INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 23, 1–18 (1996)

COMPARISON OF ACCURACY AND PERFORMANCE FOR LATTICE BOLTZMANN AND FINITE DIFFERENCE SIMULATIONS OF STEADY VISCOUS FLOW

DAVID R. NOBLE, JOHN G. GEORGIADIS AND RICHARD O. BUCKIUS

Department of Mechanical and Industrial Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, U.S.A.

SUMMARY

The lattice Boltzmann method (LBM) is used to simulate flow in an infinite periodic array of octagonal cylinders. Results are compared with those obtained by a finite difference (FD) simulation solved in terms of streamfunction and vorticity using an alternating direction implicit scheme. Computed velocity profiles are compared along lines common to both the lattice Boltzmann and finite difference grids. Along all such slices, both streamwise and transverse velocity predictions agree to within 0·5% of the average streamwise velocity. The local shear on the surface of the cylinders also compares well, with the only deviations occurring in the vicinity of the corners of the cylinders, where the slope of the shear is discontinuous. When a constant dimensionless relaxation time is maintained, LBM exhibits the same convergence behaviour as the FD algorithm, with the time step increasing as the square of the grid size. By adjusting the relaxation time such that a constant Mach number is achieved, the time step of LBM varies linearly with the grid size. The efficiency of LBM on the CM-5 parallel computer at the National Center for Supercomputing Applications (NCSA) is evaluated by examining each part of the algorithm. Overall, a speed of 13·9 GFLOPS is obtained using 512 processors for a domain size of 2176 × 2176.

KEY WORDS lattice Boltzmann; finite difference; parallel computing

1. INTRODUCTION

The lattice Boltzmann method (LBM)[1–5] is a relatively new, kinetic theory-based, numerical technique for studying fluid mechanics. LBM has been used to model various physical phenomena and has been shown to produce physically realistic results. Other quantitative studies have endeavoured to establish LBM as an alternative scheme for fluid mechanics simulations through comparisons with results produced by other more conventional methods.[6–11]

One obstacle in performing such close comparisons, however, has been the development of general, accurate boundary conditions for LBM. The simple 'bounce-back' boundary condition used in the majority of LBM simulations is a first-order method for modelling stationary walls.[12–15] Skordos[16] suggested a more accurate finite difference scheme to impose boundary and initial conditions. Noble *et al.*[17,18] recently proposed consistent hydrodynamic boundary conditions for both hexagonal and square two-dimensional grids which yield greater accuracy and do not use expensive finite difference operations. It is precisely the imposition of this boundary condition on complex solid boundaries that is addressed in this paper. This is a necessary and natural extension of previous investigations[6–11,17,18]

CCC 0271–2091/96/010001–18

Received 30 June 1995
Revised 9 October 1995

towards simulating flows around boundaries of complex topology, such as flows in porous media[19] or particulate flows.[9]

Accurate simulation of complex fluid flows requires a great deal of computational resources. Increasingly, massively parallel computers are being used to attack these problems. The two most commonly used models in parallel computing applications are the single-instruction/multiple-data (SIMD) and multiple-instruction/multiple-data (MIMD) approaches. Under the SIMD model the arrays containing the data are divided among the processors. Each processor acts on the same instructions, simultaneously operating on the data. Under the MIMD model each processor is independent, running its own programme. The SIMD model can significantly decrease execution times when the same operation is to be applied to many array elements. In general, parallel techniques are not advantageous, however, when a great deal of information must be passed from location to location, since this requires interprocessor communication.

The lattice Boltzmann algorithm is well suited for parallel computation. The evolution equations involve many local operations which can be performed simultaneously. The interprocessor communication involved is explicit in nature and can therefore be performed using optimized algorithms. Also, the slight compressibility of LBM can be viewed as advantageous. Owing to the non-local nature of the solution to the Poisson equation, finite difference schemes can spend more than half of the computational time solving for the pressure.[20] LBM does not, however, require the explicit solution of a Poisson equation for pressure. Instead, LBM incorporates a finite speed of sound and the range of influence per time step is limited to only a subset of the entire domain. In this regard LBM may be considered as a pseudocompressibility method.[21] In LBM the degree of compressibility can be controlled to achieve both efficient and accurate results.

The objective of this work is to perform a detailed comparison of the accuracy and performance of LBM and a finite difference (FD) scheme for steady flows with recirculation and non-trivial boundary conditions. The finite difference code[22] utilizes an alternating direction implicit algorithm and is formulated in terms of streamfunction and vorticity. The problem under consideration is the flow of fluid through a periodic array of cylinders which are octagonal in cross-section. The accuracy of LBM is established through pointwise comparisons of velocity and shear. The performance of LBM is quantitatively evaluated for massively parallel computing architectures.

2. LATTICE BOLTZMANN THEORY

2.1. Lattice Boltzmann method with a body force on a square lattice

In this study an orthogonal, square lattice is utilized in which each node has eight nearest neighbours. The lattice incorporates horizontal and vertical links of length Δx and diagonal links of length $\sqrt{2}\Delta x$. A velocity is associated with each direction,

$$\mathbf{e}_i = |\mathbf{e}_i|\left(\cos\left(\frac{2\pi(i-1)}{8}\right), \sin\left(\frac{2\pi(i-1)}{8}\right)\right), \quad i = 1, 2, \ldots, 8, \tag{1}$$

where $|\mathbf{e}_i| = \Delta x/\Delta t$ for the horizontal and vertical directions ($i = 1, 3, 5, 7$) and $|\mathbf{e}_i| = \sqrt{2}\Delta x/\Delta t$ for the diagonal directions ($i = 2, 4, 6, 8$). The particle distribution $f_i(\mathbf{x}, t)$ indicates the probability of finding a particle at location $\mathbf{x}$ and time t that is moving with velocity $\mathbf{e}_i$. A rest particle contribution $f_o(\mathbf{x}, t)$ is also included. The primary variables density and velocity are found from this particle distribution according to

$$\sum_i f_i = \rho, \tag{2}$$

$$\sum_i f_i \mathbf{e}_i = \rho \mathbf{u}. \tag{3}$$

A measure of the kinetic energy κ associated with the lattice velocity is also found according to

$$\tfrac{1}{2}\sum_i (\mathbf{e}_i \cdot \mathbf{e}_i) f_i = \rho\kappa. \tag{4}$$

The difference between this microscopic kinetic energy and the energy of the macroscopic velocity field is the internal energy ε:

$$\rho\varepsilon = \tfrac{1}{2}\sum_i (\mathbf{e}_i - \mathbf{u})^2 f_i = \rho\kappa - \tfrac{1}{2}\rho|\mathbf{u}|^2. \tag{5}$$

The discrete velocity Boltzmann equation is given by

$$\frac{\partial f_i}{\partial t} + \mathbf{e}_i \cdot \nabla f_i = \Omega_i(f(\mathbf{x}, t)) + F_i, \tag{6}$$

where $\Omega_i(f(\mathbf{x}, t))$ is a collision term which accounts for the creation and destruction of particles moving with velocity e_i due to particle collisions. The term F_i gives rise to a body force and is defined such that

$$\sum_i F_i \mathbf{e}_i = \mathbf{F}, \tag{7}$$

where $\mathbf{F}$ is the imposed body force. So that the forcing process conserves mass, the body force term is also subject to the constraint

$$\sum_i F_i = 0. \tag{8}$$

In this study a uniform body force in the x-direction is desired and F_i is chosen as

$$F_1 = F_2 = F_8 = C, \qquad F_4 = F_5 = F_6 = -C, \qquad F_0 = F_3 = F_7 = 0, \tag{9}$$

where C is a constant. The constant is determined by substituting the chosen distribution into (7), which gives

$$\sum_i F_i \mathbf{e}_i = 6C\frac{\Delta x}{\Delta t}\hat{x} = \mathbf{F} = |\mathbf{F}|\hat{x}, \tag{10}$$

where $\hat{x}$ is a unit vector in the x-direction. The discrete velocity Boltzmann equation is spatially and temporally discretized using a Langrangian discretization, which yields

$$f_i(\mathbf{x} + \mathbf{e}_i\Delta t, t + \Delta t) = f_i(\mathbf{x}, t) + \Omega_i(f(\mathbf{x}, t))\Delta t + F_i\Delta t. \tag{11}$$

Utilizing the linearized, single-time relaxation model of Bhatnagar *et al.*[23] applied to lattice Boltzmann,[4] the collision operator is written as

$$\Omega_i(f) = -\frac{1}{\tau}(f_i - f_i^{\mathrm{eq}}). \tag{12}$$

Using this simplification, the lattice Boltzmann evolution equation is written as

$$f_i(\mathbf{x} + \mathbf{e}_i\Delta t, t + \Delta t) = f_i(\mathbf{x}, t) + \frac{\Delta t}{\tau}\left(f_i^{\mathrm{eq}}(\mathbf{x}, t) - f_i(\mathbf{x}, t)\right) + F_i\Delta t. \tag{13}$$

It is useful to define a dimensionless relaxation time $\tau^* = \tau/\Delta t$ and a simplified body force term $F_i' = F_i \Delta t$. Substituting gives the final form of the lattice Boltzmann evolution equation with a body force as

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) + \frac{1}{\tau^*}\left(f_i^{\mathrm{eq}}(\mathbf{x}, t) - f_i(\mathbf{x}, t)\right) + F_i', \tag{14}$$

where

$$F_1' = F_2' = F_8' = \frac{|\mathbf{F}|(\Delta t)^2}{6\Delta x}, \qquad F_4' = F_5' = F_6' = -\frac{|\mathbf{F}|(\Delta t)^2}{6\Delta x}, \qquad F_0' = F_3' = F_7' = 0. \tag{15}$$

Using this technique, the solution of the fluid equations is thus reduced to two major steps. First, in a collision and forcing step the distributions are modified according to the right-hand side of (14). Second, the particle distributions stream to their nearest neighbours. It is noted that the collision and forcing calculation is completely local in that all quantities are evaluated at location $\mathbf{x}$. All of the non-local interaction takes place during the streaming step and involves only the nearest neighbours.

The equilibrium particle distribution is selected so that the continuum fluid equations are recovered when the Boltzmann transport equation is truncated to its long-wavelength and low-frequency limit. The equilibrium distribution utilized here is

$$f_o^{\mathrm{eq}} = \rho\left(\frac{2}{7} - \frac{2}{3c^2}(\mathbf{u}\cdot\mathbf{u})\right), \tag{16}$$

$$f_i^{\mathrm{eq}} = \rho\left(\frac{1}{7} + \frac{1}{3c^2}(\mathbf{e}_i\cdot\mathbf{u}) + \frac{1}{2c^4}(\mathbf{e}_i\cdot\mathbf{u})^2 - \frac{1}{6c^2}(\mathbf{u}\cdot\mathbf{u})\right), \quad i = 1, 3, 5, 7, \tag{17}$$

$$f_i^{\mathrm{eq}} = \rho\left(\frac{1}{28} + \frac{1}{12c^2}(\mathbf{e}_i\cdot\mathbf{u}) + \frac{1}{8c^4}(\mathbf{e}_i\cdot\mathbf{u})^2 - \frac{1}{24c^2}(\mathbf{u}\cdot\mathbf{u})\right), \quad i = 2, 4, 6, 8, \tag{18}$$

where $c = \Delta x/\Delta t$ is the computational speed of sound, f_i^{eq} is the equilibrium distribution of particles moving in direction i and f_o^{eq} is the equilibrium distribution of rest particles. The kinematic viscosity of the fluid, ν, and speed of sound, c_s, are given by

$$\nu = \frac{2\tau^* - 1}{6}\frac{(\Delta x)^2}{\Delta t}, \tag{19}$$

$$c_s = \sqrt{(3/7)}(\Delta x/\Delta t). \tag{20}$$

2.2 Boundary condition implementation

In order to simulate the cylinder surface using the lattice Boltzmann method, suitable boundary conditions in terms of the particle distribution must be imposed. In the consistent hydrodynamic approach proposed by Noble *et al.*,[17,18] a complete set of constraints for the particle distribution is developed such that a specified velocity profile is maintained on the boundaries. For LBM utilizing a square grid, one of these constraints is developed by requiring that the internal energy be kept constant.[18] Another feature of the consistent hydrodynamic approach is that it provides a constraint for the density at the boundary in terms of the velocity boundary conditions.

Some additional nomenclature is needed in order to describe the consistent hydrodynamic boundary conditions. First, nodes which lie wholly within the fluid are termed interior or fluid nodes (denoted by subscript f). Second, nodes which lie just outside the boundary of the fluid domain within the wall are termed wall nodes (denoted by subscript w). Last, nodes on the boundary between the fluid mass and the

wall are termed boundary nodes (denoted by subscript b). The full set of constraints governing the particle distribution function of the boundary node at location $\mathbf{x}_b$ is given by

$$\sum_{i=w\to b} f_i(\mathbf{x}_b, t+\Delta t) = \rho(\mathbf{x}_b, t+\Delta t) - \sum_{i=f,b\to b} f_i(\mathbf{x}_b, t+\Delta t), \tag{21}$$

$$\sum_{i=w\to b} f_i(\mathbf{x}_b, t+\Delta t)\mathbf{e}_i = \rho(\mathbf{x}_b, t+\Delta t)\mathbf{u}(\mathbf{x}_b, t+\Delta t) - \sum_{i=f,b\to b} f_i(\mathbf{x}_b, t+\Delta t)\mathbf{e}_i, \tag{22}$$

$$\sum_{i=w\to b} (\mathbf{e}_i \cdot \mathbf{e}_i) f_i(\mathbf{x}_b, t+\Delta t) = 2\rho(\mathbf{x}_b, t+\Delta t)\varepsilon(\mathbf{x}_b, t+\Delta t)$$

$$+\rho[u^2(\mathbf{x}_b, t+\Delta t) + v^2(\mathbf{x}_b, t+\Delta t)] - \sum_{i=f,b\to b} (\mathbf{e}_i \cdot \mathbf{e}_i) f_i(\mathbf{x}_b, t+\Delta t), \tag{23}$$

where the notation $f \to b$ is used to denote directions from neighbouring fluid nodes to the boundary node of interest. Likewise, $b \to b$ denotes directions from neighbouring boundary nodes to the boundary node of interest and $w \to b$ denotes directions from neighbouring wall nodes to the boundary node of interest. Note that the set of directions $b \to b$ includes that of the rest particle distribution which is stationary at location $\mathbf{x}_b$.

This boundary condition gives the constraints on the unknown components of the particle distribution which are produced by imposing the macroscopic velocity boundary conditions and fixed internal energy. Determining the individual unknown components of the distribution requires the application of these constraints for a boundary geometry. For an octagonal obstacle, 20 different types of boundary orientations are encountered. The explicit equations used to find the density and unknown components of the particle distribution are given in the Appendix.

3. SIMULATIONS OF FLOW THROUGH A PERIODIC ARRAY OF OCTAGONAL CYLINDERS

In this study the flow through a two-dimensional periodic array of infinite parallel octagonal cylinders is considered. The flow field is calculated using the lattice Boltzmann method (LBM) and by a finite difference (FD) method utilizing an alternating direction implicit algorithm. The FD algorithm is formulated in terms of streamfunction and vorticity. The flowfield is computed by simulating the flow around a single octagonal cylinder while imposing periodicity at each of the domain boundaries. A 68×68 computational grid used for LBM and a $41 \times 41 \times 4$ grid (four subdomains, each 41×41) used for the FD calculations are shown in Figure 1. (Finer grids are used in computations. The grids shown in Figure 1 demonstrate the structure of each computational domain.) The octagonal cross-section provides boundaries which conform to the lattice Botzmann grid which includes both the Cartesian directions as well as the diagonal directions. As a result of the 45° angle inherent to the LBM grid, the obstacle cross-section is not that of a regular octagon. Instead, the length of the sides which lie along the diagonals is $\sqrt{2}/1{\cdot}4$ times the length of the sides which lie along the horizontal and vertical grid lines. This results in a 1% difference between the length of the Cartesian and diagonal sides of the octagon.

The two computational grids are quite different in nature, but a number of grid lines are common to both grids. These lines correspond to slices through the domain and are referenced with the aid of the nine points labelled for each grid in Figure 1. Slice AHG lies at the inlet. Slice BOF cuts vertically through the centre of the obstacle. The slice that cuts horizontally through the centre of the obstacle is described by HOD. Slice GOC cuts diagonally through the domain from the lower left to the upper right.

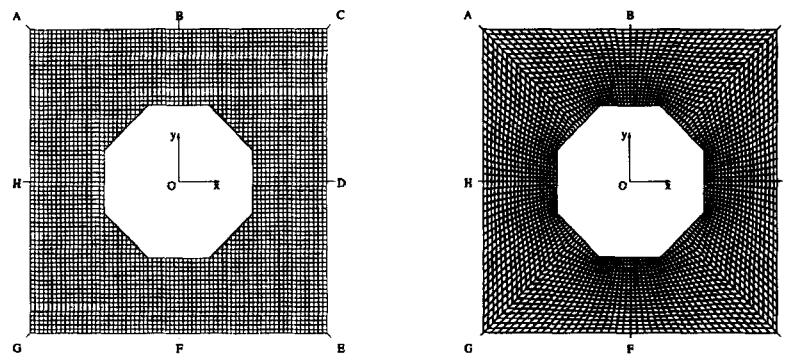

Figure 1. Computational grids for the lattice Boltzmann method and for the finite difference method using an alternating direction implicit scheme. The lattice Boltzmann grid is 68 × 68 and the finite difference grid consists of four subdomains of size 41 × 41. The eight labelled points on each grid are used to identify grid lines which are common to both grids. Slice AHG lies at the inlet of the domain. Slice BOF is a vertical slice through the centre of the obstacle. Slice GOC is a diagonal slice from the lower left to the upper right

In the lattice Boltzmann simulations reported here, the domain is initialized with the equilibrium distribution for zero velocity. The solution is marched forward in time until a steady solution is reached. In order to determine this, the largest net change in velocity for 500 time steps is computed. When the length of this vector divided by the average streamwise velocity is less than 10^{-6}, the solution is considered converged.

The lattice Boltzmann algorithm is naturally set up for problems involving a known body force. In order to impose a desired Reynolds number, a method must be developed for calculating the body force needed to maintain the specified average velocity. Force is generally equivalent to a time rate of change of momentum. The uniform body force applied over the entire volume gives rise to a total momentum increase according to

$$\int |\mathbf{F}| \mathrm{d}A = \frac{\partial}{\partial t} \int \rho u \, \mathrm{d}A. \tag{24}$$

Therefore the body force needed to increase the momentum to the level corresponding to the required average velocity is given by

$$\int |\mathbf{F}| \, \mathrm{d}A = \frac{\bar{\rho} U A - \int \rho u \, \mathrm{d}A}{\Delta t}, \tag{25}$$

where $\bar{\rho}$ is the average density of the fluid and A is the total area of the domain, including the obstacle. This force is estimated by

$$|\mathbf{F}| = \frac{\bar{\rho} U N_x N_y - \sum_{x,y} \rho u}{N_{\text{fluid}} \Delta t}, \tag{26}$$

where N_x and N_y are the numbers of nodes in the x- and y-directions respectively. The quantity N_{fluid} is the total number of nodes in the fluid and is used as an estimate of the area of the fluid. The body force is uniform over the entire domain, including both the fluid and the obstacle, but only effects the momentum of the fluid. A second approximation is made by increasing the momentum such that the

average velocity matches the required amount at the end of the collision and forcing step. Exactly matching the Reynolds number requires that the average velocity be matched at the end of the complete time step. For all simulations performed here, these approximations provide sufficient accuracy. The deviation of the actual Reynolds number from the desired Reynolds number is less than 0·5% for all cases examined.

For the FD simulations an alternating direction implicit algorithm is used to solve for streamfunction and vorticity. The initial conditions for each simulation are computed from the inviscid solution for the given Reynolds number. The solution is marched forward in time until the changes in both the streamfunction and vorticity are less than 5×10^{-6}. The velocity at each grid point is then computed by differentiating the streamfunction.

The behaviour of the flow in a periodic array of cylinders is similar to that for a single cylinder in a cross-flow. Figure 2 shows streamlines computed from lattice Boltzmann simulations for Reynolds numbers $Re = 1$, 10, 50 and 100. The Reynolds numbers for these calculations is based on the average streamwise velocity and the width of the octagon. At $Re = 1$ the flow is nearly symmetric about a vertical plane through the centre of the cylinder. At $Re = 10$ the flow is seen to separate and a recirculation bubble is formed. The flow now differs from that of a single cylinder in a uniform stream. The wake of the upstream obstacle is seen to impinge on the obstacle behind it. Consequently, the velocity along the centreline of the obstacles is moving in the direction opposite to the bulk flow. Interesting behaviour ensues for $Re = 10$, as the recirculation bubble constricts behind the obstacle, but then must expand to accommodate the next obstacle. As the Reynolds number is increased further, the wake becomes wider and the recirculation is intensified.

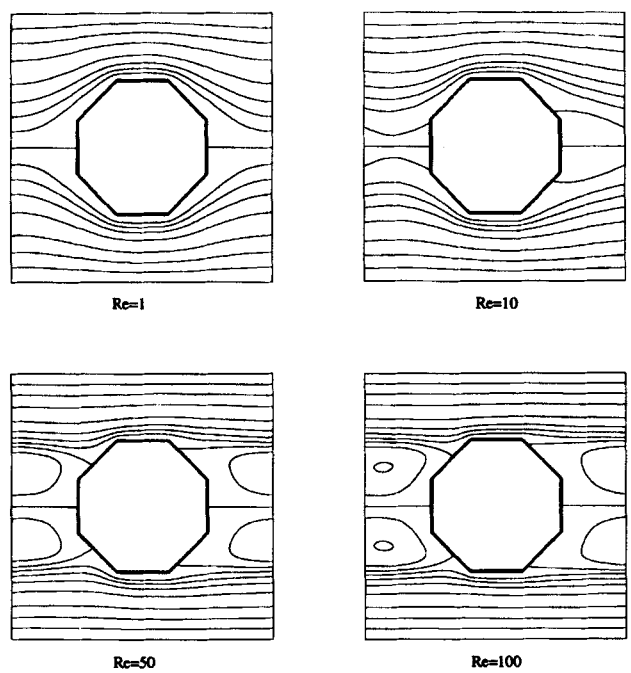

Figure 2. Character of the flow in a periodic array of octagonal cylinders at various Reynolds numbers. Shown are streamlines computed from lattice Boltzmann results for Reynolds numbers $Re = 1$, 10, 50 and 100

4. RESULTS

4.1. Comparison of results for lattice Boltzmann and finite difference solutions

Comparisons of the velocity profiles and shear are made for the lattice Boltzmann and finite difference methods. Pointwise comparisons are performed by plotting the results from a 204 × 204 LBM simulation and an 81 × 81 × 4 FD simulation along lines common to both grids. The Reynolds number for these comparisons is 50 based on the width of the obstacle and U, the average streamwise velocity of the fluid.

4.1.1. Velocity profile comparison. Figures 3 and 4 show the streamwise and transverse velocity components u and v plotted along each of the four slices cut through the domain. Since slice HOD lies along the centreline of the obstacle, the v-component of velocity is zero by symmetry and is not shown. Note that although slice GOC cuts diagonally though the domain, the velocity along this line is plotted as a function of the streamwise co-ordinate x. For all slices the LBM results, plotted using full lines, and FD results, plotted using broken lines, compare very well. For all slices the lattice Boltzmann and FD predictions for u and v agree to within 0·5% of U, the average streamwise fluid velocity. The solutions are in agreement both near and far from the obstacle and both inside and outside the wake of the obstacle.

4.1.2. Pointwise shear comparison. The viscous shear on the surface of the obstacle is calculated and compared. The dimensionless viscous shear imposed by the fluid on the obstacle is found from the vorticity by the relation[24]

$$\frac{-\nu \mathbf{n} \times \omega}{\frac{1}{2}U^2}, \tag{27}$$

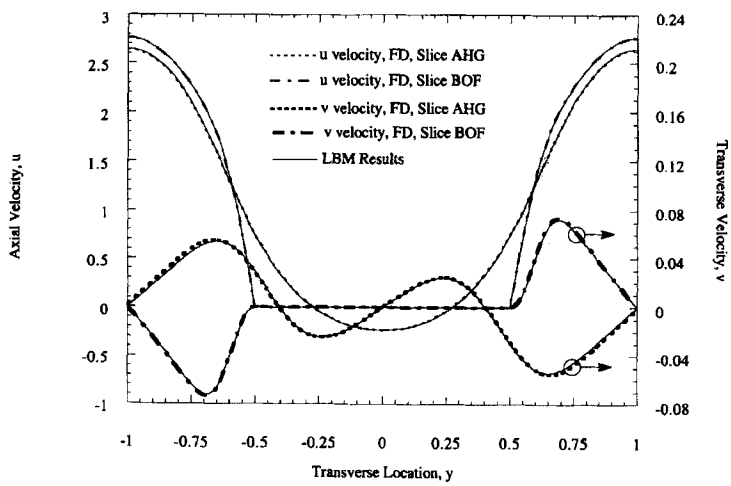

Figure 3. Comparison of lattice Boltzmann and finite difference solutions for streamwise and transverse velocity profiles along slices AHG and BOF for $Re = 50$. All lattice Boltzmann simulation results are shown with full lines. Finite different simulation results are indicated by broken lines

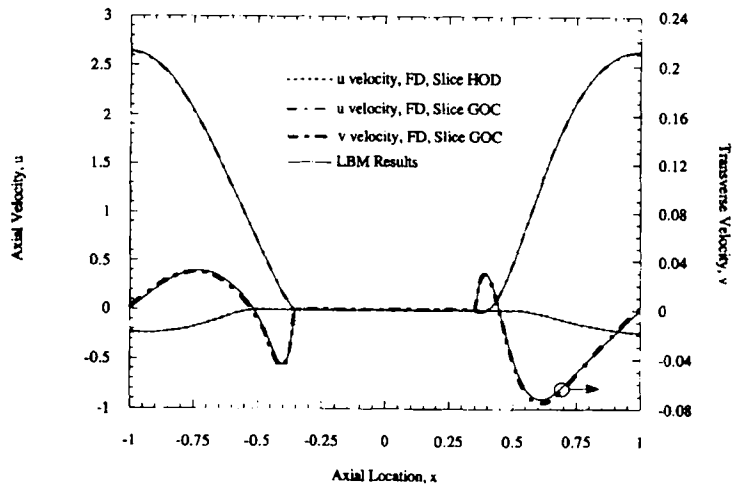

Figure 4. Comparison of lattice Boltzmann and finite difference solutions for streamwise and transverse velocity profiles along slices HOD and GOC for $Re = 50$. The transverse velocity along slice HOD is zero by symmetry and is not shown. All lattice Boltzmann simulation results are shown with full lines. Finite different simulation results are indicated by broken lines

where $\mathbf{n}$ is the outward-facing normal of the obstacle surface, ω is the fluid vorticity at the surface and ν is the kinematic viscosity. The dimensionless shear force per unit area acting in the streamwise direction is then given by

$$\frac{-\nu n_y \omega}{\frac{1}{2}U^2}, \tag{28}$$

where n_y is the y-component of the outward-facing normal and ω is the only non-zero component of the vorticity for two-dimensional flows:

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}. \tag{29}$$

For the FD solution the shear is calculated directly from the computed vorticity. For the LBM solution the vorticity is first calculated by differentiating the velocity. The shear is computed at each point on the surface of the obstacle and then plotted as a function of angle in Figure 5. The angles are calculated for co-ordinate axes fixed at the centre of the obstacle, i.e. 0° corresponds to the stagnation point on the rear of the obstacle.

Once again the results are in close agreement. Along each side of the octagon the predictions by the two methods are nearly indistinguishable in the figure. The deviations occur only near the corners of the obstacle. At these locations the slope of the shear is discontinuous as a result of the discontinuous slope of the obstacle profile. The finite difference solution is locally refined in the vicinity of the corners and appears to better resolve the discontinuity. The uniform grid of the lattice Boltzmann method is too coarse to resolve the discontinuity and this results in small 'wiggles' in the solution. The 'wiggles' are confined to small regions near the corners. Another result is that the two methods give differing predictions for the shear at the corners. The finite difference solution is also susceptible to difficulties in these regions, since the slope of the grid lines is discontinuous at the corners. The disagreement at the corners has relatively little consequences, however, since the shear is acting on an infinitesimal area of the corner.

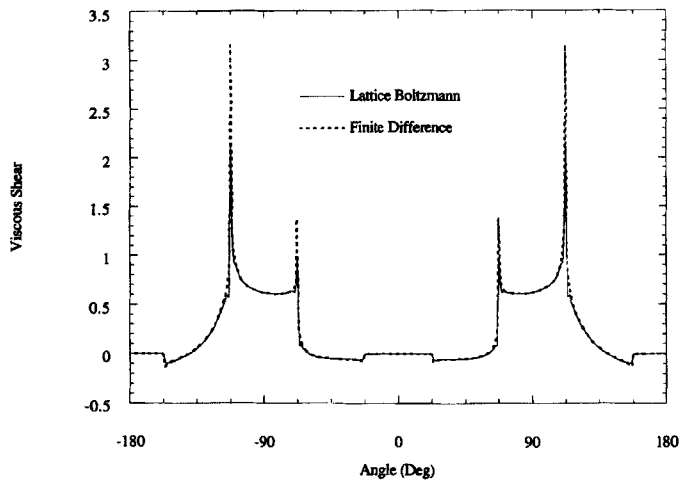

Figure 5. Comparison of lattice Boltzmann and finite difference solutions for viscous shear along the surface of the obstacle. Discontinuities in the slope of the shear occur at the corners of the obstacle

4.1.3. Drag coefficient evaluation. The drag coefficient may be computed using the results of lattice Boltzmann simulations which include the velocity and pressure at each grid point. The finite difference solution is formulated in terms of streamfunction and vorticity, however, and does not give the pressure variation. If the pressure field is to be calculated, an additional Poisson equation must be solved. The drag may be computed from the lattice Boltzmann simulation results in two different ways. First, the shear and pressure forces acting on the obstacle may be integrated to find the total net force. Second, the drag may be computed from the magnitude of the body force which is needed to drive the fluid. Since the body force is a measure of the force imparted to the fluid to keep it moving at the specified Reynolds number, the same force measures the drag imposed on the fluid by the obstacle. Thus the drag coefficient in terms of the body force is given by

$$C_d = \frac{|\mathbf{F}|A}{\frac{1}{2}\rho_0 U^2 L}, \tag{30}$$

where U is the average streamwise fluid velocity, ρ_0 is the average fluid density, L is the width of the obstacle and A is the total area of the domain. It is expected that the prediction based on the body force is superior to that based on the integration over the surface, since the integration is susceptible to errors caused by the discrete integration invervals and the discontinuous slopes of the pressure and shear. As a check, however, the drag coefficient is computed using both methods, and for all cases examined here, the predictions agree to within 2%. The drag coefficient calculated from the body force for periodic arrays of octagonal cylinders for Reynolds number varying from 0·01 to 100 is plotted in Figure 6. At low Reynolds number the relationship between the drag coefficient and the Reynolds number becomes linear. In this regime the drag coefficient is inversely proportional to the Reynolds number as expected for Stokes flow.

4.2. Performance of lattice Boltzmann technique and comparison with finite difference performance

The performance of the lattice Boltzmann method running on the CM-5 massively parallel computer at the National Center for Supercomputing Applications (NCSA) is assessed in this subsection. The behaviour of the lattice Boltzmann code as the domain size is increased is compared with that of the

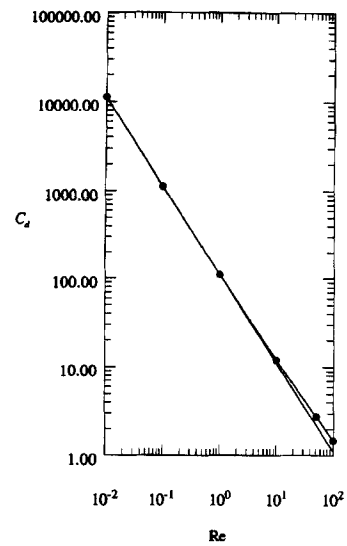

Figure 6. Computed drag coefficient as a function of Reynolds number. The drag coefficient is calculated from lattice Boltzmann predictions of the body force required to maintain the flow. Also shown is a line depicting the inverse dependence of drag on Reynolds number as expected for Stokes flow

finite difference code utilizing an alternating direction implicit scheme. The performance of LBM is also quantified in terms of its ability to effectively utilize parallel computing architectures.

4.2.1. CM-5 implementation specifics. The lattice Boltzmann code used in this study executes in single-instruction/multiple-data (SIMD) mode and is written in CM FORTRAN. CM FORTRAN is a parallel version of FORTRAN 90 designed by Thinking Machines. The CM-5 at NCSA has 512 processors each with 32 Mbytes of memory. Each processor has four vector units giving 128 MFLOPS peak 64 bit floating-point performance per processor. The processor are grouped in partitions, and partition sizes of 32, 64, 128, 256 and 512 processors are available. A separate control processor is used to give instructions to the processors in the partition.

The efficiency of the computer programme is highly dependent on the data structures utilized. In this implementation of LBM, all arrays, including each component of the particle distribution function, are stored with the x- and y-dimensions of the arrays spread across the processors. This causes spatially local calculations to be done within each processor, requiring no expensive interprocessor communication.

In general, three types of communication patterns exist: regular communications, irregular communications and reductions. Regular communications are the least expensive and are used in the streaming process of LBM. Streaming involves the movement of particle populations from one node to another in a consistent manner. This is accomplished using the intrinsic function *cshift*, which shifts data from each node to a neighbouring node. The *cshift* function implicitly assumes periodicity and data on the edge of the domain wraps around to the opposite side. For greater efficiency, this implementation of LBM utilizes the *pshift* routine from the CM Scientific Subroutine Library (CMSSL), in which multiple *cshift* operations are performed in a single subroutine. Irregular communications involve the movement of information in a non-uniform manner and are not required in LBM. Reductions are costly procedures and are encountered when information from all the processors is needed for a computation, i.e. a sum of

the total momentum in the fluid. This operation is performed in order to calculate the average velocity, which is then used to update the body force.

4.2.2. Computational cost comparison. One measure of the cost of computing a steady state flow solution is the number of iterations or time steps required to reach the converged, steady solution. Both the LBM and FD algorithms are used to solve the transient equations describing fluid motion. Consequently, each updated solution of the discrete equations is associated with a step forward in time, δt. If the physical phenomena being modelled develops on a time scale T, the number of iterations required to reach the fully developed solution will be of the order of $T/\Delta t$. Figure 7 shows how the required number of iterations varies with the total number of grid points for both the FD algorithm and LBM. For the alternating direction implicit algorithm the time step and grid size are related through the parameter r according to

$$r = \frac{\Delta t}{(\Delta x)^2}. \tag{31}$$

Thus the time step varies as the square of the grid size and the required number of iterations will increase linearly with the total number of grid points as seen in the figure. For LBM the time step and grid size are related through both the viscosity and the Mach number. The relationship through the viscosity is given by (13) and can also be written as

$$\frac{(\Delta x)^2}{\Delta t} = \frac{\nu}{(2\tau^* - 1)/6}. \tag{32}$$

The relationship through the Mach number can be shown by normalizing the velocity by the computational speed of sound:

$$M_c(\mathbf{x}, t) = \frac{|\mathbf{u}(\mathbf{x}, t)|}{\Delta x/\Delta t}, \tag{33}$$

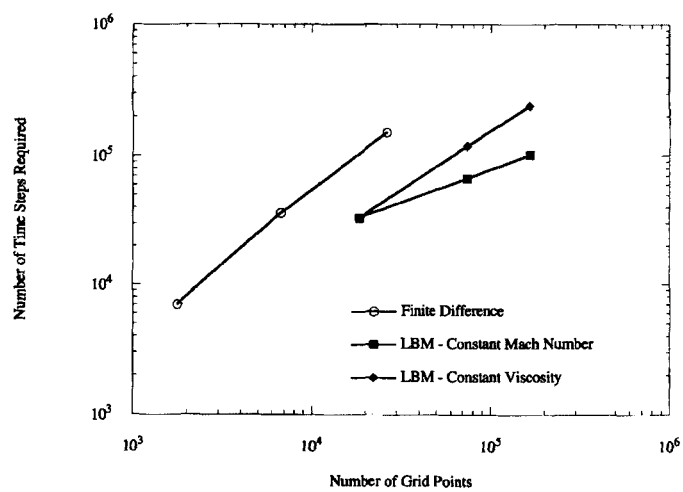

Figure 7. Scaling of iteration count with problem size for the lattice Boltzmann method and the finite difference scheme at a constant Reynolds number of 100. For a constant dimensionless relaxation time, LBM exhibits the same scaling behaviour as the finite difference algorithm, with the time step going as the square of the grid size. For the case of a constant Mach number the time step of LBM varies linearly with the grid size

where $\mathbf{u}$ is the local velocity and M_c is the local computational Mach number. Rearranging, this relationship states

$$\frac{\Delta x}{\Delta t} = \frac{|\mathbf{u}|}{M_c}. \tag{34}$$

From (32) it is apparent that for a constant viscosity and dimensionless relaxation parameter the time step varies as the square of the grid spacing. Thus, as expected, the behaviour of the required number of iterations for LBM parallel that of the FD code when a constant dimensionless relaxation time is maintained. On the other hand, from (34), if a constant Mach number is maintained, the time step varies linearly with the grid size. This causes the number of time steps to increase as the square root of the problem size.

4.2.3. Lattice Boltzmann computational performance by routine and parallel speed-up. The lattice Boltzmann algorithm used in this study is divided into four parts with differing levels of parallel performance. The first step involves the iterative calculation of the body force as described earlier. The second step involves the collision and forcing process. After computing the local velocity and density from the particle distributions, the equilibrium function is computed and the right-hand side of (14) is calculated. The third step is the streaming process in which this quantity is advected to the nearest neighbours. The last step is the calculation of the boundary conditions according to the relations described earlier.

The parallel efficiency of a routine is evaluated by computing the parallel speed-up. The parallel speed-up of a routine is defined as the ratio of the computational speed of the routine on a partition of one size to the computational speed on a reference size partition. Figure 8 shows the parallel speed-up for each part of the LBM scheme for a reference partition size of 32 processors and a fixed problem size of 272 × 272. Ideally, a process will perform twice as fast when the number of processors is doubled. However, this process is mitigated by increasing communication costs and decreased vector lengths. As the number of processors is increased for a fixed problem size, the regular communications and reductions involve increasing amounts of interprocessor communication. Also, the number of computations being performed by each processor is decreasing. This reuslts in shorter vector lengths and resulting less efficient computation.

The collision and forcing process and the boundary condition calculation maintain relatively high efficiencies as the number of processors is increased, since these routines do not involve any interprocessor communication. In fact, the boundary condition calculation exceeds the optimum efficiency when using 32 processors. This superefficiency is attributed to the way in which the data are spread across the processors. When the number of elements allocated to each vector unit by the compiler is not an integral multiple of eight, the vector length on each vector unit, the remainder is padded with values which are later discarded. The superefficiency suggests that when using 64 processors (256 vector units), there is significantly less padding imposed than for 32 processors (128 vector units) and higher computational efficiency results. The efficiency loss of these routines with further increasing the number of processors is due to the decreased vector lengths in the computations. The force calculation and streaming routine shows a much greater decrease in efficiency due to increased communication. In fact, these processes are eventually seen to require more computer time as the number of processors is increased. Another ramification of the changing efficiency is that the routines which involve interprocessor communication take up an increasing percentage of the total computer time. For a partition size of 32 processors the force calculation, collision and forcing process, streaming process and boundary condition calculation take up 7·3%, 31·7%, 24·8% and 36·2% of the total processing time respectively. For 512 processors the relative percentages are 19·8%, 16·3%, 47·7% and 16·2% for the same 272 × 272 domain.

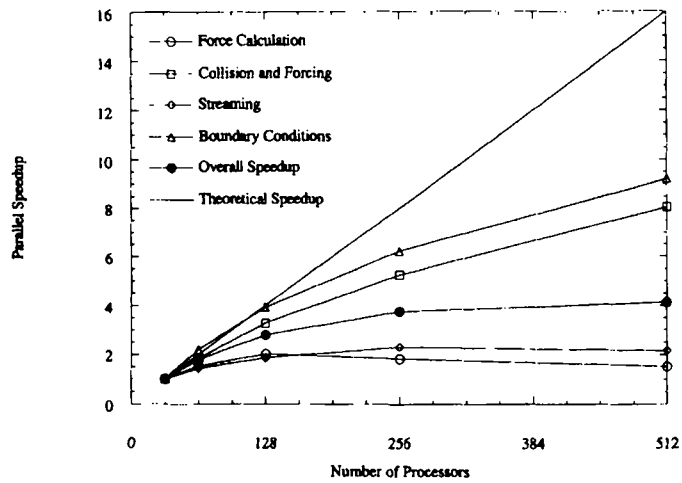

Figure 8. Parallel speed-up of each component of the lattice Boltzmann algorithm for a fixed problem size of 272 × 272

4.2.4. Scalability of lattice Boltzmann method. The scalability of an algorithm is measured by the computational speed as the number of processors is increased while simultaneously increasing the size of the computational domain. Figure 9 shows the scalability of the lattice Boltzmann code used in this study. The speed is calculated for the entire update procedure by dividing the total number of floating operations performed in 500 time steps by the total computer time for the 500 steps. A problem size of 136 × 136 is run on the 32-processor partition. Each time the number of processors is doubled, each dimension of the domain size is doubled. For the 2176 × 2176 domain solved using 512 processors, an overall speed of 13·9 GFLOPS is obtained.

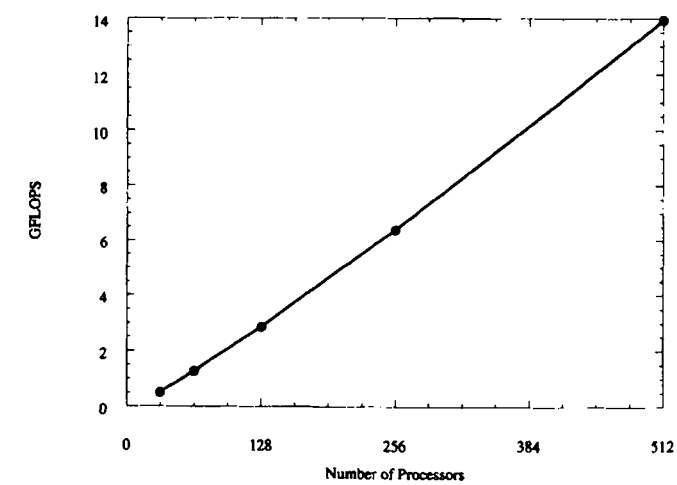

Figure 9. Scalability of the lattice Boltzmann method. The scalability is measured by the computational speed of the entire lattice Boltzmann algorithm as the number of processors and each dimension of the computational domain are simultaneously increased

5. CONCLUSIONS

The lattice Boltzmann method is shown to be an accurate and efficient scheme for computing steady recirculating flows. The flow in a periodic array of octagonal cylinders is modelled using LBM and an FD-based scheme. Through comparisons of the computed velocity profiles, LBM is shown to yield accurate velocity predictions throughout the domain, both near and far from the solid boundaries and both inside and outside the wake. Comparison of the viscous shear also demonstrates the accuracy of LBM near boundaries when utilizing the hydrodynamic boundary condition applied here. The efficiency of LBM is also established. The controllable compressibility inherent in the method is viewed as advantageous, limiting the propagation speed of disturbances and eliminating the need to solve a costly Poisson equation. The scaling of the time step with the spatial step is shown to be controllable through the computational cost and compressibility error. Overall, the LBM scheme is shown to be well suited for parallel computation. The computations involved in the collision and forcing process and those involved in the boundary condition calculation are completely local and as a result can be efficiently performed using large numbers of processors. The streaming process consists of a uniform shift of data and can be implemented using optimized communication algorithms. The lattice Boltzmann method can be used to accurately model complex fluid flows with computations in the GFLOP range using parallel architectures.

ACKNOWLEDGEMENTS

This research is supported by a National Science Foundation Graduate Fellowship, NCSA grant CBT-930030N, NSF grant CTS-9396252 and the Richard W. Kritzer Foundation. All lattice Boltzmann simulations reported here were performed on the CM-5 of the National Center for Supercomputing Applications. The authors would like to thank M. H. Shelby for his work in performing simulations and Dr. M. Wang for use of the finite difference code.

APPENDIX

The explicit equations used on the solid boundaries are developed in this appendix. For an octagonal obstacle, 20 different types of boundary orientations are needed. Each vertex requires a different set of constraints. Of the eight sides, the two horizontal surface and two vertical surfaces require one set of constraints per side. The four diagonal sides, however, require two different sets of constraints per side, one for the nodes on the boundary and one for the nodes just outside the boundary whose diagonal neighbours lie just inside the obstacle boundary. These off-boundary nodes have incomplete particle distributions, since one of their neighbours is a wall node. Unlike the boundary nodes, however, the velocity is not known at these points. The constraints given in (21)–(23) yield four equations for the four unknowns of the density, two components of velocity and the unknown component of the particle distribution. The equations, however, are non-linear, coupled, algebraic equations. Rather than acquiring the exact solution, it is approximated that the square of the magnitude of the velocity at these locations is negligible. This is well justified in the light of the close proximity of the no-slip wall. In this case the density and unknown component are computed using (21) and (23).

For each boundary geometry, equations (21)–(23) give a constraint for the density as well as constraints for each unknown component of the particle distribution. The resulting system of equations is solved to give explicit expressions for the density and particle distribution in terms of the velocity at

the boundary, the constant internal energy and the contributions of the known components of the distribution. The equations for the density and unknown components are of the form

$$\rho = \frac{A_\rho \sum\limits_{i=f,b\to b} f_i + A_{\rho u} \sum\limits_{i=f,b\to b} f_i e_{ix} + A_{\rho v} \sum\limits_{i=f,b\to b} f_i e_{iy} + A_{\rho\kappa} \sum\limits_{i=f,b\to b} f_i \dfrac{(\mathbf{e}_i \cdot \mathbf{e}_i)}{2}}{A_\rho + A_{\rho u} u + A_{\rho v} v + A_{\rho\kappa}\kappa}, \tag{35}$$

$$f_i = B_\rho \left(\rho - \sum_{i=f,b\to b} f_i \right) + B_{\rho u} \left(\rho u - \sum_{i=f,b\to b} f_i e_{ix} \right)$$

$$+ B_{\rho v} \left(\rho v - \sum_{i=f,b\to b} f_i e_{iy} \right) + B_{\rho\kappa} \left(\rho\kappa - \sum_{i=f,b\to b} f_i \frac{(\mathbf{e}_i \cdot \mathbf{e}_i)}{2} \right), \tag{36}$$

where e_{ix} and e_{iy} are the components of $\mathbf{e}_{iy}$ are the components of $\mathbf{e}_i$ in the x- and y-direction respectively and A_j and B_j are coefficients specific to each boundary geometry. Also, all quantities are evaluated at location $\mathbf{x}_b$, the boundary position, and time $t + \Delta t$. Table I lists each boundary geometry and the unknown particle distribution components for each configuration. Also listed are the coefficients A_j for the formulae used to calculate the density from the contributions of the known components of the distribution to the density, momentum and kinetic energy. Table II lists the coefficients B_j for the explicit formulae used to calculate the unknown components of the distribution.

Table I. Unknown components of the particle distribution and coefficients in (35) used to calculate the density at the boundary for each of the 20 boundary configurations

Identifier	Location	Unknowns	A_ρ	$A_{\rho u}$	$A_{\rho v}$	$A_{\rho\kappa}$
Vertex 1	$x = \frac{1}{2}, y = \frac{7}{34}$	f_1, f_2	1	−1	0	0
Vertex 2	$x = \frac{7}{34}, y = \frac{1}{2}$	f_2, f_3	1	0	−1	0
Vertex 3	$x = -\frac{7}{34}, y = \frac{1}{2}$	f_3, f_4	1	0	−1	0
Vertex 4	$x = -\frac{1}{2}, y = \frac{7}{34}$	f_4, f_5	1	1	0	0
Vertex 5	$x = -\frac{1}{2}, y = -\frac{7}{34}$	f_5, f_6	1	1	0	0
Vertex 6	$x = -\frac{7}{34}, y = -\frac{1}{2}$	f_6, f_7	1	0	1	0
Vertex 7	$x = \frac{7}{34}, y = -\frac{1}{2}$	f_7, f_8	1	0	1	0
Vertex 8	$x = \frac{1}{2}, y = -\frac{7}{34}$	f_8, f_1	1	−1	0	0
Side 1, type 1	$\frac{7}{34} < x < \frac{1}{2}, y = -x + \frac{12}{17}$	f_1, f_2, f_3	0	−1	−1	2
Side 2, type 1	$-\frac{7}{34} < x < \frac{7}{34}, y = \frac{1}{2}$	f_2, f_3, f_4	1	0	−1	0
Side 3, type 1	$-\frac{1}{2} < x < -\frac{7}{34}, y = x + \frac{12}{17}$	f_3, f_4, f_5	0	1	−1	2
Side 4, type 1	$x = -\frac{1}{2}, -\frac{7}{34} < y < \frac{7}{34}$	f_4, f_5, f_6	1	1	0	0
Side 5, type 1	$-\frac{1}{2} < x < -\frac{7}{34}, y = -x - \frac{12}{17}$	f_5, f_6, f_7	0	1	1	2
Side 6, type 1	$-\frac{7}{34} < x < \frac{7}{34}, y = -\frac{1}{2}$	f_6, f_7, f_8	1	0	1	0
Side 7, type 1	$\frac{7}{34} < x < \frac{1}{2}, y = x - \frac{12}{17}$	f_7, f_8, f_1	0	−1	1	2
Side 8, type 1	$x = \frac{1}{2}, -\frac{7}{34} < y < \frac{7}{34}$	f_8, f_1, f_2	1	−1	0	0
Side 1, type 2	Just outside of side 1	f_2	−2	0	0	1
Side 3, type 2	Just outside of side 3	f_4	−2	0	0	1
Side 5, type 2	Just outside of side 5	f_6	−2	0	0	1
Side 7, type 2	Just outside of side 7	f_8	−2	0	0	1

Table II. Coefficients in (36) used to calculate the unknown components of the particle distribution for each of the 20 boundary configurations

Location	Unknown	B_ρ	$B_{\rho u}$	$B_{\rho v}$	$B_{\rho \kappa}$
Vertex 1	f_1	1	0	−1	0
	f_2	0	0	1	0
Vertex 2	f_2	0	1	0	0
	f_3	1	−1	0	0
Vertex 3	f_3	1	1	0	0
	f_4	0	−1	0	0
Vertex 4	f_4	0	0	1	0
	f_5	1	0	−1	0
Vertex 5	f_5	1	0	1	0
	f_6	0	0	−1	0
Vertex 6	f_6	0	−1	0	0
	f_7	1	1	0	0
Vertex 7	f_7	1	−1	0	0
	f_8	0	1	0	0
Vertex 8	f_8	0		−1	0
	f_1	1	0	1	0
Side 1, type 1	f_1	1	0	−1	0
	f_2	−1	1	1	0
	f_3	1	−1	0	0
Side 2, type 1	f_2	$-\frac{1}{2}$	$\frac{1}{2}$	0	1
	f_3	2	0	0	−2
	f_4	$-\frac{1}{2}$	$-\frac{1}{2}$	0	1
Side 3, type 1	f_3	1	1	0	0
	f_4	−1	−1	1	0
	f_5	1	0	−1	0
Side 4, type 1	f_4	$-\frac{1}{2}$	0	$\frac{1}{2}$	1
	f_5	2	0	0	−2
	f_6	$-\frac{1}{2}$	0	$-\frac{1}{2}$	1
Side 5, type 1	f_5	1	0	1	0
	f_6	−1	−1	−1	0
	f_7	1	1	0	0
Side 6, type 1	f_6	$-\frac{1}{2}$	$-\frac{1}{2}$	0	1
	f_7	2	0	0	−2
	f_8	$-\frac{1}{2}$	$\frac{1}{2}$	0	−1
Side 7, type 1	f_7	1	−1	0	0
	f_8	−1	1	−1	0
	f_1	1	0	1	0
Side 8, type 1	f_8	$-\frac{1}{2}$	0	$-\frac{1}{2}$	1
	f_1	2	0	0	−2
	f_2	$-\frac{1}{2}$	0	$\frac{1}{2}$	1
Side 1, type 2	f_2	1	0	0	0
Side 3, type 2	f_4	1		0	0
Side 5, type 2	f_6	1	0	0	0
Side 7, type 2	f_8	1	0	0	0

REFERENCES

1. G. McNamara and G. Zanetti, 'Use of the Boltzmann equation to simulate lattice-gas automata', *Phys. Rev. Lett.*, **61**, 2332–2335 (1988).
2. F. Higuera and J. Jimenez, 'Lattice gas dynamics with enhanced collisions', *Europhys. Lett.*, **9**, 663–668 (1989).
3. H. Chen, S. Chen and W. H. Matthaeus, 'Recovery of the Navier–Stokes equations using a lattice Boltzmann method', *Phys. Rev. A*, **45**, R5339–R5342 (1991).
4. S. Y. Chen, H. D. Chen, D. Martinez and W. Matthaeus, 'Lattice Boltzmann model for simulation of magnetohydrodynamics', *Phys. Rev. Lett.*, **67**, 3776–3779 (1991).
5. Y. H. Qian, D. d'Humières and P. Lallemand, 'Lattice BGK models for the Navier–Stokes equation', *Europhys. Lett.*, **17**, 479–484 (1992).
6. S. Succi, R. Benzi and F. Higuera, 'The lattice Boltzmann equation: a new tool for computational fluid-dynamics', *Physica D*, **47**, 219–230 (1991).
7. S. Chen, Z. Wang, X. Shan and G. D. Doolen, 'Lattice Boltzmann computational fluid dynamics in three directions', *J. Stat. Phys.*, **68**, 379–400 (1992).
8. D. O. Martinez, W. H. Matthaeus, S. Y. Chen and D. C. Montgomery, 'Comparison of spectral method and lattice Boltzmann simulations of two-dimensional hydrodynamics', *Phys. Fluids*, **6**, 1285–1298 (1994).
9. A. J. C. Ladd, 'Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part II. Numerical results', *J. Fluid Mech.*, **271**, 311–339 (1994).
10. M. B. Reider and J. D. Sterling, 'Accuracy of discrete-velocity BGK models for the simulation of the incompressible Navier–Stokes equations', *Comput. Fluids*, **24**, 459–467 (1995).
11. S. Hou, Q. Zou, S. Chen, G. D. Doolen and A. C. Cogley, 'Simulation of cavity flow by the lattice Boltzmann method', *J. Comput. Phys.*, **118**, 329–347 (1995).
12. D. d'Humières and P. Lallemand, 'Numerical simulations of hydrodynamics with lattice gas automata in two dimensions', *Complex Syst.*, **1**, 599–632 (1987).
13. R. Cornubert, D. d'Humières and D. Levermore, 'A Knudsen layer theory for lattice gases', *Physica D*, **47**, 241–259 (1991).
14. I. Ginzbourg and P. M. Adler, 'Boundary flow condition analysis for the three-dimensional lattice Boltzmann model', *J. Phys. II France*, **4**, 191–214 (1994).
15. D. P. Ziegler, 'Boundary conditions for lattice Boltzmann simulations', *J. Stat. Phys.*, **71**, 1171–1177 (1993).
16. P. A. Skordos, 'Initial and boundary conditions for the lattice Boltzmann method', *Phys. Rev. E.*, **48**, 4823–4842 (1993).
17. D. R. Noble, S. Chen, J. G. Georgiadis and R. O. Buckius, 'A consistent hydrodynamic boundary condition for the lattice Boltzmann method', *Phys. Fluids*, **7**, 203–209 (1995).
18. D. R. Noble, J. G. Georgiadis and R. O. Buckius, 'Direct assessment of lattice Boltzmann hydrodynamics and boundary conditions for recirculating flows', *J. Stat. Phys.*, in press.
19. S. Chen, K. Diemer, G. D. Doolen, K. Eggert, C. Fu, S. Gutman and B. J. Travis, 'Lattice gas automata for flow through porous media', *Physica D*, **47**, 2–84 (1991).
20. F. M. Najjar and S. P. Vanka, 'Numerical study of separated–reattaching flow', *Theor. Comput. Fluid Dyn.*, **5**, 291–308 (1993).
21. J. K. Dukowicz, 'Computational efficiency of the hybrid penalty–pseudocompressibility method for incompressible flow', *Comput. Fluids*, **2**, 479–486 (1994).
22. M. Wang and J. G. Georgiadis, 'Parallel computation of forced convection using domain composition', *Numer. Heat Transfer B*, **20**, 41–59 (1991).
23. P. Bhatnagar, E. P. Gross and M. K. Krook, 'A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems', *Phys. Rev.*, **94**, 511–525 (1954).
24. G. K. Batchelor, *An Introduction to Fluid Dynamics*. Cambridge University Press, Cambridge, 1967.