Springer

Journal of
Mechanical
Science and
Technology

Journal of Mechanical Science and Technology 22 (2008) 1192~1201

www.springerlink.com/content/1738-494x

A dynamic boundary model for implementation of boundary conditions in lattice-Boltzmann method

Jinfen Kang, Sangmo Kang and Yong Kweon Suh*

Department of Mechanical Engineering, Dong-A University, 840, Hadan-dong, Saha-gu, Busan, 604-714, Korea

(Manuscript Received August 6, 2007; Revised March 4, 2008; Accepted March 11, 2008)

Abstract

Methodology of implementation of the no-slip boundary condition in the lattice-Boltzmann method affects overall accuracy of the numerical solutions as well as the stability of the solution procedure. We propose a new algorithm, i.e., the method of using a dynamic equation for establishing no-slip boundary conditions on walls. The distribution functions on the wall along each of the links across the physical boundary are assumed to be composed of equilibrium and non-equilibrium parts which inherit the idea of Guo's extrapolation method (2002). In the proposed algorithm, we employed a dynamic equation to correct the velocity error occurring on the physical boundary. Numerical results show that the dynamic boundary model is featured with improved accuracy and simplicity. The proposed method is postulated to be useful especially in the study on microfluidic mixing.

Keywords: Dynamic-equation method; Lattice-Boltzmann method; No-slip boundary condition

1. Introduction

Lattice-Boltzmann method (LBM) has become a promising alternative fluid-dynamic computational platform aside from the traditional CFD method due to its simplicity for implementation and ease in handling complex boundary conditions. However, in the development of LBM, there are still several problems open to further improvement. The correct implementation of the no-slip boundary condition, among others, is one of the crucial techniques to improve, as it plays an important role in the overall accuracy of the numerical solutions as well as the stability of the solution procedure. There are various approximate methods for the treatment of no-slip boundary condition. Among them, the most representative methods are the bounce-back method [1-4], Yu's interpolation method [5] and Guo's extrapolation method [6]. The bounce-back scheme is particularly simple and has played a major role in making LBM popular among CFD researchers, particularly as applied to the porous-media flow. However, the bounce-back method produces results with only first-order spatial accuracy [7] unless a proper spatial arrangement of the boundary location (i.e., placing the boundary wall just halfway between the grid nodes) is established [3]. By applying Yu's and Guo's method, we can get more accurate results, but the stability of solution is not always satisfying, especially when applied to microfluidic flows that are featured with low Reynolds numbers and low velocity values; in LBM simulation, such parameter settings necessitate large value of relaxation time τ, which usually causes a stability problem.

In order to enhance the accuracy and numerical stability of LBM, particularly for dealing with microfluidic flows, we developed a new model, i.e., dynamic boundary model. The present study is devoted to the feasibility of this dynamic treatment method. Accuracy and stability studies have been carried out on 2-D Poiseuille flow, oscillating Couette flow between two parallel planes, Couette flow between two circular cylinders and lid-driven cavity flow.

*Corresponding author. Tel.: +82 51 200 7648, Fax.: +82 51 200 7656
E-mail address: yksuh@dau.ac.kr
DOI 10.1007/s12206-008-0307-y

2. Background

2.1 Lattice-Boltzmann method

The origin of LBM traces back to the idea that one can construct the fluid flows from a group of fictitious particles having identical mass and speed, but different velocity directions. They have motions like suspended pollens in a cup of water. They can move, change their positions, collide with each other and exchange their momentum. The initial model to realize this idea is the lattice gas cellular automata (LGCA) which has been gradually developed to LBM. In LBM, the statistical concept, i.e., particle mass distribution function, f, is introduced to replace the real particle in LGCA; the momentum, density, velocity and pressure, etc., can be obtained from the evolution of f [4].

During more than two decades, LBM has been developed to several model classes adapted to different fluid applications. In the present study, we employed the D2Q9 (two-dimensional nine-velocity lattice type shown in Fig. 1) LBGK model [8], and applied the modified equilibrium distribution function which was proposed by He and Luo [9] for incompressible flows. The distribution function is determined by the following evolution equation.

$$f_\alpha(\mathbf{x}+\mathbf{e}_\alpha\delta t, t+\delta t) - f_\alpha(\mathbf{x},t) = -\frac{1}{\tau}\left[f_\alpha(\mathbf{x},t) - f_\alpha^{eq}(\mathbf{x},t)\right] \quad (1)$$

In the above equation, $\mathbf{x}$ is the coordinate of the lattice node of interest, t the time, δt the time step (in the present simulation, $\delta t = 1$), τ the dimensionless relaxation time, $\alpha = 0$, 1, 2,…, 8, the link number around a nodal point in the lattice system, and $\mathbf{e}_\alpha$ the discrete velocity vector, which for D2Q9 lattice space is

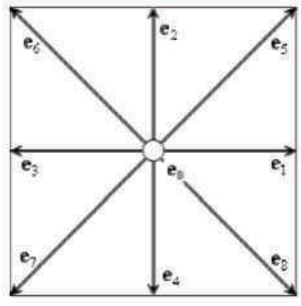

Fig. 1. Discrete velocity vectors of D2Q9 lattice.

$$\mathbf{e}_\alpha = \begin{cases} (0,0) & \alpha = 0 \\ (1,0),(0,1),(-1,0),(0,-1) & \alpha = 1,2,3,4 \\ (1,1),(-1,1),(1,-1),(-1,-1) & \alpha = 5,6,7,8 \end{cases} \quad (2)$$

Further, f_α is the particle mass distribution function, and f_α^{eq} the equilibrium distribution function along the link number α at a node of interest given by

$$f_\alpha^{eq} = w_\alpha\left\{\rho + \rho_0\left[\frac{1}{c_s^2}\mathbf{e}_\alpha\cdot\mathbf{u} + \frac{1}{2c_s^4}(\mathbf{e}_\alpha\cdot\mathbf{u})^2 - \frac{1}{2c_s^2}(\mathbf{u}\cdot\mathbf{u})\right]\right\} \quad (3)$$

where, $\mathbf{u}$ is the fluid velocity, ρ the fluid density, ρ_0 the reference density (constant), c_s the sound speed, and w_α the weighting factor; $w_\alpha = 4/9$ for $\alpha = 0$, $w_\alpha = 1/9$ for $\alpha = 1,2,3,4$, and $w_\alpha = 1/36$ for $\alpha = 5,6,7,8$.

The fluid density and mass flux can be evaluated by the following formulae:

$$\rho = \sum_{\alpha=0}^{8} f_\alpha = \sum_{\alpha=0}^{8} f_\alpha^{eq} \quad (4)$$

$$\rho\mathbf{u} = \sum_{\alpha=0}^{8} \mathbf{e}_\alpha f_\alpha = \sum_{\alpha=0}^{8} \mathbf{e}_\alpha f_\alpha^{eq} \quad (5)$$

During one-time-step computation, Eq. (1) can be solved by two sub-steps, i.e., collision and streaming;

(i) collision step:

$$\tilde{f}_\alpha(\mathbf{x},t) = f_\alpha(\mathbf{x},t) - \frac{1}{\tau}\left[f_\alpha(\mathbf{x},t) - f_\alpha^{eq}(\mathbf{x},t)\right] \quad (6)$$

(ii) streaming step:

$$f_\alpha(\mathbf{x}+\mathbf{e}_\alpha\delta t, t+\delta t) = \tilde{f}_\alpha(\mathbf{x},t) \quad (7)$$

where ~ denotes the post-collision state of the distribution function. It is noted that the collision step is local and the streaming step involves no special computation.

2.2 Boundary condition methods in literature

In general, to finish the streaming step, the distribution functions at the solid nodes near the physical boundaries, $\tilde{f}_{\bar{\alpha}}(\mathbf{x}_s,t)$ (see Fig. 2), need to be speci-

fied after the collision step. We briefly describe here the three typical methods that are used in evaluating these functional values for the purpose to comparatively evaluate the dynamic boundary method; bounce-back method [4], Yu's interpolation method [5], and Guo's extrapolation method [6].

Bounce-back is the most popular and simplest scheme. If a node is included in boundary solid, the normal collision computation is omitted and the distribution functions are bounced back. The subsequent streaming step brings f_α back into the fluid domain. For a stationary wall, it is equivalent to setting $\tilde{f}_{\bar{\alpha}}(\mathbf{x}_s,t)=\tilde{f}_\alpha(\mathbf{x}_1,t)$, where α denotes the velocity direction toward the physical boundary of the fluid domain, and $\bar{\alpha}$ the opposite direction (Fig. 2). For a moving wall, a certain amount of momentum should be added to the bounced particle distribution function:

$$\tilde{f}_{\bar{\alpha}}(\mathbf{x}_s,t)=\tilde{f}_\alpha(\mathbf{x}_1,t)+6w_\alpha\rho_w\mathbf{e}_\alpha\cdot\mathbf{u}_w^e \tag{8}$$

where ρ_w is the fluid density at the wall, $\mathbf{u}_w^e$ the desired exact velocity specified on the boundary wall node 'W'.

In Yu's interpolation scheme [5], one obtains $f_{\bar{\alpha}}(\mathbf{x}_1,t)$ on node '1' after streaming step which is $\tilde{f}_{\bar{\alpha}}(\mathbf{x}_s,t-\delta t)$ brought from the node 'S' during the streaming step. Yu's scheme is constructed based on the principle of momentum balance to ensure the no-slip condition ($\mathbf{u}_w=0$) on the wall, considering the momentum balance in the direction of α:

$$f_{\bar{\alpha}}(\mathbf{x}_w,t)=f_\alpha(\mathbf{x}_w,t) \tag{9}$$

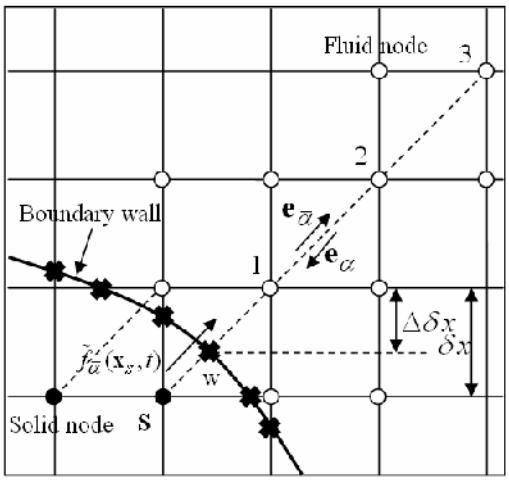

Fig. 2. Layout of the lattices near a curved boundary.

A simple interpolation is then used to obtain the distribution functions on the boundary wall node $f_\alpha(\mathbf{x}_w,t)$ given as

$$f_\alpha(\mathbf{x}_w,t)=f_\alpha(\mathbf{x}_1,t)+\Delta\left[f_\alpha(\mathbf{x}_s,t)-f_\alpha(\mathbf{x}_1,t)\right] \tag{10}$$

Here, Δ is the fraction of an intersected link in the fluid region given by

$$\Delta=\frac{\left|\mathbf{x}_1-\mathbf{x}_w\right|}{\left|\mathbf{x}_1-\mathbf{x}_s\right|},\qquad 0\le\Delta\le 1 \tag{11}$$

Using $f_{\bar{\alpha}}(\mathbf{x}_w,t)$ and $f_{\bar{\alpha}}(\mathbf{x}_1+e_{\bar{\alpha}}\delta t,t)$, one obtains $f_{\bar{\alpha}}(\mathbf{x}_1,t)$ using a linear interpolation:

$$f_{\bar{\alpha}}(\mathbf{x}_1,t)=f_{\bar{\alpha}}(\mathbf{x}_w,t)+\frac{\Delta}{1+\Delta}\left[f_{\bar{\alpha}}(\mathbf{x}_1+e_{\bar{\alpha}}\delta t,t)-f_{\bar{\alpha}}(\mathbf{x}_w,t)\right] \tag{12}$$

If the boundary is driven by non-zero wall velocity, $\mathbf{u}_w$, then an extra momentum term is added to $f_{\bar{\alpha}}(\mathbf{x}_w,t)$ in Eq. (9):

$$f_{\bar{\alpha}}(\mathbf{x}_w,t)=f_\alpha(\mathbf{x}_w,t)+6w_\alpha\rho_w\mathbf{e}_\alpha\cdot\mathbf{u}_w \tag{13}$$

For Guo's extrapolation method [6], the distribution function at a solid node near the wall $\tilde{f}_{\bar{\alpha}}(\mathbf{x}_s,t)$ is decomposed into two parts, the equilibrium part and the non-equilibrium one.

$$\tilde{f}_\alpha(\mathbf{x}_s,t)=f_\alpha^{eq}(\mathbf{x}_s,t)+f_\alpha^{ne}(\mathbf{x}_s,t) \tag{14}$$

The non-equilibrium part $f_{\bar{\alpha}}^{ne}(\mathbf{x}_s,t)$ is approximated from those of the neighboring fluid nodes along the link α by using $f_{\bar{\alpha}}^{ne}(\mathbf{x}_s,t)=f_{\bar{\alpha}}^{ne}(\mathbf{x}_1,t)$ for $\Delta\ge 0.75$ and $f_{\bar{\alpha}}^{ne}(\mathbf{x}_s,t)=\Delta f_{\bar{\alpha}}^{ne}(\mathbf{x}_1,t)+(1-\Delta)f_{\bar{\alpha}}^{ne}(\mathbf{x}_2,t)$ for $\Delta<0.75$. The equilibrium part is determined by a fictitious equilibrium distribution

$$f_{\bar{\alpha}}^{eq}(\mathbf{x}_s,t)=w_\alpha\left\{\overline{\rho}_s+\rho_0\left[\frac{1}{c_s^2}\mathbf{e}_{\bar{\alpha}}\cdot\mathbf{u}_s+\frac{1}{2c_s^4}(\mathbf{e}_{\bar{\alpha}}\cdot\mathbf{u}_s)^2-\frac{1}{2c_s^2}(\mathbf{u}_s\cdot\mathbf{u}_s)\right]\right\} \tag{15}$$

where ρ_s is the fluid density on the solid node 'S', being approximated to be equal to ρ_1 (ρ at the node '1' in Fig. 2), and $\mathbf{u}_s$ the fictitious velocity on the solid node near the boundary, is determined by a linear extrapolation using

$$\mathbf{u}_{s1} = \left(\mathbf{u}_w + (\Delta - 1)\mathbf{u}_1\right)/\Delta \tag{16a}$$

$$\mathbf{u}_{s2} = \left(2\mathbf{u}_b + (\Delta - 1)\mathbf{u}_2\right)/(1+\Delta) \tag{16b}$$

It is proposed to use $\mathbf{u}_s = \mathbf{u}_{s1}$ for $\Delta \geq 0.75$, and use a linear interpolation from $\mathbf{u}_{s1}$ and $\mathbf{u}_{s2}$ for $\Delta < 0.75$, i.e., $\mathbf{u}_s = \Delta\mathbf{u}_{s1} + (1-\Delta)\mathbf{u}_{s2}$. The boundary-condition enforcement is completed by the evaluation of the post-collision distribution function as follows:

$$\tilde{f}_{\bar{\alpha}}(\mathbf{x}_s, t) = f^{eq}_{\bar{\alpha}}(\mathbf{x}_s, t) + (1-\tau^{-1}) f^{ne}_{\bar{\alpha}}(\mathbf{x}_s, t) \tag{16}$$

3. Dynamic boundary model

Similarly to Guo's extrapolation method, in our new method we also decomposed the distribution function at a solid wall node along the links across the physical boundary $\tilde{f}_{\alpha}(\mathbf{x}_s, t)$ into equilibrium and non-equilibrium parts like Eq. (14).

The equilibrium part is approximated by Eq. (15). Now, in order to obtain the fictitious velocity on the solid nodes near the physical boundary, $\mathbf{u}_s$, we employed a dynamic equation given by

$$\frac{d\mathbf{u}_s}{dt} = r(\mathbf{u}^e_w - \mathbf{u}^o_w) \tag{18}$$

where $\mathbf{u}^e_w$ is the desired exact velocity specified on the boundary wall node 'W' and $\mathbf{u}^o_w$ the calculated boundary-wall velocity. Eq. (18) can also be written in a discrete form as follows:

$$\mathbf{u}_s = \mathbf{u}^o_s + r\delta t(\mathbf{u}^e_w - \mathbf{u}^o_w) \tag{19}$$

Here, r is a relaxation factor, which affects the convergence property of the computation.

According to this dynamic equation, if $\mathbf{u}^o_w$ is larger than the desired wall velocity $\mathbf{u}^e_w$, the solid node velocity $\mathbf{u}_s$ will be decreased, and vice versa. Then, the value of fictitious velocity on the solid nodes $\mathbf{u}_s$ can be automatically corrected every time step until the numerical wall velocity $\mathbf{u}_w$ matches the exact velocity $\mathbf{u}^e_w$.

For code implementation, the present wall velocity $\mathbf{u}^o_w$ can be computed by using an extrapolation method with the given velocities at nodes '1', '2' and '3' (illustrated in Fig. 2). We use one of the following three algorithms for the extrapolation depending on the velocity data available on the nearby nodes:

$$\begin{cases} \text{0th-order:} \\ \mathbf{u}^o_w = \mathbf{u}^o_1 \\ \text{linear extrapolation:} \\ \mathbf{u}^o_w = (1+\Delta)\mathbf{u}^o_1 - \Delta\mathbf{u}^o_2 \\ \text{quadratic extrapolation:} \\ \mathbf{u}^o_w = \dfrac{1}{2}(1+\Delta)(2+\Delta)\mathbf{u}^o_1 - \Delta(2+\Delta)\mathbf{u}^o_2 + \dfrac{1}{2}\Delta(1+\Delta)\mathbf{u}^o_3 \end{cases} \tag{20}$$

In most cases, since both $\mathbf{u}_3$ and $\mathbf{u}_2$ are available, we use the quadratic extrapolation algorithm.

The non-equilibrium part $f^{ne}_{\alpha}(\mathbf{x}_s, t)$ is approximated by using a first-order extrapolation scheme based on the non-equilibrium distribution functions on the neighboring fluid nodes:

$$f^{ne}_{\alpha}(\mathbf{x}_s, t) = \beta f^{ne}_{\alpha}(\mathbf{x}_1, t) + (1-\beta) f^{ne}_{\alpha}(\mathbf{x}_2, t) \tag{21}$$

where β is a parameter to be determined.

4. Computational assessment

For assessment of the proposed dynamic boundary model, we studied several 2-D fluid flow problems, including Poiseuille flow, oscillating Couette flow between two planar walls and rotating Couette flow between two circular cylinders. We applied the dynamic method as well as other methods to these flows, compared the results and checked the spatial, temporal accuracy and the numerical stability. Lid-driven cavity flow is also studied to test the ability of handling fluid problems with geometric singularity.

To evaluate the computational accuracy of the results, we made use of two quantities,: the wall-slip velocity error U_S and the rms error in the velocity profile ε_{rms}. The wall-slip velocity error U_S is given by

$$U_S = \frac{\left|\mathbf{u}_w - \mathbf{u}^e_w\right|}{U_0} \tag{22}$$

Here, $\mathbf{u}_w$ is the numerical velocity on the boundary wall which can be obtained by the quadratic extrapolation (Eq. (20)). As $\mathbf{u}^e_w$ is initially known and fixed, the difference between $\mathbf{u}_w$ and $\mathbf{u}^e_w$ provides a measure of the accuracy of the method. The rms error ε_{rms} is calculated by

$$\varepsilon_{rms} = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left[\mathbf{u}^e_w(j) - \mathbf{u}(j)\right]^2 / \left|\mathbf{u}\right|^2_{\max}} \tag{23}$$

It reflects the root-mean-square of the difference between the normalized velocity obtained form the numerical computation and that from the analytic solution.

Implementing the no-slip boundary conditions on the solid walls and evaluating the performance of different methods is our main purpose in this study. We applied the proposed dynamic method to several flow models and compared the accuracy as well as the numerical stability with other three methods: bounce-back scheme, Yu's interpolation method and Guo's extrapolation method.

4.1 2-D poiseuille flow

The grid system for 2-D Poiseuille flow is specified as illustrated in Fig. 3. The boundary conditions for this flow are set as follows. For the left inlet, we applied a parabolic profile

$$u_{in}(y) = 4U_0 y(H-y)/H^2 \tag{24}$$

which corresponds to the exact velocity profile of the fully-developed Poiseuille flow. Here, y is the vertical coordinate with its origin at the bottom wall, its discrete value on the node j being given by $y_j = j-1+\Delta$, $H = N_y - 1 + 2\Delta$ the height of the fluid domain, N_y the grid number along y direction, and U_0 the given maximum inlet velocity which was set at $U_0 = 0.01$ in the lattice unit. Each of the boundary walls is located at exactly halfway between two grid lines, i.e., $\Delta = 0.5$. As the outlet boundary condition, zero gradient was imposed on the main stream velocity. We applied no-slip boundary condition at the top and bottom solid walls by using different treatment methods and the same relaxation time $\tau = 5$. In applying the present method, we have special parameters to be determined, relaxation factor r in (18) and coefficient β in (21). These two parameters affect the convergence property and accuracy of the whole numerical computation. For 2-D Poiseuille flow, we chose $r = 0.1$ and $\beta = 0$.

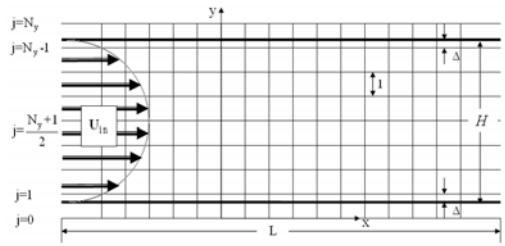

Fig. 3. Lattice system for 2-D Poiseuille flow.

Fig. 4 shows the variation of U_S with respect to different number of grids N_y obtained by applying four different boundary-treatment methods. The value of U_S given by the present method is seen to be much smaller than those by other schemes. In fact, it is on the order of 10^{-15} which can be considered as the computer's machine error. So we suppose that there is almost no numerical wall-slip velocity error by applying the present method to the Poiseuille flow.

Fig. 5 shows ε_{rms} obtained by applying four different methods. We can see that the magnitude of ε_{rms} by the present method is smaller than the others by several orders of magnitude. In Fig. 6, ε_{rms} is also checked with different positions of physical wall Δ on the lattice. The variation of ε_{rms} by the present method not only shows small error values but also provides much less dependence on Δ.

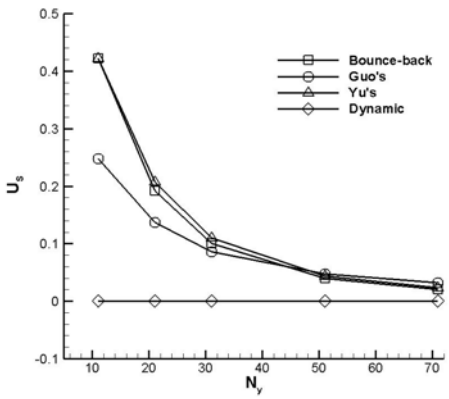

Fig. 4. Variation of wall-slip velocity error given by the dynamic method in comparison with other three methods in Poiseuille flow ($\Delta = 0.5$, $\tau = 5$, $U_0 = 0.01$, $r = 0.1$, $\beta = 0$).

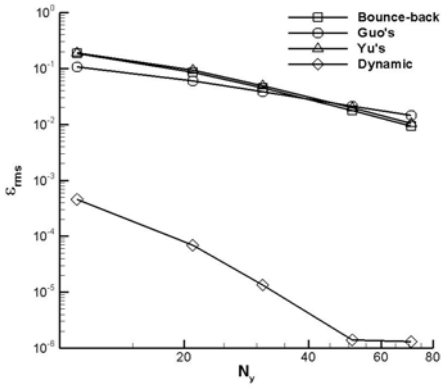

Fig. 5. Variation of ε_{rms} error given by the dynamic method in comparison with other three methods in Poiseuille flow ($\Delta = 0.5$, $\tau = 5$, $U_0 = 0.01$, $r = 0.1$, $\beta = 0$).

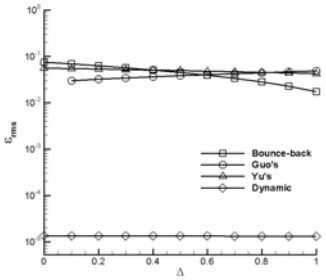

Fig. 6. Dependence of ε_{rms} on Δ representing the boundary wall position in Poiseuille flow ($\tau = 5$, $U_0 = 0.01$, $N_y = 31$, $r = 0.1$, $\beta = 0$).

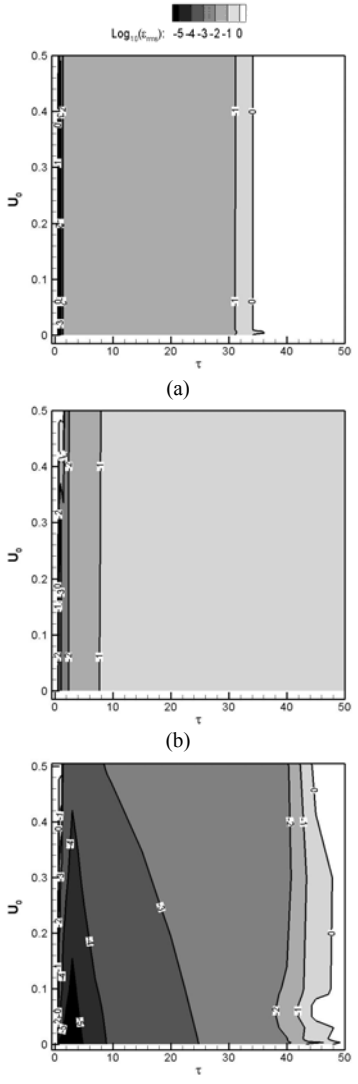

Fig. 7. Stability represented by ε_{rms} depending on τ and U_0 in Poiseuille flow; (a) Guo's method, (b) Yu's method, and (c) dynamic method ($\Delta = 0.5$, $N_y = 31$, $r = 0.1$, $\beta = 0$).

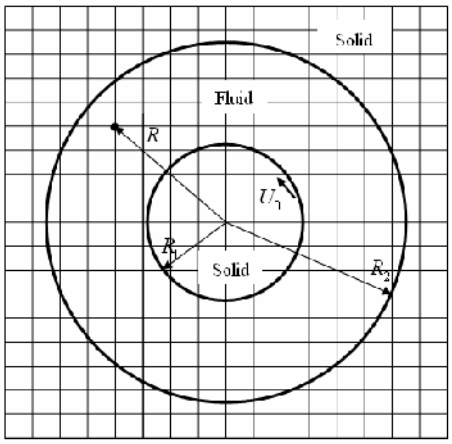

Fig. 8. Lattice system for Couette flow between two circular cylinders.

We evaluated the numerical stability of the present and other methods in terms of the ε_{rms} -contour in the parameter space (τ, U_0) as shown in Fig. 7. In this figure, the gray color contours represent the level of error ε_{rms} ; the dark color denotes a high numerical accuracy and light/white color a low accuracy including even the numerical overflow. The present method shows superiority as the dark color dominates large range of the value of τ and U_0 .

4.2 Couette flow between two circular cylinders

In Couette flow between two circular cylinders, the inner circle of radius R_1 is rotating with linear velocity $U_0 = 0.01$ and the outer circle of radius R_2 is stationary. The grid system is fixed at 95×95 . For such a Couette flow, we have an analytical solution given by

$$u_\theta(R) = \frac{U_0 \kappa}{1-\kappa^2}\left(\frac{R_2}{R} - \frac{R}{R_2}\right) \tag{25}$$

where $u_\theta(R)$ is the velocity component along the tangential direction, R the radial coordinate and κ the ratio of inner and outer circle radius ($\kappa = R_1 / R_2$). We fixed $R_2 = 42.5$, and adjusted κ to control the geometry.

Figs. 9 and 10 show ε_{rms} obtained by applying different boundary-treatment methods at $\tau = 0.6$ (Fig. 9) and $\tau = 1.5$ (Fig. 10). The parameters r and β for the dynamic method are set at $r = 0.0001$ and $\beta = 0$. These results indicate that the advantage of the dynamic boundary model

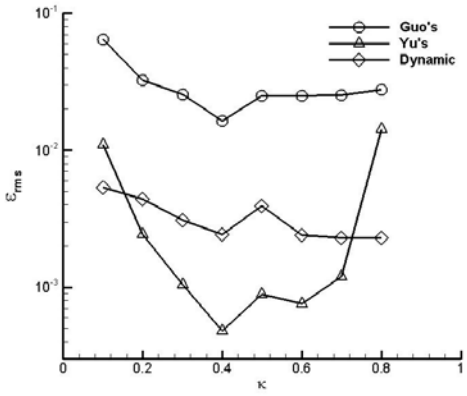

Fig. 9. Comparison of ε_{rms} among different methods in Couette flow at $\tau = 0.6$ ($\mathrm{Re} = 10$, $r = 0.0001$, $\beta = 0$).

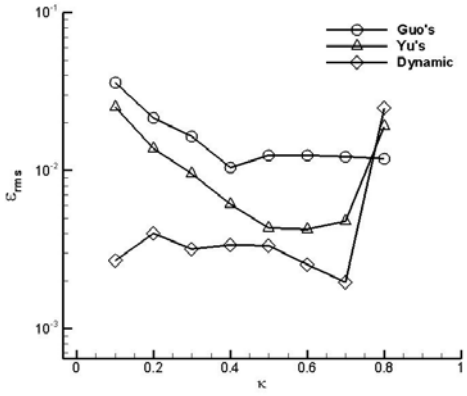

Fig. 10. Comparison of ε_{rms} among different methods in Couette flow at $\tau = 1.5$ ($\mathrm{Re} = 10$, $r = 0.0001$, $\beta = 0$).

appears more distinctive for large τ values. Fig. 11 shows the distribution of ε_{rms} in the parameter space (τ, U_0). Considering that the numerical results with ε_{rms} less than 0.01 are acceptable, we can see that the dynamic method is more robust than Yu's and Guo's method in treating the boundary conditions of Couette flow between two circular cylinders.

4.3 Oscillating couette flow

For an oscillating Couette flow between two parallel planes, the channel geometry is set at $L = 4H + 1$ (Fig. 12). The boundary wall is located halfway between two neighboring horizontal grid lines, i.e., we set $\Delta = 0.5$. The bottom wall of the channel oscillates with velocity

$$u(0,t) = U_0 \sin(\omega t) \tag{26}$$

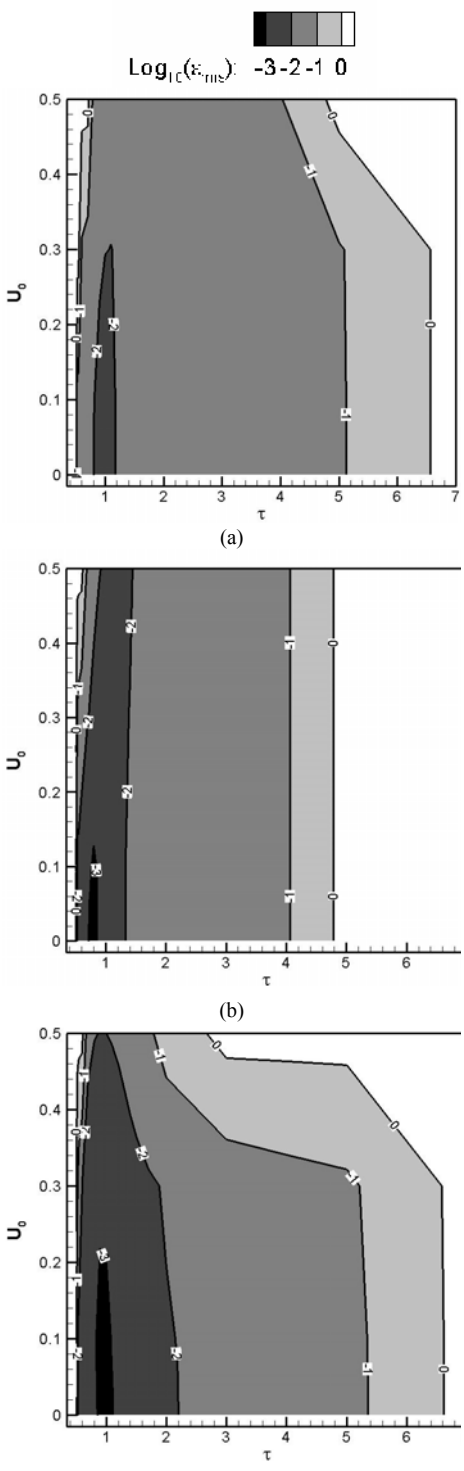

Fig. 11. Stability represented by ε_{rms} depending on τ and U_0 in Couette flow; (a) Guo's method, (b) Yu's method, and (c) dynamic method ($\kappa = 0.2$, $r = 0.0001$, $\beta = 0$).

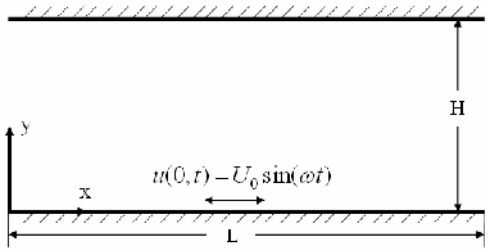

Fig. 12. Sketch of oscillating Couette flow.

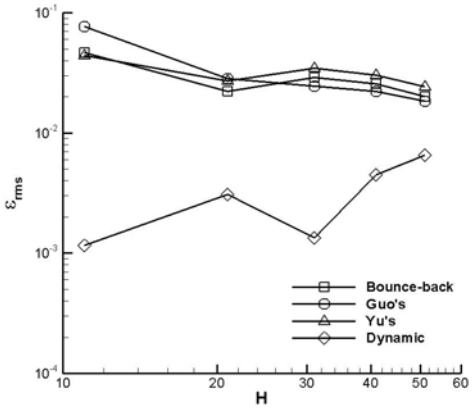

Fig. 13. Comparison of ε_{rms} among different methods in oscillating Couette flow ($U_0=0.01$, $\tau=5$, $r=1$, $\beta=0$).

Under such an arrangement, the flow velocity has an exact solution given by

$$u_{exa}=U_0 e^{-\lambda y}\sin(\omega t-\lambda y) \tag{27}$$

where $\lambda=\sqrt{\omega/2\upsilon}$ and υ is the kinetic viscosity in lattice unit. In the study, we set $\omega=0.01$, $U_0=0.01$ and $\tau=5$. For the dynamic method, we selected $r=1$ and $\beta=0$. The numerical results shown by ε_{rms} in Fig. 13 indicate that the present method is more accurate in treating this fluid problem too than others.

4.4 Lid-driven cavity flow

In order to check the flexibility of the present method in solving the standard flow problem but having corner singularities, we employed it to the lid-driven cavity flow (Fig. 14). We applied Guo's and present methods in implementing the no-slip boundary conditions for stationary side and bottom walls and $U_0=0.01$ for the sliding top wall. In simulation, the grid system is fixed at 51×51, the parameters at $\Delta=0.5$, $\tau=0.8$ and $\mathrm{Re}=5$ in both methods and

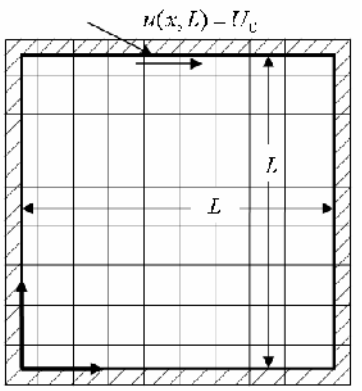

Fig. 14. Lattice system for 2-D lid-driven cavity flow.

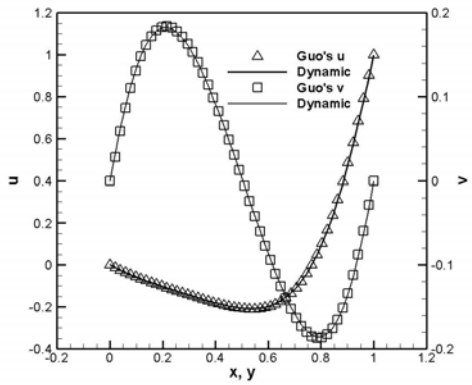

Fig. 15. Comparison of u and v velocities along vertical and horizontal centerline of lid-driven cavity flow between Guo's and dynamic methods ($U_0=0.01$, $\tau=0.8$, $\mathrm{Re}=5$, $\Delta=0.5$, $r=0.1$, $\beta=0$).

$r=0.1$ and $\beta=0$ in the dynamic method. Fig. 15 shows results being in a very good agreement with Guo's. In fact, it is an elaborate task to apply the present method to the corner of lid-driven flow. Special consideration and care need to be given, because different choice of the extrapolation scheme to obtain $\mathbf{u}_w^o$ leads to a different effect, and even influences the solution stability.

4.5 Convergence study and further discussions

We have demonstrated that by using the dynamic boundary model, one can get more accurate and stable results. But the convergence is another property to be checked. We found that the relaxation time τ is the main factor affecting the convergence property of the dynamic boundary model. We studied the convergence property for the 2-D Poiseuille flow at $U_0=0.01$ and $\Delta=0.5$. For the dynamic method,

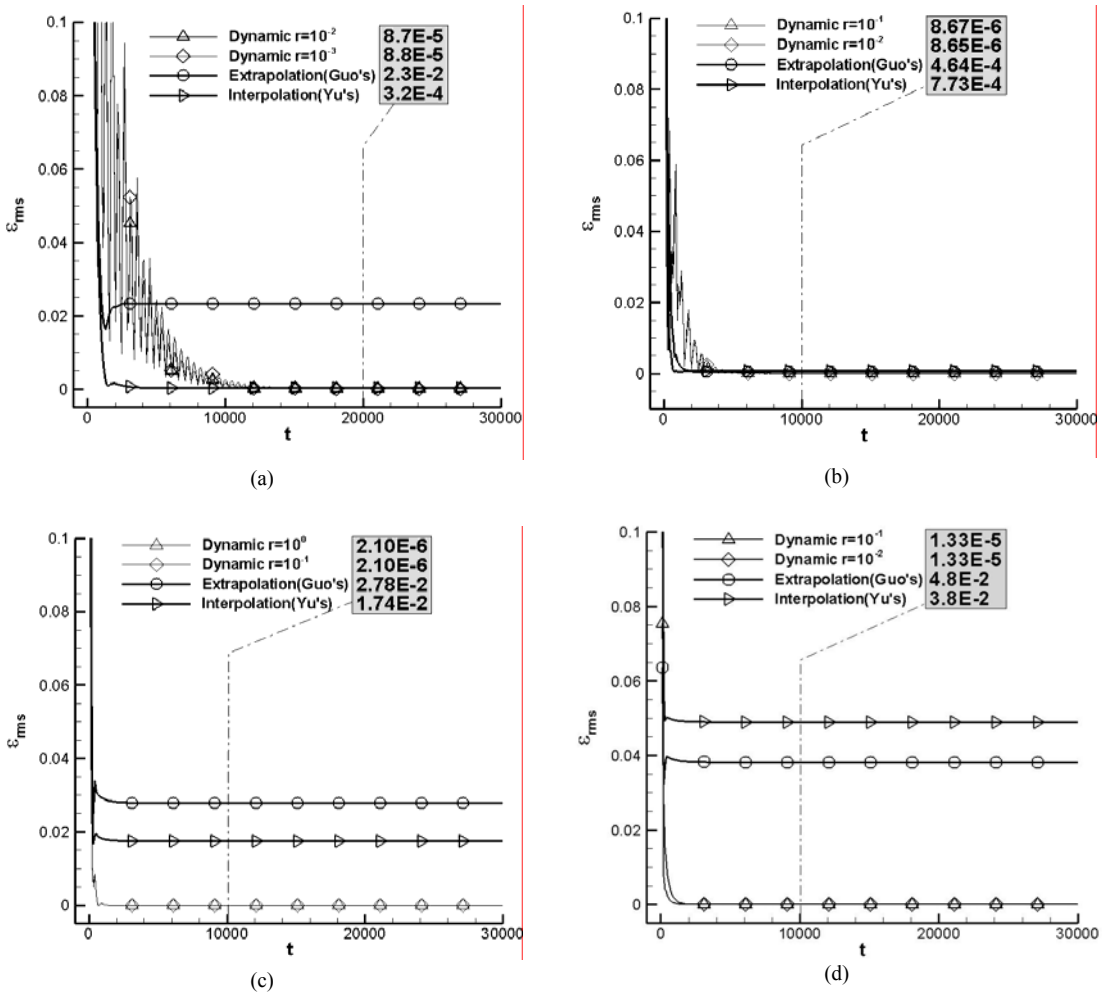

Fig. 16. Convergence properties of different methods used for implementing the no-slip boundary condition in LBM for the Poiseuille flow at $U_0 = 0.01$, $\Delta = 0.5$, and $\beta = 0$; (a) $\tau = 0.6$, (b) $\tau = 1$, (c) $\tau = 3$, (d) $\tau = 5$.

we checked the performance at the β value fixed at $\beta = 0$ but with the r value in the range $0.001 \le r \le 1$. Fig. 16 displays the numerical results at four different τ values. The comparison shows that at small value of τ, applying the dynamic boundary model can cause rather slow convergence and can reveal fluctuation at the initial transient state. However, increase of τ brings faster convergence as well as higher accuracy with the dynamic boundary model.

We have shown that applying the dynamic boundary model to the implementation of no-slip boundary conditions in LBM is suitable and robust especially when τ is large. The Reynolds number, $\mathrm{Re} = U_0 L / \upsilon$, can be written as $\mathrm{Re} = 3U_0 L/(\tau - 0.5)$ from the relation $\tau = 3\upsilon + 0.5$. So, to make the Reynolds number small, we must set either U_0 small or τ large. On the other hand in studying the fluid mixing, the value of U_0 must be set as large as possible because the mixing phenomenon is usually governed by particle advection, and small U_0 would lead to longer CPU time. Therefore, in order to apply LBM to the fluid mixing especially in microfluidics, which is characterized by low Reynolds numbers, we must find a suitable scheme such as the present method that is robust even at high τ values. Even so, we still need to investigate the effect of r on the convergence property more elaborately and hopefully find some way to improve the convergence property.

5. Conclusions

In this study, we have presented a new LBM boundary-condition-treatment method, namely, dynamic boundary model. We applied the present method together with other existing typical methods to several flow problems. The comparison study demonstrated that the proposed dynamic boundary model possesses the ability to produce high accuracy. We also derived a conclusion that our dynamic boundary model is particularly suitable in simulating microfluidic mixing problem as it is more robust and accurate at low Reynolds numbers than other methods.

Acknowledgment

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the National Research Laboratory Program funded by the Ministry of Science and Technology (No. 2005-1091).

References

[1] D. P. Ziegler, Boundary Conditions for Lattice Boltzmann Simulations, *J. Stat. Phys.,* 711 (1993) 1171-1177.

[2] A. J. C. Ladd, Numerical Simulation of Particular Suspensions via a Discretized Boltzmann Method, *J. Flui Mech.,* 271 (1994) 311-339.

[3] X. He, Q. Zou, L. S. Luo and M. Dembo, Analytic Solutions and Analysis on Non-slip Boundary Condition for the Lattice Boltzmann BGK Model, *J. Stat. Phys.,* 87 (1997) 115-136.

[4] S. Succi, The Lattice Boltzmann Equation for Fluid Dynamics and Beyond, Oxford, Clarendon Press, (2001) 84-87.

[5] D. Yu, R. Mei and W. Shyy, A Unified Boundary Treatment in Lattice Boltzmann Method, *1st Aerospace Sciences Meeting and Exhibit, AIAA,* (2003) 2003-2953.

[6] Z. Guo, C. Zheng and B. Shi, An Extrapolation Method for Boundary Conditions in Lattice Boltzmann Method, *Phys. Fluids,* 14 (6) (2002) 2007-2010.

[7] I. Ginzbourg and P. M. Alder, Boundary Flow Condition Analysis for the Three-dimensional Lattice Boltzmann Model, *J. Phys. II France,* (1994) 191-214.

[8] Y. H. Qian, D. D'Humieres and P. Lallemand, Lattice BGK Models for Navier-Stokes Equation, *Europhys. Lett.,* 17 (1992) 479-484.

[9] X. He and L. Luo, Lattice Boltzmann Model for the Incompressible Navier-stokes Equation, *J. Stat. Phys.,* 88 (1997) 927-944.