodes denoted by the gray circles in Fig. 5. Note that the above initialization procedure takes place at the post-streaming stage of the base grid at time $t3$. After the initialization step, only the fine grid boundary conditions along the bold solid line in Fig. 5 are required. They are obtained in the same fashion as above.

If node *A* in Fig. 5 is on the wall and the bounce-back boundary condition is applied, the interpolation of f_i values for node *B* is not possible. A special treatment for assigning values to fine grid nodes of this type is described below. First, the average velocities at node *B* in Fig. 5 are calculated, through interpolation, by using the velocities at node *C* and by invoking the no-slip boundary condition at node *A*. Then f_i values are estimated using the equilibrium distribution function [Eq. (2)].

V. RESULTS AND DISCUSSIONS

Simulation of a lid-driven cavity flow at a Reynolds number 1000 on a single grid 255×255 (case *a* in Table I) is first carried out for later comparison. The grid size is chosen to be close to a 256×256 lattice used by Hou *et al.* [17]. They differ by one grid point in the *x* and *y* directions to allow grid coarsening by a factor of 2 in our case. The same problem with the same mesh is also calculated by the U²RANS finite-volume code [19] for comparison. We will also compare our results with the benchmark numerical solutions of Ghia *et al.* [20], who used a finite-difference stream-vorticity method. The convergence criterion is taken as that the change of the maximum value of stream function for 10 000 consecutive iterations is less than 5×10^{-5}, similar to that of Hou *et al.* [17]. Thus, for case *a*, the number of iterations for convergence is 200 000.

At a Reynolds number of 1000, three vortices are generated: one near the center of the domain, one at the lower left corner, and one at the lower right corner. The maximum and minimum values of the stream function ψ associated with these vortices, and their locations, are listed in Table II. The maximum stream function ψ_{max} for case *a* is found at nearly the same location as others, and is only 2.5% smaller in magnitude. Since the normalized grid spacing is $1/254\approx0.004$, the locations of the lower vortices differ, if any, by only one grid point as compared with other cases.

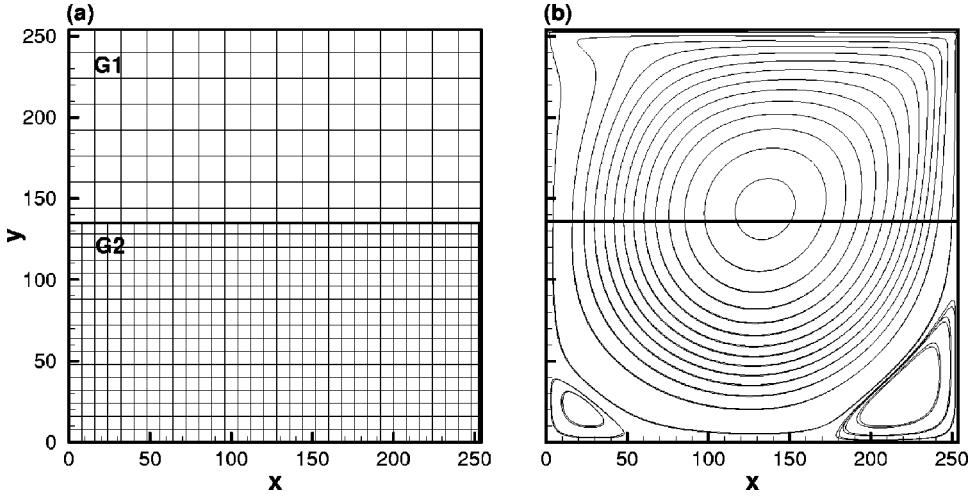

FIG. 6. Mesh and solution for case *b* (refer to Table I). (a) Grid layout; only every eighth grid line is displayed. (b) Contours of the stream function on both grids.

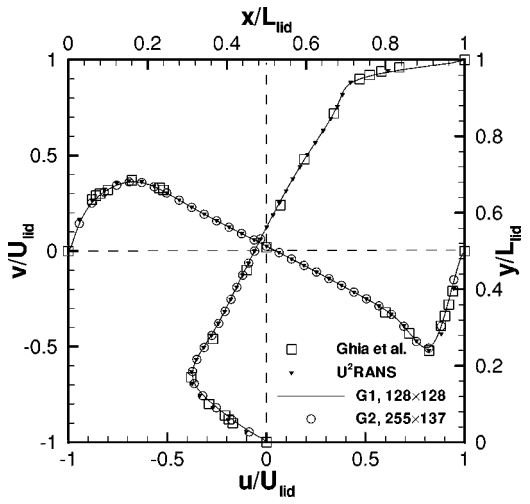

FIG. 7. Velocity profiles $u(y)/U_{lid}$ and $v(x)/U_{lid}$ at the vertical and horizontal centerlines (denoted by dashed lines) of the lid-driven cavity at Re=1000. The benchmark data of Ghia *et al.* are displayed for comparison. $G1$ and $G2$ stand for the coarse and fine grids, respectively (refer to case *b* in Table I).

A. Two-grid system

For case *b* in Table I, the fine grid $G2$ consists of 255 $\times$137 grid points, and is placed at the lower half of the domain, as shown in Fig. 6(a). The vertical extent of the grid is chosen to be slightly above the horizontal centerline, allowing a comparison of v velocity profile at the centerline on both grids, and with the aforementioned numerical results as well. The contours of the stream function on both grids are shown in Fig. 6(b). The contour lines exhibit a smooth transition along the intergrid boundary, indicating that no unphysical solution occurs there. In addition, the contours on both grids almost overlap each other except in the vicinity of two lower corner vortices, where velocity magnitudes are relatively small.

Figure 7 displays the normalized $u(y)$ and $v(x)$ velocity profiles along the centerlines on both grids together with the numerical solutions of Ghia *et al.* [20] and the U²RANS code. Clearly, these data are in excellent agreement. Given that the horizontal centerline at $y/L_{lid}=0.5$ on the $G2$ grid is only slightly below the upper boundary of the $G2$ grid, the intergrid communication procedure described in Sec. IV properly transfers the base grid information to the fine grid.

In Table II, the locations and stream functions of the centers of three vortices for case *b* are listed. The $G1$ grid solution is expected to contain more error, and predicts a relatively lower ψ_{min} for the two lower vortices. On the other hand, the $G2$ grid solution produces more accurate ψ_{min} values than the $G1$ grid as compared with the results from Ghia *et al.* [20], Hou *et al.* [17], and the U²RANS model. By using Richardson extrapolation [Eq. (8)] and solutions from the $G1$ and $G2$ grids, we are able to estimate the average error in the $G2$ grid. This error is marked in Fig. 3, and its order of magnitude agrees well with the foregoing error analysis. We also use the $G1$ grid solution of this case along with the fine grid solution of case *a* for an evaluation of the error of the latter. The result (Fig. 3) shows that the error has the same order of magnitude as other fine grid cases. Thus the $G2$ grid solution is more accurate than the $G1$ grid solution.

B. Four-grid system

Next we test a four-grid system, which is composed of a coarse base grid $G1$ and three fine grids $G2$, $G3$, and $G4$ (refer to case *c* in Table I). The grid layout is shown in Fig. 8(a). The $G2$ and $G3$ grids cover the lower left and right corner vortices, respectively; the $G4$ grid is placed at the upper left corner. The contours of stream function at time 200 000 for the four grids are displayed in Fig. 8(b). These contours almost collapse, except those near the lower corner vortices. A comparison of the lower left vortex in Table II shows that its location differs from the $G2$ solution of case *b* only by one grid point in the x and y directions. Its ψ_{min} value is even closer to the results obtained from the conventional methods. The same feature is observed in the lower right vortex. We further use Eq. 8 to estimate the average error associated with these three fine grids, and mark it in Fig. 3. The order of magnitude of the error is in good agreement with other fine grid solutions.

C. CPU time

Another interesting feature is the CPU time measured by the function DTIME for cases *a*, *b*, and *c*. They are 1029, 130.9, and 309.3 CPU seconds, respectively. Note that for cases *b* and *c*, the time $t3$ (refer to Fig. 4), immediately after initiating the fine grid solution procedure, is 160 000. It is

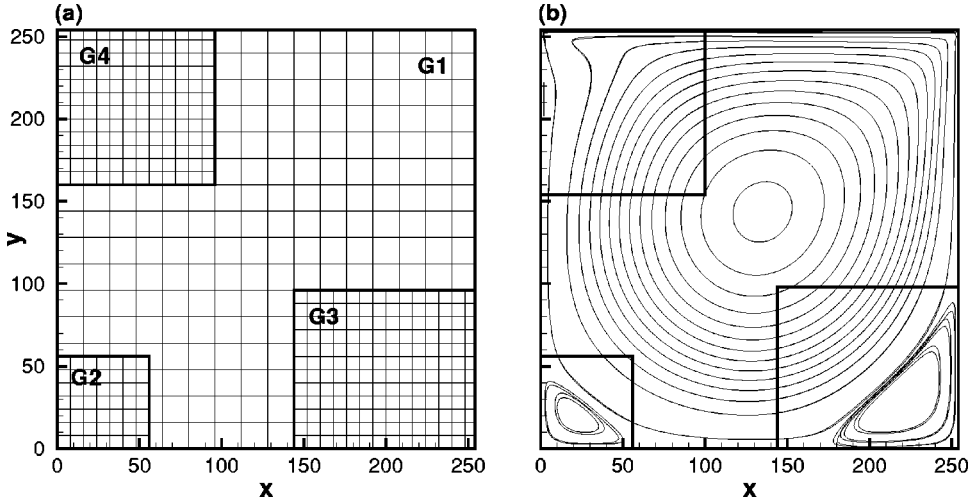

FIG. 8. Mesh and solution for case *c* (refer to Table I). (a) Grid layout; only every eighth grid line is displayed. (b) Contours of the stream function on all of the grids.

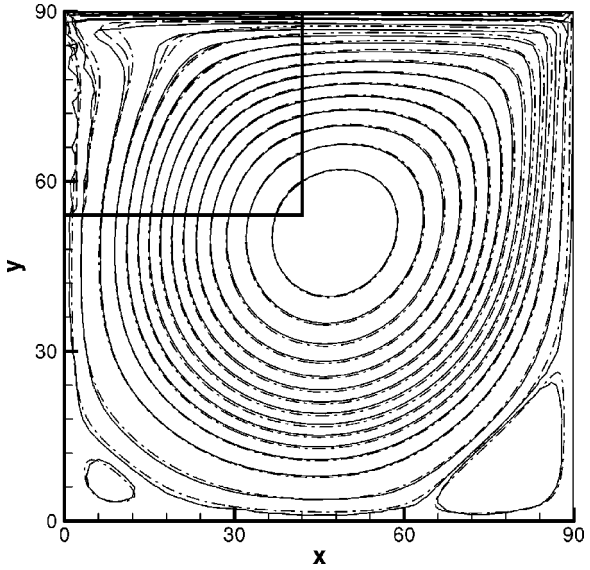

FIG. 9. Contours of the stream function. Case *d*: solid line, $G1$ 46×46; dashed line, $G2$ 43×37. Case *a*: dot-dashed line, fine grid 255×255.

equivalent to 80 000 iterations with $\delta t=2$ on the $G1$ grid. This explains why a considerable saving of the CPU time can be achieved. Case *b* is about eight times faster than case *a*, with the same accurate solutions at designated areas, while case *c* is about three times faster. Case *c* consumes more CPU time than case *b* because more interpolation is involved.

D. Removal of the oscillatory solution

At high Reynolds numbers or with coarse grids, oscillatory solutions may develop at the upper left corner of the cavity [17]. The objective of this section is to investigate whether the proposed composite grid method is able to alleviate this problem. We simulate the same flow using a very coarse grid $G1$, 46×46, which is referred to as case *d* in Table I. In this case, a fine grid $G2$ is nested at the upper left corner. The contours of stream function for both $G1$ and $G2$ grids are depicted in Fig. 9. The contours of stream function for case *a* are also displayed in the same figure for comparison. Figure 9 shows that the oscillation at the upper left corner is removed by the composite grid method.

VI. CONCLUSION

In this paper, we have presented a composite lattice Boltzmann method, which allows local grid refinement on the framework of a regular lattice. This method is based on the observation that discrete particle distribution functions are interchangeable among grids of different length and time scales at the post-streaming stage, if the ratio of length and time scales in these grids is kept constant. The accuracy of the discrete distribution function on the base grid level seems to have little effect on the fine grid solution.

We first examine the overall order of accuracy of the scheme and the discretization errors on grids of different length and time scales. The results show that the LBE method has an order of accuracy of 1.4 for the lid-driven cavity studied. A comparison of CPU time required for each case shows that a considerable saving in CPU time can be achieved if coarser grids are used in regions where flow structures are smooth or not important.

The strategy of the proposed method is, first, to carry out a simulation on a coarse base grid. This allows a rapid propagation of information on boundaries throughout the entire simulation domain, and captures large-scale eddy structures effectively. Then the solution is mapped to the fine grid using linear interpolation. Subsequently, the dependent variables on the base and fine grids are advanced in time with the fine grid boundary conditions obtained from the base grid solution through linear interpolation.

We have demonstrated that the proposed method can produce more accurate solutions at designated regions nested by the fine grids at a much lower computational cost, and that the composite fine grid can remove oscillatory solutions caused by insufficient grid points.

ACKNOWLEDGMENTS

We thank Professor Shiyi Chen at Johns Hopkins University for providing a two-dimensional LBE code. This work was supported by the Carver Scientific Research Initiative Grants Program at the University of Iowa.

[1] S. Chen and G.D. Doolen, Annu. Rev. Fluid Mech. **30**, 329 (1998).
[2] X. He and L.-S. Luo, Phys. Rev. E **55**, R6333 (1997).
[3] X. He and L.-S. Luo, Phys. Rev. E **56**, 6811 (1997).
[4] X. He, L.-S. Luo, and M. Dembo, J. Comput. Phys. **129**, 357 (1996).
[5] X. He and G.D. Doolen, Phys. Rev. E **56**, 434 (1997).
[6] X. He and G.D. Doolen, J. Comput. Phys. **134**, 306 (1997).
[7] N. Cao, S. Chen, S. Jin, and D. Martínez, Phys. Rev. E **55**, R21 (1997).
[8] R. Mei and W. Shyy, J. Comput. Phys. **143**, 426 (1998).
[9] S. Succi, G. Amati, and R. Benzi, J. Stat. Phys. **81**, 5 (1995).
[10] H. Xi, G. Peng, and S.-H. Chou, Phys. Rev. E **59**, 6202 (1999).
[11] H. Xi, G. Peng, and S.-H. Chou, Phys. Rev. E **60**, 3380 (1999).
[12] O. Filippova and D. Hänel, J. Comput. Phys. **147**, 219 (1998).
[13] R. Mei, L.-S. Luo, and W. Shyy, J. Comput. Phys. **155**, 307 (1999).
[14] P.P. Sullivan, J.C. McWilliams, and C.-H. Moeng, Boundary-Layer Meteorol. **80**, 167 (1996).
[15] P.L. Bhatnagar, E.P. Gross, and M. Krook, Phys. Rev. **94**, 511 (1954).
[16] J.H. Ferziger and M. Perić, *Computational Methods for Fluid Dynamics* (Springer, New York, 1999).
[17] S. Hou, Q. Zou, S. Chen, G. Dollen, and A.C. Cogley, J. Comput. Phys. **118**, 329 (1995).
[18] D.H. Rothman and S. Zaleski, *Lattice-Gas Cellular Automata: Simple Models of Complex Hydrodynamics* (Cambridge University Press, New York, 1997).
[19] Y. Lai, A1AA J. (to be published).
[20] U. Ghia, K.N. Ghia, and C.T. Shin, J. Comput. Phys. **48**, 387 (1982).