A mass conserving boundary condition for the lattice Boltzmann equation method

Jie Bao *, Peng Yuan, Laura Schaefer

Mechanical Engineering Department, University of Pittsburgh, Pittsburgh, PA 15261, United States

ARTICLE INFO

ABSTRACT

In lattice Boltzmann (LB) simulations, the widely used wall boundary conditions (BCs) proposed by Filipova and Hänel (FH) and Mei, Luo and Shyy (MLS) result in constant mass leakage in certain circumstances. In this paper, we have analyzed the source of the leakage. Based on this analysis, we propose a second-order accurate mass conserving wall BC. In our BC, the distribution function at a wall node is decomposed into its equilibrium and non-equilibrium parts. The mass conservation is guaranteed by enforcing a mass conserving rule in the construction of the fictitious equilibrium distribution part. We have shown through several benchmark test problems involving steady and unsteady flows that our new BC not only eliminates the constant mass leakage, but also has many other advantages over the FH and MLS BCs.

1. Introduction

In LBE simulations [1–4], to some extent, developing accurate and efficient BCs is as important as developing an accurate computation scheme itself, since they will influence the stability of the computation. The most common and simplest solid wall BC is the bounce-back boundary condition. In this BC, when a particle distribution streams to a wall node, it scatters back to the fluid node along its incoming link. However, the bounce-back BC only gives first-order numerical accuracy. To improve it, many BCs have been proposed in the past [5–8], such as the halfway bounce-back scheme [9,10], extrapolation scheme [11] and non-equilibrium bounce-back scheme [12]. However, most of these BCs are only suitable for flat walls. When applied to curved walls, these BCs will result in jagged boundaries, and additional errors will therefore be introduced. Recently, Filipova and Hänel (FH) proposed a curved wall BC [13], which later was improved by Mei, Luo and Shyy (MLS) [14,15]. Both the FH and MLS BCs are second-order accurate and have the ability to model curved geometries, and therefore are widely used in the literature. However, these models do not consider body forces, like gravity or a magnetic field, and we have found that when a body force is applied on the fluids, the requirement of mass conservation cannot be exactly met. Hence, based on the FH and MLS BC, we developed a mass conserving solid wall BC for the cases with body force. While, Chen's volumetric (as opposed to pointwise) lattice Boltzmann theory [16–18] can ensure mass conservation in the whole simulation domain with a body force, the true strengths of LBE methods lie in their ability to simulate multi-phase fluids for both single and multi-component fluids. The most widely used multi-phase, multi-component or thermal LBE models are based on the pointwise LBE theory [19–24], so it is still necessary to develop a mass conserving solid curved wall BC for the pointwise LBE model. Doing so extends the applicability of these multi-component, multi-phase and thermal models to many more practical problems.

---

* Corresponding author.
  E-mail address: jib5@pitt.edu (J. Bao).

2. Implementation of a mass conserving solid wall BC

2.1. Formulation of the FH and MLS BCs

Our mass conserving solid wall BC is derived from the FH and MLS BC, so we will briefly discuss both of them. As shown in Fig. 1, $\mathbf{e}_\alpha$ and $\mathbf{e}_{\bar{\alpha}}$ denote directions opposite to each other, $\mathbf{x}_b$ is a boundary node, and $\mathbf{x}_f$ is a fluid node. The curved wall is located between a boundary node and fluid node, with $\Delta = \frac{|\mathbf{x}_f - \mathbf{x}_w|}{|\mathbf{x}_f - \mathbf{x}_b|}$ denoting the fraction of an intersected link in the fluid region. Obviously, $0 \leqslant \Delta \leqslant 1$. In order to finish the streaming step, we need to know $\tilde{f}_{\bar{\alpha}}(\mathbf{x}_b, t)$ at boundary node $\mathbf{x}_b$, where $\tilde{f}_{\bar{\alpha}}$ denotes the post-collision state of the distribution function.

FH proposed the following treatment for $\tilde{f}_{\bar{\alpha}}(\mathbf{x}_b, t)$ on curved boundaries:

$$\tilde{f}_{\bar{\alpha}}(\mathbf{x}_b, t) = (1 - \chi)\tilde{f}_{\alpha}(\mathbf{x}_f, t) + \chi f_{\alpha}^{(*)}(\mathbf{x}_b, t) + 2w_\alpha \rho \frac{3}{c^2} \mathbf{e}_{\bar{\alpha}} \cdot \mathbf{u}_w \tag{1}$$

where $\mathbf{u}_w \equiv \mathbf{u}(\mathbf{x}_w, t)$ is the velocity at the wall, χ is the weighting factor that controls the linear interpolation between $\tilde{f}_\alpha(\mathbf{x}_f, t)$ and $f_\alpha^{(*)}(\mathbf{x}_b, t)$, and $f_\alpha^{(*)}(\mathbf{x}_b, t)$ is given by a fictitious equilibrium distribution:

$$f_\alpha^{(*)}(\mathbf{x}_b, t) = w_\alpha \rho(\mathbf{x}_f, t) \left[1 + \frac{3}{c^2} \mathbf{e}_\alpha \cdot \mathbf{u}_{bf} + \frac{9}{2c^4} (\mathbf{e}_\alpha \cdot \mathbf{u}_f)^2 - \frac{3}{2c^2} \mathbf{u}_f \cdot \mathbf{u}_f \right] \tag{2}$$

In Eq. (2), $\mathbf{u}_f \equiv \mathbf{u}(\mathbf{x}_f, t)$ is the fluid velocity near the wall, $\mathbf{u}_{bf}$ is to be chosen, and the weighting factor χ depends on $\mathbf{u}_{bf}$. In FH's original paper, the following combination of $\mathbf{u}_{bf}$ and χ were given:

$$\mathbf{u}_{bf} = (\Delta - 1)\mathbf{u}_f/\Delta + \mathbf{u}_w/\Delta \quad \text{and} \quad \chi = (2\Delta - 1)/\tau \quad \text{for } \Delta \geqslant 1/2 \tag{3a}$$

$$\mathbf{u}_{bf} = \mathbf{u}_f \quad \text{and} \quad \chi = (2\Delta - 1)/(\tau - 1) \quad \text{for } \Delta < 1/2 \tag{3b}$$

In order to improve the stability of the scheme, MLS revised the expressions for $\mathbf{u}_{bf}$ and χ to the following form:

$$\mathbf{u}_{bf} = [1 - 3/(2\Delta)]\mathbf{u}_f + 3/(2\Delta)\mathbf{u}_w \quad \text{and} \quad \chi = (2\Delta - 1)/(\tau + 1/2) \quad \text{for } \Delta \geqslant 1/2 \tag{4a}$$

$$\mathbf{u}_{bf} = \mathbf{u}_{ff} = \mathbf{u}_f(\mathbf{x}_f + \mathbf{e}_{\bar{\alpha}} \delta t, t) \quad \text{and} \quad \chi = (2\Delta - 1)/(\tau - 2) \quad \text{for } \Delta < 1/2 \tag{4b}$$

These boundary treatments can handle curved boundaries as well as moving boundaries.

2.2. Mass leakage with the FH and MLS BCs

Usually in LBE simulations, the mass of the system is not conserved exactly. There will be mass loss/gain at each time step during the beginning stage of the simulation, which is the so-called "mass leakage" in the literature. Since in many cases the leakage is very small and eventually approaches zero with enough time steps, its effect on the simulation results is neglected. However, under some circumstances, this mass leakage grows and will affect the simulation result.

One simple yet very useful case to consider is static flow under gravity, $\mathbf{G} = -\rho g \mathbf{j}$. Fig. 2 shows such a system. We set $\Delta = 1$, $u_w = 0$, $\tau = 1$, $g = 0.001$ and set the lattice resolution to be $nx \times ny = 100 \times 50$. The pressure drop $\frac{dp}{dx}$ can be varied from the order of 1×10^{-6} to the order of 10^{-3}, and the Reynolds number can be adjusted in the range from 1.0 to about 2000. Using
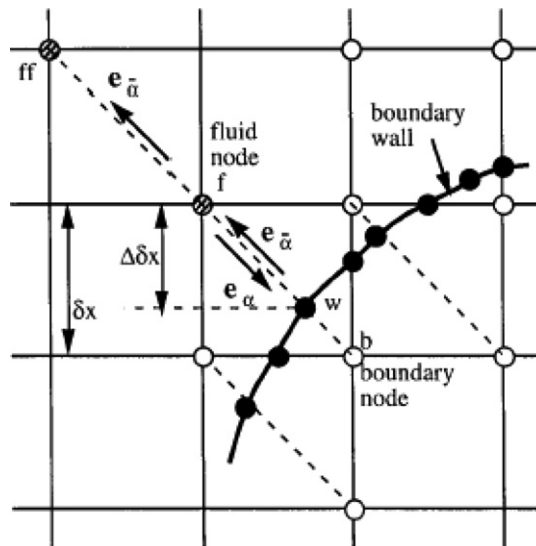

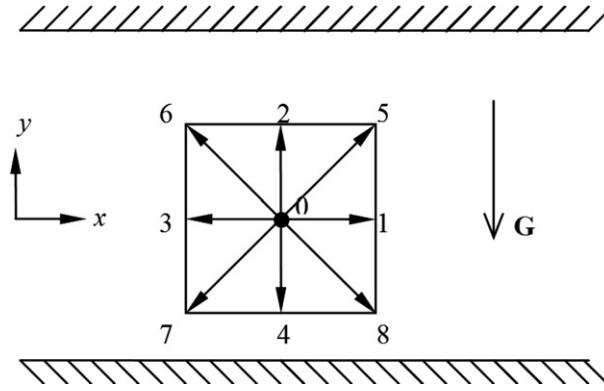**Fig. 1.** Layout of the lattice and curved wall boundary (after [14]).

Fig. 2. Schematic of static flow under gravity.

the FH BC, the system mass will keep decreasing at a constant rate, as shown in Fig. 3. Shown in Fig. 4 is the mass changing rate (defined as $mass|_{t+100} - mass|_t$) varying with time, which fluctuates at the beginning stage, and then instead of approaching to zero as in a typical LBE simulation, approaches a constant negative value.

By setting $\Delta = 1$ and $\tau = 1$, the weighting coefficient is $\chi = 1$. Therefore, if $\mathbf{u}_w = 0$, Eq. (1) becomes $\tilde{f}_{\tilde{\alpha}}(\mathbf{x}_b, t) = f_{\alpha}^{(*)}(\mathbf{x}_b, t)$. Considering another extreme case, that of $\Delta = 1/2$, then $\chi = 0$, which means that $\tilde{f}_{\tilde{\alpha}}(\mathbf{x}_b, t) = \tilde{f}_{\alpha}(\mathbf{x}_f, t)$. In that case, the mass is exactly conserved in the simulations. From this comparison, we can see the mass leakage is due to the $f_{\alpha}^{(*)}(\mathbf{x}_b, t)$ term in Eq. (1).

Since the mass leakage results from the interchange of mass between fluid nodes and boundary nodes, we now consider two fluid nodes next to the boundary, located at $y = 1$ and $y = ny - 1$, respectively, and only study their outgoing and incoming PDFs. In order to simplify our analysis, the direct body forcing approach, which is simple and commonly used, was applied to incorporate the body force [25]. In this approach, the body force is included in the scheme after the collision step by

$$\tilde{f}_{\alpha}(\mathbf{x}_i, t) \quad \Leftarrow \quad \tilde{f}_{\alpha}(\mathbf{x}_i, t) - w_{\alpha}\frac{3}{c^2}\rho(\mathbf{x}_i, t)g\mathbf{e}_{\alpha}\cdot\hat{\mathbf{j}} \tag{5}$$

For the upper fluid node at $y = ny - 1$, the outgoing PDFs are f_2, f_5 and f_6 which are given by

$$\tilde{f}_2(ny - 1, t) \approx \tilde{f}_2^{(eq)}(ny - 1, t) - \frac{\rho(ny - 1, t)g}{3c^2} \tag{6a}$$

$$\tilde{f}_5(ny - 1, t) \approx \tilde{f}_5^{(eq)}(ny - 1, t) - \frac{\rho(ny - 1, t)g}{12c^2} \tag{6b}$$

$$\tilde{f}_6(ny - 1, t) \approx \tilde{f}_6^{(eq)}(ny - 1, t) - \frac{\rho(ny - 1, t)g}{12c^2} \tag{6c}$$

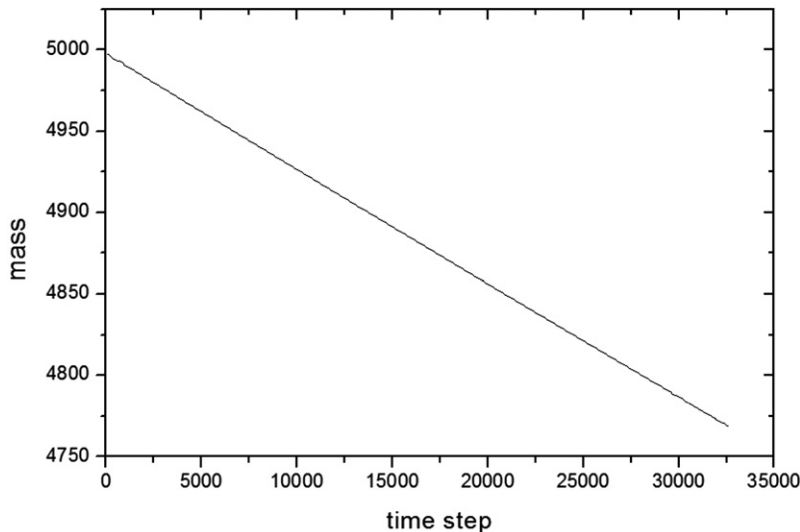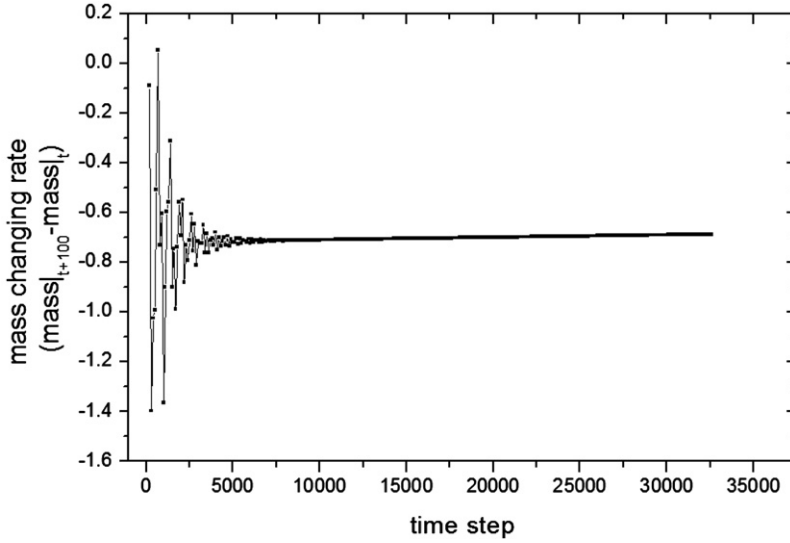Fig. 3. System mass changes with time using FH BC.

Fig. 4. Mass changing rate varying with time.

In the above equations, use the "=" sign when $\tau = 1$. Similarly, for the lower fluid node at $y = 1$, the outgoing PDFs are f_4, f_7 and f_8 which are given by

$$\tilde{f}_4(1,t) \approx \tilde{f}_4^{(eq)}(1,t) + \frac{\rho(1,t)g}{3c^2} \tag{6d}$$

$$\tilde{f}_7(1,t) \approx \tilde{f}_7^{(eq)}(1,t) + \frac{\rho(1,t)g}{12c^2} \tag{6e}$$

$$\tilde{f}_8(1,t) \approx \tilde{f}_8^{(eq)}(1,t) + \frac{\rho(1,t)g}{12c^2} \tag{6f}$$

For $\tau = 1$, the incoming PDFs are

$$\tilde{f}_4(ny - 1, t) = f_2^{(*)}(ny, t) \approx \tilde{f}_2^{(eq)}(ny - 1, t) \tag{7a}$$

$$\tilde{f}_7(ny - 1, t) = f_5^{(*)}(ny, t) \approx \tilde{f}_5^{(eq)}(ny - 1, t) \tag{7b}$$

$$\tilde{f}_8(ny - 1, t) = f_6^{(*)}(ny, t) \approx \tilde{f}_6^{(eq)}(ny - 1, t) \tag{7c}$$

$$\tilde{f}_2(1, t) = f_4^{(*)}(0, t) \approx \tilde{f}_4^{(eq)}(1, t) \tag{7d}$$

$$\tilde{f}_5(1, t) = f_7^{(*)}(0, t) \approx \tilde{f}_7^{(eq)}(1, t) \tag{7e}$$

$$\tilde{f}_6(1, t) = f_8^{(*)}(0, t) \approx \tilde{f}_8^{(eq)}(1, t) \tag{7f}$$

The "$\approx$" is obtained because $f_\alpha^{(*)}(\mathbf{x}_b, t)$ is given by a fictitious equilibrium distribution, Eq. (2), which is in the same form as $f_\alpha^{(eq)}$ in

$$f_\alpha^{(eq)} = \rho w_\alpha \left[1 + \frac{3}{c^2} \mathbf{e}_\alpha \cdot \mathbf{u} + \frac{9}{2c^4} (\mathbf{e}_\alpha \cdot \mathbf{u})^2 - \frac{3}{2c^2} \mathbf{u} \cdot \mathbf{u} \right] \tag{8}$$

The only difference is that in the second term, $\mathbf{u}_{bf}$ is used instead of $\mathbf{u}_f$. Since $\mathbf{u}_{bf} \approx \mathbf{u}_f$ (for the FH BC, when $\Delta < 1/2$, $\mathbf{u}_{bf} = \mathbf{u}_f$), therefore $f_\alpha^{(*)}(\mathbf{x}_b, t) \approx \tilde{f}_\alpha^{(eq)}(\mathbf{x}_f, t)$.

Subsequently, the mass leakage of one pair of fluid nodes, i.e., the upper and the lower fluid node will be

$$\sum_{\text{outgoing}} \tilde{f} - \sum_{\text{incoming}} \tilde{f} = \tilde{f}_2(ny - 1, t) + \tilde{f}_5(ny - 1, t) + \tilde{f}_6(ny - 1, t) + \tilde{f}_4(1, t) + \tilde{f}_7(1, t) + \tilde{f}_8(1, t) - [\tilde{f}_4(ny - 1, t)$$

$$+ \tilde{f}_7(ny - 1, t) + \tilde{f}_8(ny - 1, t) + \tilde{f}_2(1, t) - \tilde{f}_5(1, t) - \tilde{f}_6(1, t)]$$

$$\approx \frac{1}{2c^2} [\rho(1, t) - \rho(ny - 1, t)]g = \frac{1}{2} [\rho(1, t) - \rho(ny - 1, t)]g \tag{9}$$

Since lattice Boltzmann fluids are intrinsically compressible, introducing gravity into the LB scheme will result in density variation in the system. The density difference between the upper and lower fluid nodes therefore produces the mass leakage. Supposing that there is no density variation along the *x*-direction, which is verified by the simulation, and also assuming

that the density change is small during a short time interval, say 100 time steps, the mass leakage of the system in this time interval can be calculated by

$$\text{mass leakage} = 0.5 \times (nx + 1) \times \text{ time interval } \times [\rho(ny - 1) - \rho(1)]g \tag{10}$$

Fig. 5 shows the density at $x = nx/2$ as a function of height when $t = 32{,}600$. Using Eqs. (9) and (10) we can then estimate the mass leakage during 100 time steps:

$$\text{mass leakage} = 0.5 \times 101 \times 100 \times (0.887033 - 1.024421) \times 0.001 = -0.6938$$

The actual mass leakage during this time interval is -0.6867 (see Fig. 4), which is very close to our estimate, an error of 0.03%.

The above discussion clearly shows that the mass leakage is due to the density variation in the system, or more precisely, due to the fact that the $f_\alpha^{(*)}(\mathbf{x}_b, t)$ term does not correctly reflect the density variation. Since $\tilde{f}_{\bar{\alpha}}(\mathbf{x}_b, t)$ is the linear combination of $\tilde{f}_\alpha(\mathbf{x}_f, t)$ and $f_\alpha^{(*)}(\mathbf{x}_b, t)$, to improve the stability of the scheme, MLS tried to change the weighting coefficients of these two terms. In the commonly used range of τ, MLS's revision produces a smaller weighting coefficient χ and therefore a better stability. However, the MLS BC cannot eliminate the constant mass leakage problem that is exhibited when using the FH BC in certain circumstances. In most cases, it can only decrease the mass changing rate, unless $\chi = 0$.

It should be noted here that for illustration purposes, the intensity of gravity $g = 0.001$ used in the example is quite large, but the ability to account for a large magnitude of gravity is very useful in many practical problems, as we will discuss this in the following section. Usually, in the LBE simulations, the intensity of the body force is restricted to satisfy the incompressible limit, $\mathbf{u} \ll c_s$. In this example, the common practice is to set $g * ny \ll c_s^2$, which results in a smaller density variation, and, it should be noted, a smaller, but still present, mass leakage.

2.3. A mass conserving solid wall BC

From the above discussions, we can see that a boundary condition that can preserve the total mass in a given system is very important for LBE simulations. Based on the FH and MLS BCs, we have defined a new boundary condition, called the mass conserving solid wall boundary condition. We still use linear interpolation between $\tilde{f}_\alpha(\mathbf{x}_f, t)$ and $f_\alpha^{(*)}(\mathbf{x}_b, t)$ (Eq. (1)) to find $\tilde{f}_{\bar{\alpha}}(\mathbf{x}_b, t)$. However, we change the density term in the expression of $f_\alpha^{(*)}(\mathbf{x}_b, t)$, (Eq. (2)) to guarantee mass conservation, since the $f_\alpha^{(*)}(\mathbf{x}_b, t)$ term is responsible for the mass leakage. Therefore, we define that

$$f_\alpha^{(*)}(\mathbf{x}_b, t) = w_\alpha \rho(\mathbf{x}_w, t) \left[1 + \frac{3}{c^2} \mathbf{e}_\alpha \cdot \mathbf{u}_{bf} + \frac{9}{2c^4} (\mathbf{e}_\alpha \cdot \mathbf{u}_f)^2 - \frac{3}{2c^2} \mathbf{u}_f \cdot \mathbf{u}_f \right] \tag{11}$$

where $\rho(\mathbf{x}_w, t)$ is called the wall density.

It must be determined what $\rho(\mathbf{x}_w, t)$ term guarantees mass conservation. We discuss this in the context of D2Q9 model. Shown in Fig. 6 are the known and unknown PDFs of a flat boundary site at the lower wall boundary after the streaming step.
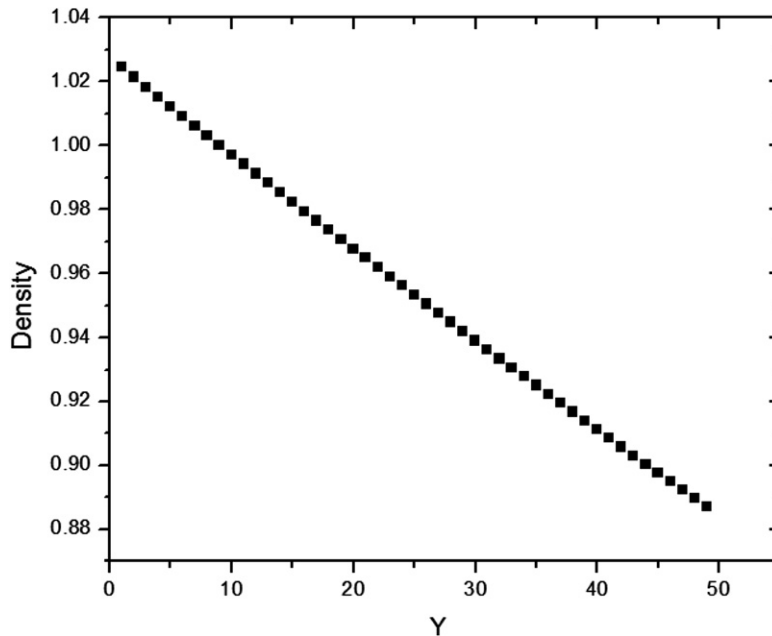

Fig. 5. Density as a function of height at a selected time.

Unknown (Incoming) PDFs: $f_2 = ?$ $f_5 = ?, f_6 = ?$

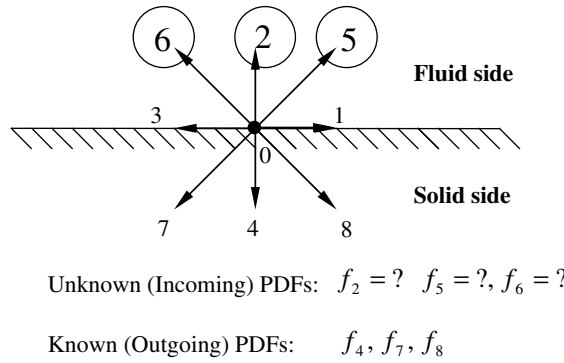Known (Outgoing) PDFs: f_4, f_7, f_8

Fig. 6. PDFs of a flat boundary site at the lower wall boundary after the streaming step.

The outgoing PDFs are f_4, f_7, f_8, which are known and the incoming PDFs are f_2, f_5, f_6, which are unknown. Mass conservation requires that $f_2 + f_5 + f_6 = f_4 + f_7 + f_8$. Assume that all the outgoing PDFs also satisfy Eq. (11), with an unknown $\rho(\mathbf{x}_w, t)$ term. Then summing these together, we find

$$f_4 + f_7 + f_8 = \frac{1}{6}\rho(\mathbf{x}_w, t)[1 - 3u_{bf}^y + 3(u_f^y)^2] \tag{12}$$

where u_{bf}^y is the y-component of $\mathbf{u}_{bf}$ and u_f^y is the y-component of $\mathbf{u}_f$. Therefore, $\rho(\mathbf{x}_w, t)$ will be

$$\rho(\mathbf{x}_w, t) = 6\frac{f_4 + f_7 + f_8}{1 - 3u_{bf}^y + 3(u_f^y)^2} \tag{13}$$

Then by substituting the expression of $\rho(\mathbf{x}_w, t)$ into Eq. (11), the unknown PDFs f_2, f_5, f_6 can be obtained. It is straightforward to show that this boundary treatment indeed satisfies the condition $\sum_{outgoing} f = \sum_{incoming} f$, so the total mass is conserved.

In order to verify our new BC, simulations of the same problem specified in the previous section were conducted. We found that the linear mass leakage was eliminated and the mass of the system was indeed conserved. The mass distribution at $x = nx/2$ when $t = 60,000$ is plotted in Fig. 7 together with simulation results obtained by using the halfway bounce-back BC and FH BC. All three BCs give linear density distributions. The results of the new BC match well with that of the halfway bounce-back scheme. However, the results of the FH BC differ considerably from the other two because of mass leakage. The-
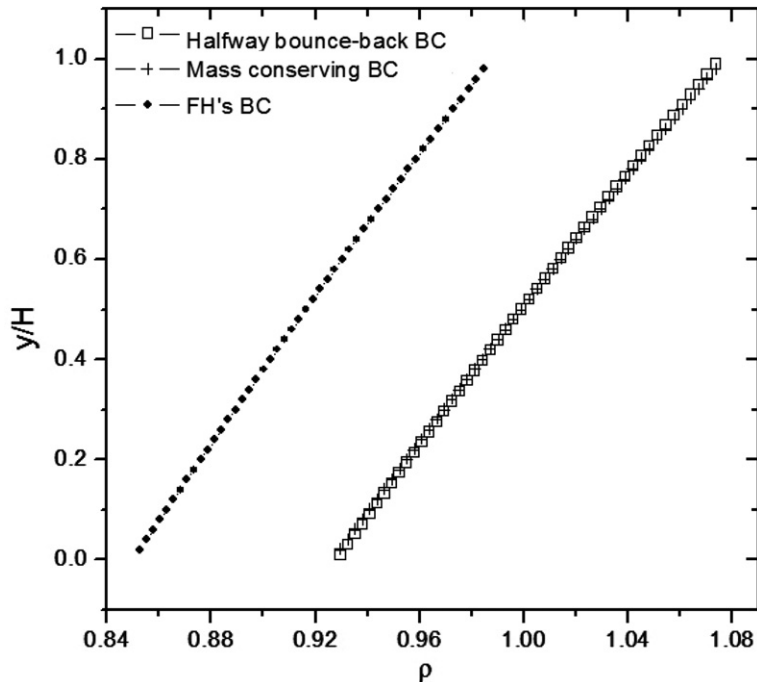

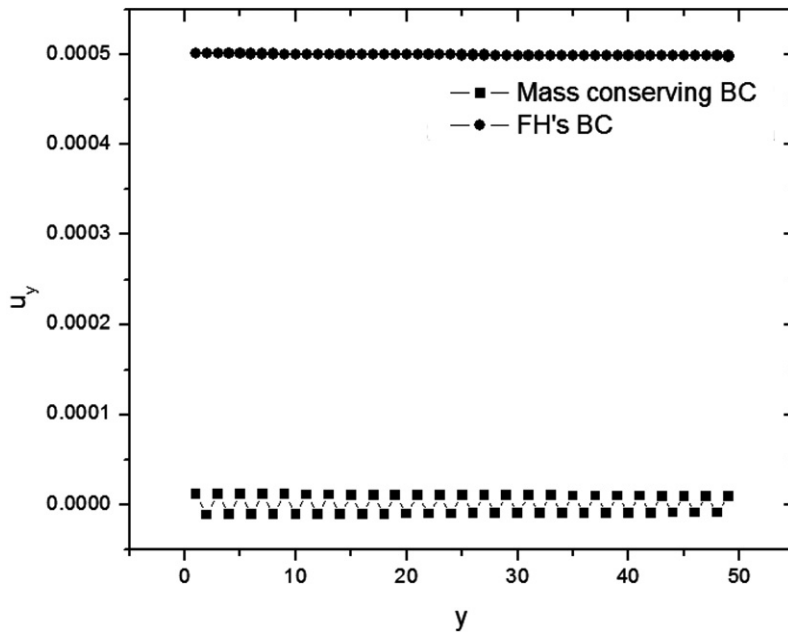**Fig. 7.** The mass distribution at $x = nx/2$ when $t = 60,000$.

Fig. 8. *y*-Component of velocity at $x = nx/2$ when $t = 60{,}000$.

oretically, all of the x- and y-components of velocity should be zero. However, due to the compressibility of the scheme, one will get non-zero y-velocities in the simulation. A comparison of y-velocities between our new BC and the FH BC is given in Fig. 8, where y-velocities at $x = nx/2$ when $t = 60{,}000$ are plotted. Even apart from its mass-conserving advantages, the y-velocities generated by our new scheme are much smaller (about 1/50) than those of the FH BC.

Using the same idea, i.e., setting $\sum_{\text{outgoing}} f = \sum_{\text{incoming}} f$, the new BC can be easily extended to other types of flat wall boundaries as well as curved wall boundaries. Fig. 9 shows three types of flat wall boundaries in a 2-D configuration. Both for the flat wall and convex corner boundaries, there is one pair of distributions that point into the flow (f_1 and f_3 for a flat wall; f_6 and f_8 for a convex corner), as illustrated by the black plain arrows. These we call fluid–fluid distributions. Since their values are known after streaming and they will not bring net mass change into the flow, we do not need to calculate their value nor need to include them in the procedure of finding the unknown distributions. In the concave corner case, there is one pair of distributions (f_6 and f_8), neither of which point to the flow. We call these buried distributions. Differing from fluid–fluid distributions, the buried distributions are not known after the streaming step. However, since they stream no mass into the flow, in the simulation we can simply switch their values as if they were bounce-back to the opposite direction or give the average value to each of them. Therefore, we do not need to include them in the procedure of finding the unknown distributions.

After defining these distributions, it is clear that these three cases are essentially the same. Therefore, for determining the incoming distributions, we can follow the same procedure as described in the flat wall case, i.e., use outgoing distributions to find the $\rho(\mathbf{x}_w, t)$ term, then substituting it into Eq. (11) to calculate the incoming distributions.
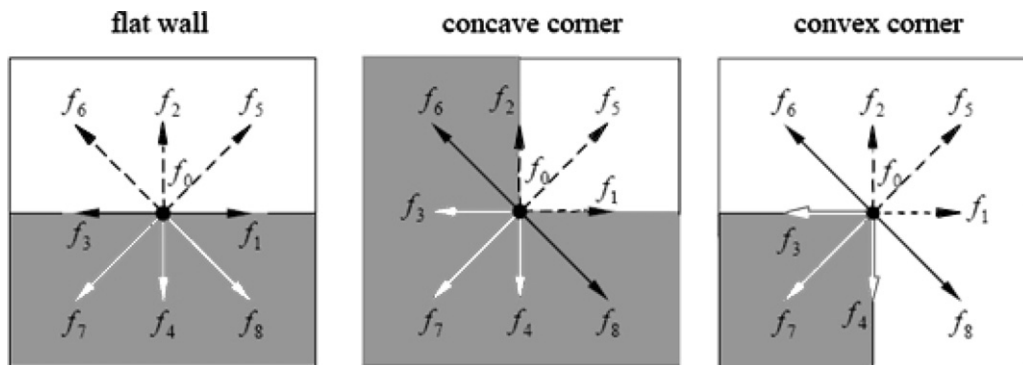

Fig. 9. Three flat wall boundary configurations. Black dashed arrows, white plain arrows and black plain arrows represent the incoming distribution, outgoing distribution and fluid–fluid (buried) distribution. Gray areas represent the solid part.

The new mass conserving boundary treatment can be easily extended to the 3-D case as well as for a curved wall boundary. In doing so, one needs to distinguish the incoming, outgoing, and fluid–fluid (buried) distributions as discussed above. Also for the curved wall boundary, the fraction of the intersected link Δ is not a constant over the entire wall, so one needs to decide its value for each boundary node. Meanwhile, the $f_\alpha^{(*)}(\mathbf{x}_b, t)$ term now correctly reflects the compressible effect, and the weighting coefficients $(1 - \chi$ and $\chi)$ are a function of τ. By using our boundary condition treatment, τ can work in a larger range, so that the range of χ values is also enlarged. Hence, the calculation of $\tilde{f}_\alpha(\mathbf{x}_b, t)$ through Eq. (1) becomes more flexible. The working range of τ and the stability of the model related to τ are discussed in the following benchmark section.

Even, in problems where a large gravity is not typically used as mentioned in Section 2.2, a relatively large gravity can increase the efficiency of the code in some cases. For example, the Rayleigh number, which is important in many thermal-fluid problems, is defined as $Ra = \frac{g\beta\Delta T (ny)^3}{v\alpha}$, where β is the thermal expansion coefficient, and ny is the lattice size in the y-direction. Hence, in such problems, when we want to simulate a large Rayleigh number, we can increase the gravity instead of using more grids in the y-direction, because increasing gravity does not significantly add to the computation load. By doing so, the efficiency of the code can be obviously increased when a relatively larger Rayleigh number is needed. Although increasing the grids in the y-direction seems more efficient, because it is a cubic, the increasing of grids lead to a large increase in time and money for many cases. For example, to increase the Rayleigh number 100 times, by only increasing the number of grids takes four times as long computationally. Additionally, because each lattice must be squire or cubic (for a 3-D problem), to maintain the ratio of simulation domain, when ny is increased, we often need to also increase nx and nz. As we know, Rayleigh number commonly range from 10^3 to 10^8 for practical engineering problems. Hence, adjusting the Rayleigh number only by changing ny would make the simulation very inefficient. Besides, the amount of available physical memory and operation system's addressing capability [26] strictly limits the number of grids. For 3-D problems, especially for multi-component, multi-phase or thermal systems, the memory usage often reaches to that limit. From Eq. (10), increasing the grids also increases the mass leakage. Finally, because the requirement of mass conservation can be exactly met, the stability of the model is greatly increased, and we will discuss this in detail in the following benchmark tests section.

3. Benchmark tests for the new BC

The new mass conserving BC is as simple to implement as the FH and MLS BCs yet demonstrates a better performance, as in the previously presented example. In this subsection, we will use several benchmark problems to test and illustrate this new BC further.

3.1. 2-D Poiseuille flow

Fro 2-D Poiseuille flow driven by a pressure gradient, the analytic solution is

$$u_{\text{exact}}(y) = -\frac{1}{2}\frac{\mathrm{d}p}{\mathrm{d}x}\frac{H^2}{\rho v}(\eta^2 - \eta) \tag{14}$$

where $\frac{\mathrm{d}p}{\mathrm{d}x}$ is the pressure gradient, which equals -1×10^{-6} in the following simulation tests, $H = ny - 2 + 2\Delta$ is the height of the channel and $\eta = (j - 1 + \Delta)/H$. In order to evaluate the computational error of the scheme, the following relative L_2-norm is defined:

$$E_2 = \frac{\left\{\int_0^H [u_{\text{LBE}}(y) - u_{\text{exact}}(y)]^2\,\mathrm{d}y\right\}^{1/2}}{\left[\int_0^H u_{\text{exact}}^2(y)\,\mathrm{d}y\right]^{1/2}} \tag{15}$$

where $u_{\text{LBE}}(y)$ is the LBE solution of the velocity.

First, we examine the accuracy of the new BC. Fig. 10 shows the dependence of the relative L_2-norm error on the lattice resolution for a number of different Δs. Also shown in Fig. 10 are the magnitudes of the slopes, k, obtained by linear fitting the error data for different Δs. By examining these k values, one can see that second-order accuracy was indeed maintained. It has been proven that the LBE method gives second-order accuracy for the interior points. Hence the overall second-order accuracy obtained here means that the accuracy of the boundary condition is at least of the second order.

Fig. 11 shows the relative L_2-norm error as a function of Δ for 2-D channel flow, which reaches its minimum value when $\Delta \approx 0.35$. For other Δ values, the error increases approximately linearly, which is similar to MLS's result [14,13].

Table 1 gives the L_2-norm error obtained by different boundary conditions for $\Delta = 0.75$, $\tau = 0.6$. For the new boundary condition, both FH's and MLS's approach of determining $\mathbf{u}_{\text{bf}}$ were used, denoted by "New + FH" and "New + MLS" in the table. From the table, it is clear that using the same $\mathbf{u}_{\text{bf}}$, the new BC results in the same error as the MLS BC and smaller errors than the FH BC at higher grid resolutions.

An important advantage of the new BC over the FH and MLS BCs independent of the magnitude of the gravity is its excellent stability. Fig. 12(a) and (b) shows the stable and unstable regions in the LBE computation for 2-D Poiseuille flow using the FH and MLS BCs. For $\Delta < 1/2$, there exists a certain region of τ, under which the computation is not stable. For the FH and MLS BCs, this region is around $\tau = 1$ and 2, respectively.
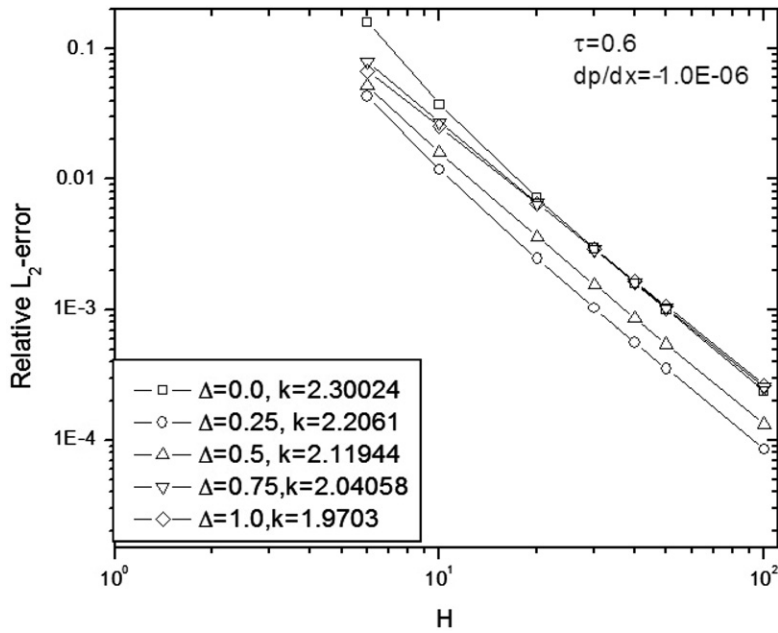
Fig. 10. Dependence of the relative L_2-norm error on the lattice resolution for 2-D channel flow.
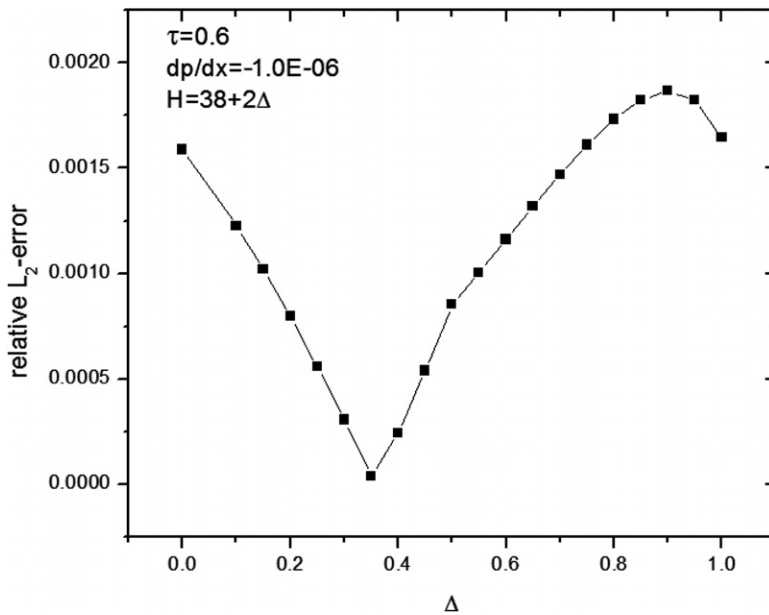

Fig. 11. Relative L_2-norm error as a function of Δ for 2-D channel flow.

Fig. 13 gives regions of stability and instability on the (Δ, τ) plane in the LBE computation for 2-D Poiseuille flow our new BC. Note that the scale for the y-axis is much finer than in Fig. 12(a) and (b). This means that in a simulation, one can use a τ value that is very close to 2.0. Meanwhile, since mass conservation is guaranteed, different weighting factors will not greatly affect the simulation result. If we need to use $\tau = 2.0$ in the simulation, we can simply switch to FH's weighting factor to avoid the singular point.

This instability is due to the mass leakage. For the FH BC, when τ is close to 1.0, then the absolute value of the weighting factor before the $f_\alpha^{(*)}(\mathbf{x}_b, t)$ term becomes very big, since $\chi = (2\Delta - 1)/(\tau - 1)$. We know that the mass leakage in the FH scheme is related to the $f_\alpha^{(*)}(\mathbf{x}_b, t)$ term, therefore a large $|\chi|$ will result in a large mass leakage for the system, and finally make the simulation unstable. As stated previously, MLS noticed this problem and changed the definition of the weighting

Table 1
The L_2-norm error obtained by different boundary conditions

BC	ny				
	10	20	30	40	50
FH	2.7256083E−02	6.5637481E−03	2.8808801E−03	1.6132460E−03	1.3193971E−03
New + FH	2.7256083E−02	6.5637481E−03	2.8808800E−03	1.6103690E−03	1.0267685E−03
MLS	3.0453999E−02	7.3338629E−03	3.2188895E−03	1.8021873E−03	1.4392486E−03
New + MLS	3.0453999E−02	7.3338629E−03	3.2188895E−03	1.8021873E−03	1.4392486E−03
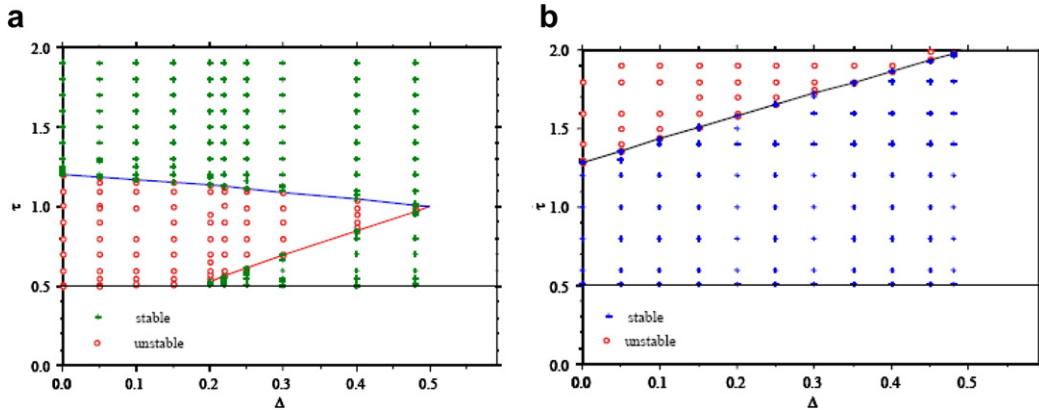
Fig. 12. Regions of stability and instability in the LBE computation for 2-D Poiseuille flow (after [3]): (a) using the FH BC and (b) using the MLS BC.
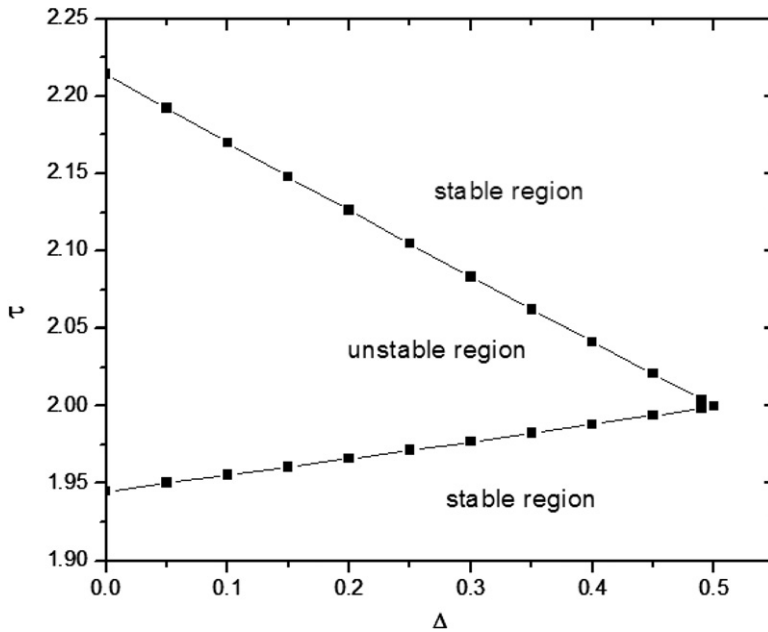

Fig. 13. Regions of stability and instability in the LBE computation for 2-D Poiseuille flow using the mass conserving BC.

factor to $\chi = (2\Delta - 1)/(\tau - 2)$. This will make the simulation stable for τ values around 1.0; however, for τ values around 2.0, the simulation becomes unstable due to the same reason. To verify this, we performed simulations for a point inside the unstable region ($\Delta = 0.2$, $\tau = 1/0.55 = 1.8181$) using the MLS BC. Fig. 14 gives the system mass as a function of time, which increases exponentially and finally causes the simulation to collapse.
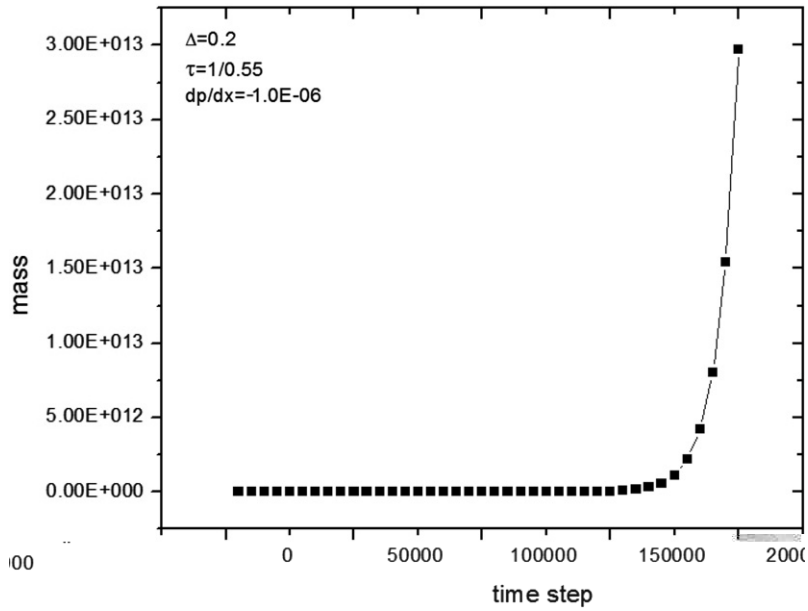
Fig. 14. System mass as a function of time using the MLS BC for a point inside the unstable region.

FH's treatment also encounters numerical instability for $\varDelta \geqslant 0.5$, as shown in Fig. 15. By modifying χ and $\mathbf{u}_{bf}$, MLS's treatment encounters no numerical instability for $\varDelta \geqslant 0.5$. In our new BC, due to mass conservation, there is also no numerical instability encountered for $\varDelta \geqslant 0.5$, no matter whether FH's or MLS's χ and $\mathbf{u}_{bf}$ are used.

We also compared τ_{min}, the minimum value of τ at which the computation is stable for a given $\varDelta$, for the different schemes. It is reported by Guo et al. [27] that for $\varDelta = 0.1$, $\tau_{min} = 0.509$ for the MLS treatment and 0.506 for his treatment. For $\varDelta = 0.2$, $\tau_{min} = 0.505$ for the MLS treatment and 0.50003 for his treatment. For $\varDelta \geqslant 0.3$, the simulations are still stable for both treatments even as $\tau - 0.5 = 10^{-7}$. For our boundary treatment, under the same conditions, we found that even for $\varDelta = 0.1$, the computation is still stable for $\tau - 0.5 = 2.5 \times 10^{-11}$. Therefore, our BC has a much better numerical stability than the other two methods. Furthermore, the Reynolds number is defined as

$$Re = \frac{\vec{u}\, n}{(\tau - 0.5)c^2} \tag{16}$$

Therefore, a small τ_{min} can significantly increase the Reynolds numbers without increasing computational load.
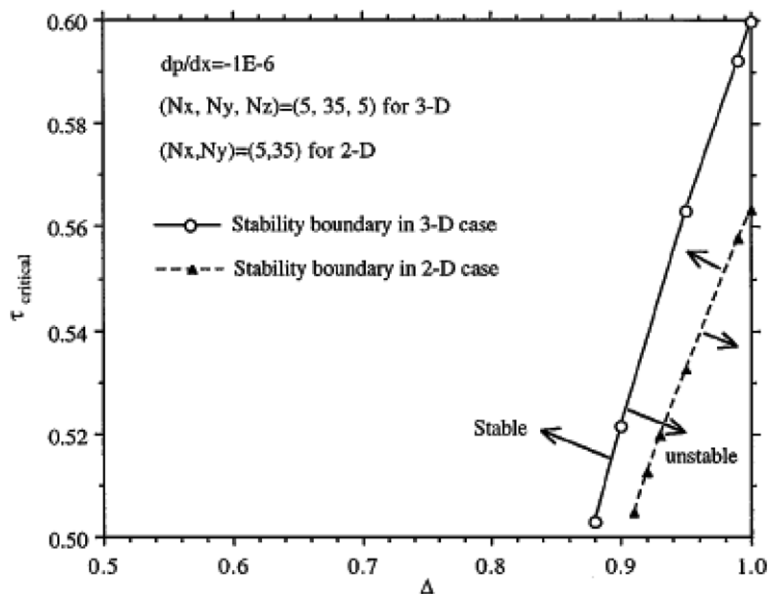

Fig. 15. Stability boundary of the FH scheme in a square duct flow for $\varDelta$ near 1.0 (after [15]).

3.2. Fully developed flow in a 2-D curved pipe

The FH and MLS's BC were designed for curved boundary, so we also test our new boundary condition for steady flow in a curved pipe, which is very common in practical thermal-fluid systems. As shown in Fig. 16, the fluid flows into the pipe through cross section A and flows out though cross section B. The gravity is applied on the fluid along the negative y-direction.

On the wall of the pipe, Eqs. (1), (4), (8), (11) and (13) are used to update the boundary conditions for the $\tilde{f}_\alpha(\mathbf{x}_b, t)$s. Fig. 17 shows the resulting x-velocity and y-velocity contours using the mass conserving BC, respectively. Fig. 18 shows the total system mass changes with time. Our new boundary condition can cause the total system mass to converge to a constant value, while the MLS BC demonstrates a steady mass leakage.

3.3. Steady and unsteady flow over a circular cylinder

For examining the accuracy of the mass conserving BC method on a curved wall, the final set of tests concerns 2-D steady and unsteady flows around a circular cylinder placed in a rectangular channel. For steady-state flow, the problem at $Re = 10$ is tested. For unsteady flow, the simulation is at $Re = 100$, resulting in a periodical vortex sheet.
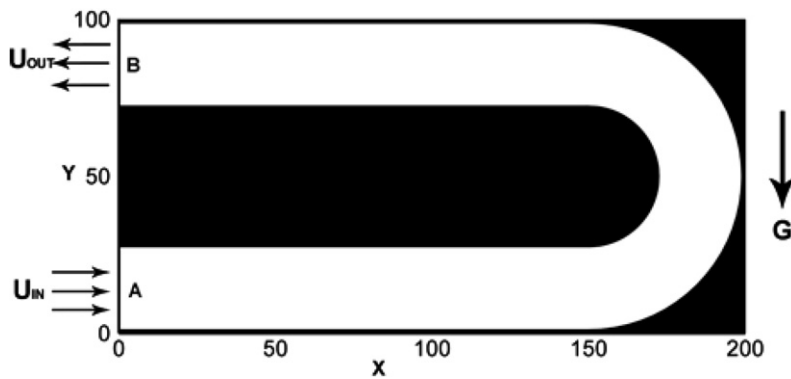

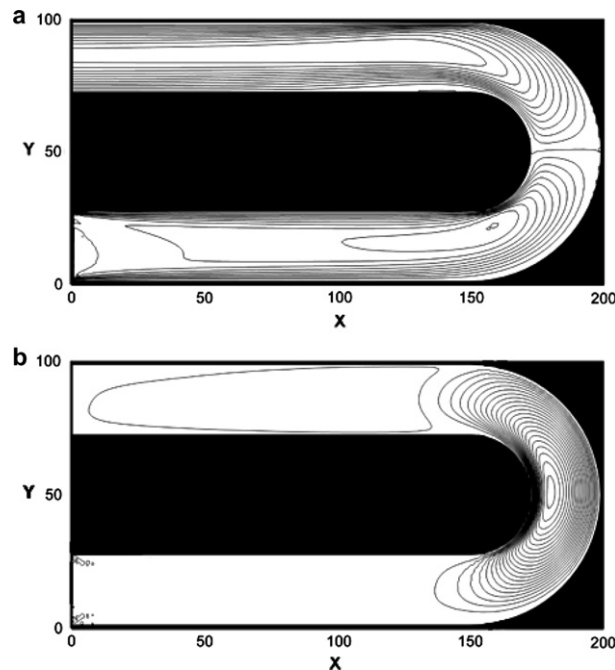Fig. 16. Schematic computational area for flow in a curved pipe.

Fig. 17. Flow in a curved pipe: (a) x-velocity contour and (b) y-velocity contour.
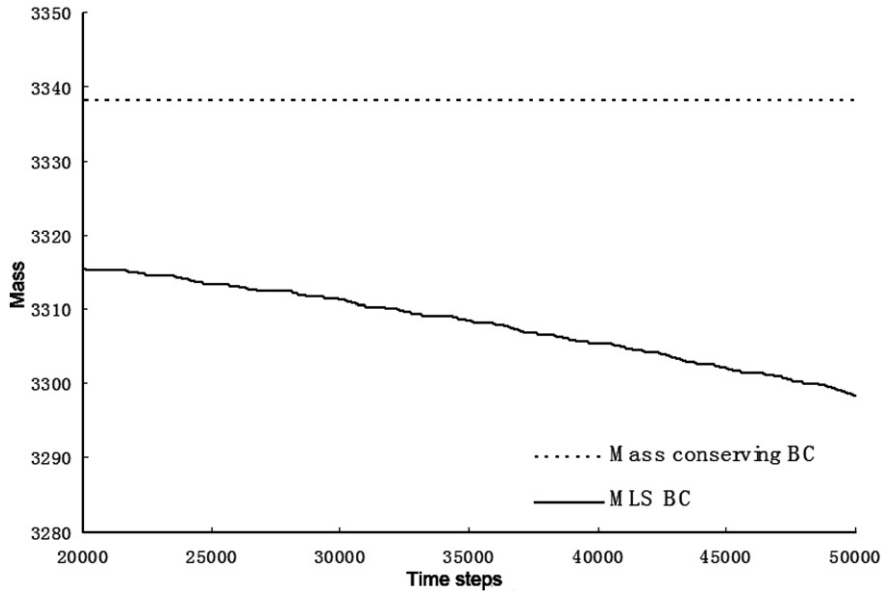
Fig. 18. System mass change with time using the mass conserving BC and MLS BC.

As shown in Fig. 19, for steady-state problems, the circular cylinder is placed in a domain with 105×70 nodes, and the center of the cylinder is at the origin $(0,0)$ of the grids. The periodical boundary condition treatment is used for the upper and lower boundaries. A uniform velocity is used at the inlet, and the extrapolation boundary condition is used at the outlet.

On the surface of the circular cylinder, mass conserving BC treatment is used to update the boundary conditions for the $\tilde{f}_\alpha(\mathbf{x}_b, t)$s. Fig. 20 shows the streamline and the contour of pressure at $Re = 2Vr/\upsilon = 10$ using $r = 3.5$. Fig. 21 shows the velocity profile $u(x = 0, y)/V$ for $H/r = 20$ at $Re = 10$. The MLS BC was also used in the same conditions and grids for comparison. The finite difference solution is obtained using body-fitted coordinates and are distributed along the upper surface of the circle over 200 grid points [14]. Fig. 22 shows the centerline $(y = 0)$ velocity variations, upstream and downstream, at $Re = 10$. The results using the MLS BC and finite different results are also presented for comparison.

As can be seen, the mass conserving solid wall BC results agree well with the result of the finite difference solution, and has a similar accuracy as the MLS BC. The difference in the results between the MLS and mass conserving BCs is around 0.01% of the MLS BC's results (this difference cannot clearly be seen from the figures). Hence, replacing $\rho(\mathbf{x}_f, t)$ with $\rho(\mathbf{x}_w, t)$ in Eq. (6) would not reduce the accuracy for problems with geometries that include curved geometry boundaries.

For unsteady flow, the problem is tested in a 300×100 node regime, with $r = 6$. The periodical boundary condition treatment is used for the upper and lower boundaries. Again, a uniform velocity is given at the inlet, and the extrapolation
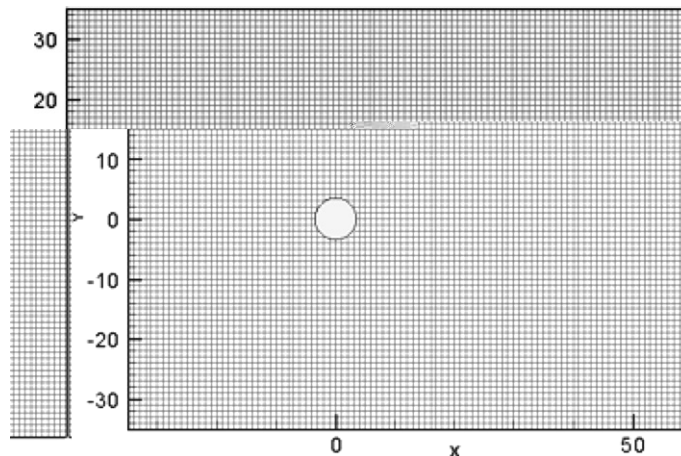

Fig. 19. Schematic computational area for steady flow over a circular cylinder. The cylinder has a diameter of 7 lattice units. The cylinder center-to-center distance (H) is 70 lattice units.
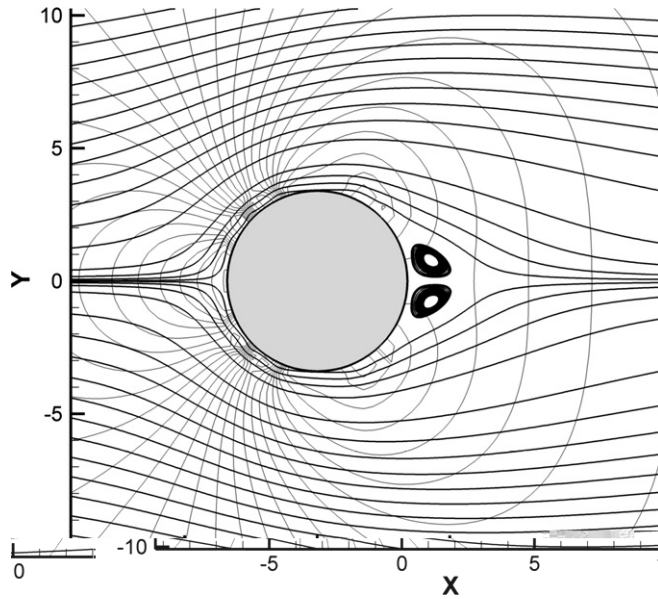
Fig. 20. Steady flow around a cylinder at Re = 10, with streamlines and pressure contours.
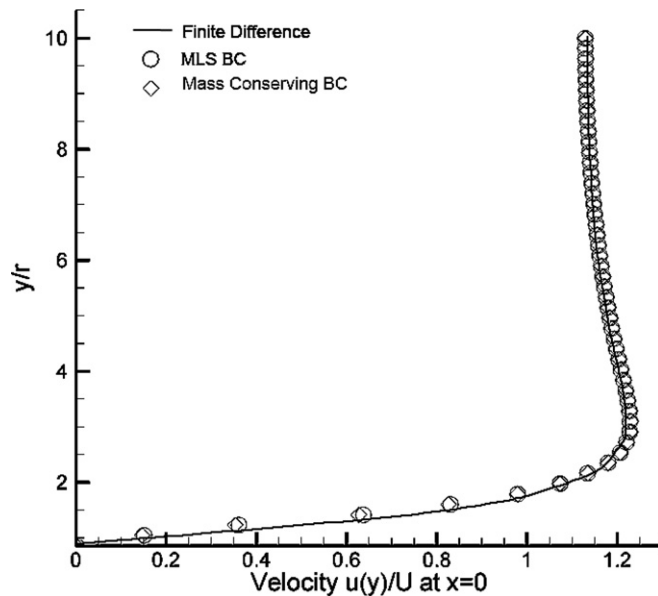

Fig. 21. Comparison of the velocity profiles at x = 0 for the MLS BC, mass conserving BC and finite difference result for Re = 10.

boundary condition is used at the outlet. The results of the developed periodical flow are shown in Fig. 23 for the x-velocity and y-velocity contours, respectively.

4. Discussion

In summary, we propose a second-order accurate mass conserving boundary condition for the LBE method. Several benchmark test problems were used to validate the accuracy and examine the robustness of the proposed BC. Compared with the FH and MLS BCs, our new BC has numerous advantages. First, the mass leakage is uniformly smaller than in the other schemes, and will not result in the constant mass leakage that occurs for the other BCs in some cases. These cases include larger magnitudes of gravity or magnetic force, and also include artificial gravity for certain problems as discussed in
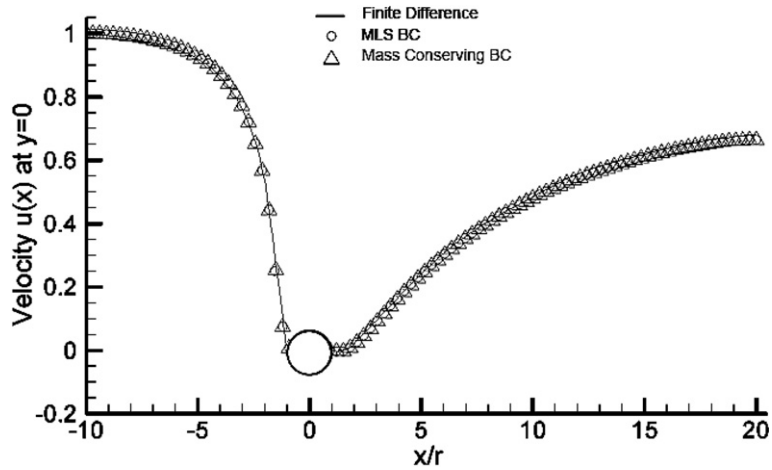
Fig. 22. Comparison of the velocity variation at the centerline ($y = 0$) for the MLS BC, mass conserving BC and finite difference result for $Re = 10$.
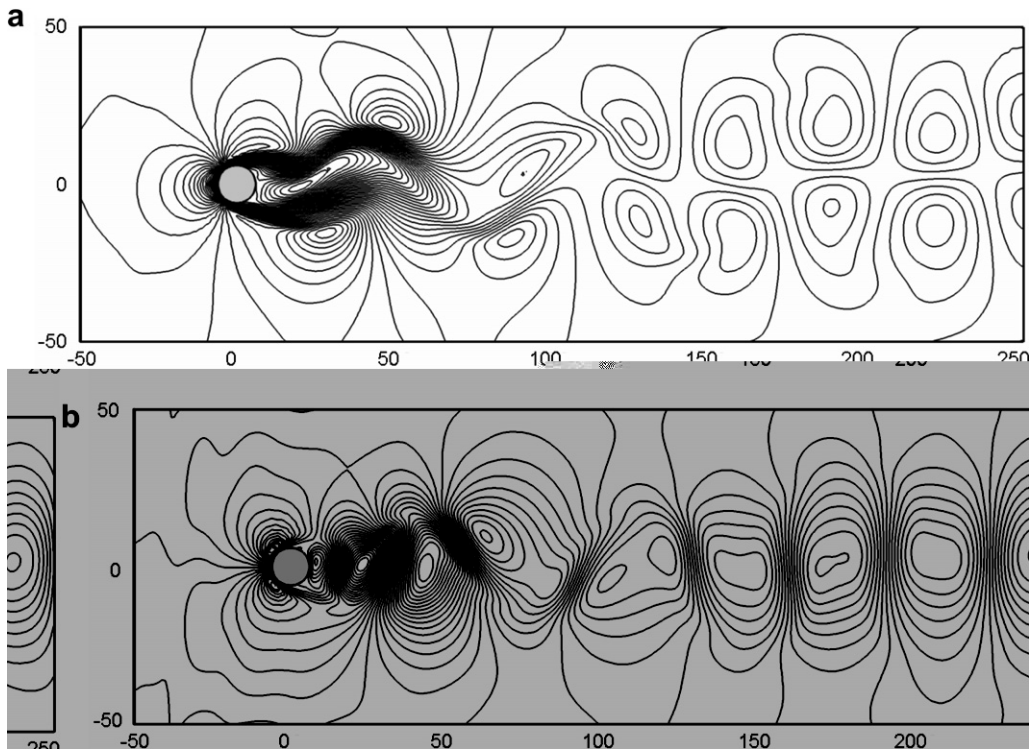

Fig. 23. Unsteady flow around a cylinder at $Re = 100$: (a) x-velocity contour and (b) y-velocity contour.

Section 2.3. For thermal problem with natural convection, by changing the magnitude of gravity artificially instead of number of grids the Rayleigh number can be adjusted in a large range (from 10^3 to 10^8 for different practical problems) without significantly increasing the computation load. Secondly, our proposed BC demonstrates a much better stability than any other BC used in the simulations. Both the unstable region and the achievable minimum τ value are much smaller than those of the other schemes. Additionally, in the normal region of Δ and τ, i.e., the stable region for all BCs, it gives second-order accuracy and comparable or better results than the other schemes. Finally, it is not sensitive to the interpolation (weighting) factor and choice of $\mathbf{u}_{bf}$. In conclusion, our mass conserving boundary condition for the LBE method not only eliminates the mass leakage and increases the stability of the code at the same time, but also can increase the efficiency of the LBE code for some practical problems.

References

 [1] R. Benzi, S. Succi, M. Vergassola, The lattice Boltzmann equation: theory and applications, Phys. Rep. 222 (1992) 145.
 [2] S. Chen, G.D. Doolen, Lattice Boltzmann method for fluid flows, Ann. Rev. Fluid Mech. 30 (1998) 329–364.
 [3] D. Yu, R. Mei, L.-S. Luo, W. Shyy, Viscous flow commutations with the method of lattice Boltzmann equation, Progr. Aerosp. Sci. 39 (2003) 329–367.
 [4] S. Succi, The Lattice Boltzmann Equation for Fluid Dynamics and Beyond, Oxford University Press, Oxford, 2001.
 [5] D.R. Noble, S. Chen, J.G. Georgiadis, R.O. Buckius, A consistent hydrodynamic boundary condition for the lattice Boltzmann method, Phys. Fluids 7 (1995) 203–209.
 [6] P. Ziegler, Boundary conditions for lattice Boltzmann simulations, J. Stat. Phys. 71 (1993) 1171–1177.
 [7] R. Mei, W. Shyy, On the finite difference-based lattice Boltzmann method in curvilinear coordinates, J. Comput. Phys. 143 (1998) 426–448.
 [8] T. Inamuro, M. Yoshino, F. Ogino, A non-slip boundary condition for lattice Boltzmann simulations, Phys. Fluids 7 (1995) 2928–2930.
 [9] A.J.C. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation, J. Fluid Mech. 271 (1994) 285–339.
[10] C.K. Aidun, Y. Lu, Lattice Boltzmann simulations of solid particles suspended in fluid, J. Stat. Phys. 81 (1995) 49–61.
[11] Q. Zou, X. He, On pressure and velocity boundary conditions for the lattice Boltzmann BGK model, Phys. Fluids 9 (1997) 1591–1598.
[12] S. Chen, D. Martínez, R. Mei, On boundary conditions in lattice Boltzmann method, Phys. Fluids 8 (1996) 2527–2536.
[13] O. Filippova, D. Hänel, Grid refinement for lattice-BGK models, J. Comput. Phys. 147 (1998) 219–228.
[14] R. Mei, L.-S. Luo, W. Shyy, An accurate curved boundary treatment in the lattice Boltzmann method, J. Comput. Phys. 155 (1999) 307–329.
[15] R. Mei, W. Shyy, D. Yu, L.-S. Luo, Lattice Boltzmann method for 3-D flows with curved boundary, J. Comput. Phys. 161 (2000) 680–699.
[16] H. Chen, Volumetric formulation of the lattice Boltzmann method for fluid dynamics: basic concept, Phys. Rev. E 58 (1998) 3955–3963.
[17] H. Chen, C. Teixeira, K. Molvig, Realization of fluid boundary conditions via discrete Boltzmann dynamics, Int. J. Mod. Phys. C 9 (8) (1998) 1281–1292.
[18] Y. Li, R. Shock, R. Zhang, H. Chen, Numerical study of flow past an impulsively started cylinder by the lattice-Boltzmann method, J. Fluid Mech. 519 (2004) 273–300.
[19] X. Shan, H. Chen, Lattice Boltzmann model for simulation flows with multiple phases and components, Phys. Rev. A 47 (1993) 1815–1819.
[20] X. Shan, H. Chen, Simulation of nonideal gases and liquid–gas phase transitions by the lattice Boltzmann equation, Phys. Rev. E 48 (1994) 2941–2948.
[21] X. Shan, G. Doolen, Multicomponent lattice-Boltzmann model with interparticle interaction, J. Stat. Phys. 81 (1995) 379–393.
[22] X. Shan, G. Doolen, Diffusion in a multicomponent lattice Boltzmann equation model, Phys. Rev. E 54 (1996) 3614–3620.
[23] P. Yuan, L. Schaefer, Equations of state in a lattice Boltzmann model, Phys. Fluids 18 (4) (2006) 042101–042111.
[24] A. D'Orazio, S. Succi, C. Arrighetti, Lattice Boltzmann simulation of open flows with heat transfer, Phys. Fluids 15 (9) (2003) 2778–2781.
[25] J. Buick, C. Greated, Gravity in the lattice Boltzmann model, Phys. Rev. E 61 (2000) 5307–5320.
[26] J.L. Hennessy, D.A. Patterson, Computer Architecture, fourth ed., Morgan Kaufman, Los Altos, CA, 2006.
[27] Z. Guo, C. Zheng, An extrapolation method for boundary conditions in lattice Boltzmann method, Phys. Fluids 14 (2002) 2007–2010.