Taylor Series Expansion and Least Squares-based Lattice Boltzmann Method

C. Shu
Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge
Crescent, Singapore 117576, e-mail: mpeshuc@nus.edu.sg

Keywords: Taylor series expansion, Least square optimization, Lattice Boltzmann Method

Abstract:

The Taylor series expansion- and least square-based lattice Boltzmann method (TLLBM) recently developed by us is a flexible lattice Boltzmann approach capable of simulating incompressible viscous flows with complex geometry. The method is based on the standard lattice Boltzmann equation (LBE), Taylor series expansion and the least squares optimization. The final formulation is an algebraic form and essentially has no limitation on the mesh structure and lattice model. In this paper, we will give the details of TLLBM and present some examples of its application.

1. Introduction

The development of the lattice Boltzmann method (LBM) as an alternative computational fluid dynamics approach has attracted more and more attentions in recent years [1]. However, because of the essential restriction of the standard lattice Boltzmann equation (LBE) to the lattice-uniformity, the broad application of the LBM in engineering has been greatly hampered. For many practical problems, an irregular grid or a meshless structure is always preferable due to the fact that curved boundaries can be described more accurately, and that computational resources can be used more efficiently with it.

The drawback of the standard LBE restricting to the lattice-uniformity comes from its precursor—the lattice gas cellular automata (LGCA). In the LGCA, the symmetry of lattice, which guarantees the isotropy of the fourth tensor consisting of particle velocities, is an essential condition to obtain the Navier-Stokes equations. By this condition, a particle at one lattice node must move to its neighboring node in one time step. This is the condition of lattice-uniformity. Although the LBE [2] with Bhatnagar-Gross-Krook (BGK) [3] model has made great improvements over the LGCA, it also inherits the feature of lattice-uniformity, which makes it macroscopically similar to a uniform Cartesian-grid solver.

Theoretically, the feature of lattice-uniformity is not necessary to be kept because the distribution functions are continuous in physical space. Currently, there are two ways to improve the standard LBM so that it can be applied to complex problems. One is the interpolation-supplemented LBM (ISLBM) proposed by He and his colleagues [4-5]. The other is based on the solution of a differential lattice Boltzmann equation (LBE). For complex problems, the differential LBE can be solved by the finite difference (FDLBE) method with the help of coordinate transformation [6] or by the finite volume (FVLBE) approach [7-8]. Numerical experiences have shown that these methods have good capability in real applications. However, the ISLBE has an extra computational effort for interpolation at every time step and it also has a strict restriction on selection of interpolation points, which requires upwind nine points for two-dimensional problems and upwind twenty-seven points for three-dimensional problems if a structured mesh is used. For the FDLBE and FVLBE methods, one needs to select efficient approaches such as upwind schemes to do numerical discretization in order to get the stable solution. As a consequence, the computational efficiency greatly depends on the selected numerical scheme. In addition, the numerical diffusion may affect the accuracy of results, especially in the region where the flow gradient is large.

In order to implement the LBE more efficiently for flows with arbitrary geometry, a new version of LBM, which is based on the standard LBM, the well-known Taylor series expansion, the idea of developing Runge-Kutta method, and the least squares approach, was proposed by Shu et al. [9-10]. The new approach is termed as Taylor series expansion- and least square-based lattice Boltzmann method (TLLBM). The final form of TLLBM is an algebraic formulation, in which the coefficients only depend on the coordinates of mesh points and lattice velocity, and are computed in advance. The new method is also free of lattice models. In recent years, TLLBM has been accurately and efficiently applied to simulate many incompressible flow problems [9-13]. In the following, we will first give a detailed description of TLLBM, and then present some examples of its application.

2. Taylor Series Expansion- and Least Squares-based Lattice Boltzmann Method (TLLBM)

The TLLBM [10] is based on the well-known fact that the density distribution function is a continuous function in physical space and can be well defined in any mesh system. Let us start with the standard LBE. The two dimensional, standard LBE with BGK approximation can be written as

$$f_\alpha(x+e_{\alpha x}\delta t, y+e_{\alpha y}\delta t, t+\delta t) = f_\alpha(x,y,t) + \frac{f_\alpha^{eq}(x,y,t) - f_\alpha(x,y,t)}{\tau}, \ \alpha = 0,1,\ldots,N, \qquad (1)$$

where τ is the single relaxation time; f_α is the density distribution function along the α direction; f_α^{eq} is its corresponding equilibrium state, which depends on the local macroscopic variables such as density ρ and velocity **U**(u, v); δt is the time step and $\mathbf{e}_\alpha(e_{\alpha x}, e_{\alpha y})$ is the particle velocity in the α direction; N is the number of discrete particle velocities. Obviously, the standard LBE consists of two steps: collision and streaming. The macroscopic density ρ and momentum density $\rho\mathbf{U}$ are defined as

$$\rho = \sum_{\alpha=0}^{N} f_\alpha, \qquad \rho\mathbf{U} = \sum_{\alpha=0}^{N} f_\alpha \mathbf{e}_\alpha \qquad (2)$$

Suppose that a particle is initially at the grid point (x,y,t). Along the α direction, this particle will stream to the position $(x+e_{\alpha x}\delta t, y+e_{\alpha y}\delta t, t+\delta t)$. For a uniform lattice, $\delta x = e_{\alpha x}\delta_t$, $\delta y = e_{\alpha y}\delta_t$. So, $(x+e_{\alpha x}\delta t, y+e_{\alpha y}\delta t)$ is at the grid point. In other words, equation (1) can be used to update the density distribution functions exactly at the grid points. However, for a non-uniform grid, $(x+e_{\alpha x}\delta t, y+e_{\alpha y}\delta t)$ is usually not at the grid point $(x+\delta x, y+\delta y)$. In the numerical simulation, we are only interested in the density distribution function at the mesh point for all the time levels. So, the macroscopic properties such as the density, flow velocity can be evaluated at every mesh point. To get the density distribution function at the grid point $(x+\delta x, y+\delta y)$ and the time level $t+\delta t$, we need to apply the Taylor series expansion or other interpolation techniques. In this work, the Taylor series expansion is used. Note that the time level for the position $(x+e_{\alpha x}\delta t, y+e_{\alpha y}\delta t)$ and the grid point $(x+\delta x, y+\delta y)$ is the same, that is, $t+\delta t$. So, the expansion in the time direction is not necessary. As shown in Fig.1, for simplicity, we let point A represent the grid point (x_A, y_A, t), point A' represent the position $(x_A+e_{\alpha x}\delta t, y_A+e_{\alpha y}\delta t, t+\delta t)$, and point P represent the position $(x_P, y_P, t+\delta t)$ with $x_P = x_A+\delta x, y_P = y_A+\delta y$. So, equation (1) gives

$$f_\alpha(A', t+\delta t) = f_\alpha(A,t) + \left[f_\alpha^{eq}(A,t) - f_\alpha(A,t)\right]/\tau \qquad (3)$$

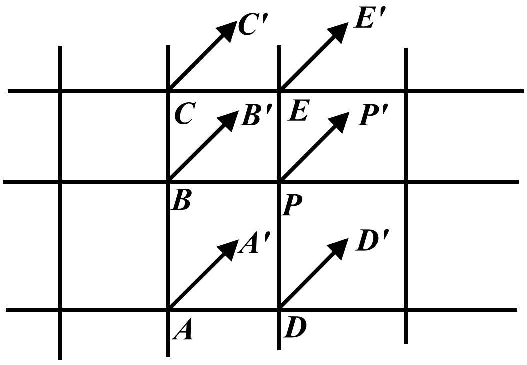


Fig. 1 Configuration of Particle Movement along the α Direction

For the general case, A' may not coincide with the mesh point P. At first, we consider the Taylor series expansion with truncation to the first order derivative terms. So, $f_\alpha(A', t+\delta t)$ can be approximated by the corresponding function and its derivatives at the mesh point P as

$$f_\alpha(A', t+\delta t) = f_\alpha(P, t+\delta t) + \Delta x_A \frac{\partial f_\alpha(P, t+\delta t)}{\partial x} + \Delta y_A \frac{\partial f_\alpha(P, t+\delta t)}{\partial y} + O[(\Delta x_A)^2, (\Delta y_A)^2] \quad (4)$$

where $\Delta x_A = x_A + e_{\alpha x}\delta t - x_P$, $\Delta y_A = y_A + e_{\alpha y}\delta t - y_P$. Note that the above approximation has a truncation error of the second order. Substituting equation (4) into equation (3) gives

$$f_\alpha(P, t+\delta t) + \Delta x_A \frac{\partial f_\alpha(P, t+\delta t)}{\partial x} + \Delta y_A \frac{\partial f_\alpha(P, t+\delta t)}{\partial y} = f_\alpha(A, t) + \frac{f_\alpha^{eq}(A, t) - f_\alpha(A, t)}{\tau} \quad (5)$$

It is indicated that equation (5) is a first order differential equation, which only involves two mesh points *A* and *P*. When a uniform grid is used, $\Delta x_A = \Delta y_A = 0$, equation (5) is reduced to the standard LBE (3). Solving equation (5) can provide the density distribution functions at all the mesh points. In this work, we try to develop an explicit formulation to update the distribution function. In fact, our new development is inspired from the Runge-Kutta method. As we know, the Runge-Kutta method is developed to improve the Taylor series method in the solution of ordinary differential equations (ODEs). Like equation (5), Taylor series method involves evaluation of different orders of derivatives to update the functional value at the next time level. For a complicated expression of given ODEs, this application is very difficult. To improve the Taylor series method, the Runge-Kutta method evaluates the functional values at some intermediate points and then combines them (through the Taylor series expansion) to form a scheme with the same order of accuracy. With this idea in mind, we look at equation (5). We know that at the time level $t+\delta t$, the density distribution function and its derivatives at the mesh point *P* are all unknowns. So, equation (5) has 3 unknowns in total. To solve for the 3 unknowns, we need 3 equations. However, equation (5) just provides 1 equation. We need additional 2 equations to close the system. As shown in Fig. 1, we can see that along the α direction, the particles at two mesh points P, B at the time level *t* will stream to the new positions P', B' at the time level $t+\delta t$. The distribution functions at these new positions can be computed through equation (1), which are given below

$$f_\alpha(P', t+\delta t) = f_\alpha(P, t) + \left[f_\alpha^{eq}(P, t) - f_\alpha(P, t)\right]/\tau \quad (6)$$

$$f_\alpha(B', t+\delta t) = f_\alpha(B, t) + \left[f_\alpha^{eq}(B, t) - f_\alpha(B, t)\right]/\tau \quad (7)$$

Using Taylor series expansion with truncation to the first order derivative terms, $f_\alpha(P', t+\delta t)$, $f_\alpha(B', t+\delta t)$ in above equations can be approximated by the function and its derivatives at the mesh point *P*. As a result, equations (6)-(7) can be reduced to

$$f_\alpha(P,t+\delta t)+\Delta x_P\frac{\partial f_\alpha(P,t+\delta t)}{\partial x}+\Delta y_P\frac{\partial f_\alpha(P,t+\delta t)}{\partial y}=f_\alpha(P,t)+\frac{f_\alpha^{eq}(P,t)-f_\alpha(P,t)}{\tau} \tag{8}$$

$$f_\alpha(P,t+\delta t)+\Delta x_B\frac{\partial f_\alpha(P,t+\delta t)}{\partial x}+\Delta y_B\frac{\partial f_\alpha(P,t+\delta t)}{\partial y}=f_\alpha(B,t)+\frac{f_\alpha^{eq}(B,t)-f_\alpha(B,t)}{\tau} \tag{9}$$

where $\Delta x_P = e_{\alpha x}\delta t$, $\Delta y_P = e_{\alpha y}\delta t$,

$\Delta x_B = x_B + e_{\alpha x}\delta t - x_P$, $\Delta y_B = y_B + e_{\alpha y}\delta t - y_P$

Equations (5), (8)-(9) form a system to solve for 3 unknowns. The solution of this system gives

$$f_\alpha(P,t+\delta t)=\Delta_P/\Delta \tag{10}$$

where $\Delta = \Delta x_A\Delta y_B - \Delta x_B\Delta y_A + \Delta x_B\Delta y_P - \Delta x_P\Delta y_B + \Delta x_P\Delta y_A - \Delta x_A\Delta y_P$

$$\Delta_P = (\Delta x_A\Delta y_B - \Delta x_B\Delta y_A)g_P + (\Delta x_B\Delta y_P - \Delta x_P\Delta y_B)g_A + (\Delta x_P\Delta y_A - \Delta x_A\Delta y_P)g_B$$

$$g_P = f_\alpha(P,t)+\left[f_\alpha^{eq}(P,t)-f_\alpha(P,t)\right]/\tau$$

$$g_A = f_\alpha(A,t)+\left[f_\alpha^{eq}(A,t)-f_\alpha(A,t)\right]/\tau$$

$$g_B = f_\alpha(B,t)+\left[f_\alpha^{eq}(B,t)-f_\alpha(B,t)\right]/\tau$$

Note that g_P, g_A, g_B are actually the post-collision state of the distribution function at the time level *t* and the mesh point *P, A, B* respectively. Equation (10) has the second order of truncation error, which may introduce a large numerical diffusion. To improve the accuracy of numerical computation, we need to truncate the Taylor series expansion to the second order derivative terms. For the two-dimensional case, this expansion involves 6 unknowns, that is, one distribution function, two first order derivatives, and three second order derivatives at the time level $t+\delta t$. To solve for these unknowns, we need 6 equations to close the system. This can be done by applying the second order Taylor series expansion at 6 points. As shown in Fig. 1, the particles at 6 mesh points P, A, B, C, D, E at the time level *t* will stream to positions P', A', B', C', D', E' at the time level $t+\delta t$. The distribution functions at these new positions can be computed through equation (1). Then by using the second order Taylor series expansion at these new positions in terms of the distribution function and its derivatives at the mesh point *P*, we can get the following equation system

$$g_i = \{s_i\}^T\{V\} = \sum_{j=1}^{6} s_{i,j}V_j\ ,\ \ i = P, A, B, C, D, E \tag{11}$$

where

$$g_i = f_\alpha(x_i,y_i,t)+\left[f_\alpha^{eq}(x_i,y_i,t)-f_\alpha(x_i,y_i,t)\right]/\tau$$

$$\{s_i\}^T = \{1, \Delta x_i, \Delta y_i, (\Delta x_i)^2/2, (\Delta y_i)^2/2, \Delta x_i\Delta y_i\}$$

$$\{V\} = \{f_\alpha, \partial f_\alpha/\partial x, \partial f_\alpha/\partial y, \partial^2 f_\alpha/\partial x^2, \partial^2 f_\alpha/\partial^2 y, \partial^2 f_\alpha/\partial x\partial y\}^T$$

g_i is the post-collision state of the distribution function at the *i*th point and the time level *t*, $\{s_i\}^T$ is a vector with 6 elements formed by the coordinates of mesh points, $\{V\}$ is the vector of unknowns at the mesh point *P* and the time level $t+\delta t$, which also has 6 elements, $s_{i,j}$ is the *j*th element of the vector $\{s_i\}^T$ and V_j is the *j*th element of the vector {*V*}. Our target is to find its first element $V_1 = f_\alpha(P,t+\delta t)$. Equation system (11) can be put into the following matrix form

$$[S]\{V\}=\{g\} \tag{12}$$

where $$\{g\} = \{g_P, g_A, g_B, g_C, g_D, g_E\}^T$$

$$[S] = [s_{i,j}] = \begin{bmatrix} \{s_P\}^T \\ \{s_A\}^T \\ \{s_B\}^T \\ \{s_C\}^T \\ \{s_D\}^T \\ \{s_E\}^T \end{bmatrix} = \begin{bmatrix} 1 & \Delta x_P & \Delta y_P & (\Delta x_P)^2/2 & (\Delta y_P)^2/2 & \Delta x_P \Delta y_P \\ 1 & \Delta x_A & \Delta y_A & (\Delta x_A)^2/2 & (\Delta y_A)^2/2 & \Delta x_A \Delta y_A \\ 1 & \Delta x_B & \Delta y_B & (\Delta x_B)^2/2 & (\Delta y_B)^2/2 & \Delta x_B \Delta y_B \\ 1 & \Delta x_C & \Delta y_C & (\Delta x_C)^2/2 & (\Delta y_C)^2/2 & \Delta x_C \Delta y_C \\ 1 & \Delta x_D & \Delta y_D & (\Delta x_D)^2/2 & (\Delta y_D)^2/2 & \Delta x_D \Delta y_D \\ 1 & \Delta x_E & \Delta y_E & (\Delta x_E)^2/2 & (\Delta y_E)^2/2 & \Delta x_E \Delta y_E \end{bmatrix}$$

$$\Delta x_C = x_C + e_{\alpha x}\delta t - x_P, \ \Delta y_C = y_C + e_{\alpha y}\delta t - y_P$$

$$\Delta x_D = x_D + e_{\alpha x}\delta t - x_P, \ \Delta y_D = y_D + e_{\alpha y}\delta t - y_P$$

$$\Delta x_E = x_E + e_{\alpha x}\delta t - x_P, \ \Delta y_E = y_E + e_{\alpha y}\delta t - y_P$$

The expressions of $\Delta x_P, \Delta y_P, \Delta x_A, \Delta y_A, \Delta x_B, \Delta y_B$ have been given previously. Since [S] is a 6×6 dimensional matrix, it is very difficult to get an analytical expression for the solution of equation system (12). We need to use a numerical algorithm to get the solution. Note that the matrix [*S*] only depends on the coordinates of mesh points, which can be computed once and stored for the application of equation (12) at all time levels.

In practical applications, it was found that the matrix [*S*] might be singular or ill-conditioned. To overcome this difficulty and make the method be more general, we introduce the least squares approach to optimize the approximation by equation (11). Equation (11) has 6 unknowns (elements of the vector {*V*}). If equation (11) is applied at more than 6 mesh points, then the system is over-determined. For this case, the unknown vector can be decided from the least squares method. For simplicity, let the mesh point *P* be represented by the index $i = 0$, and its adjacent points be represented by index $i = 1,2,...,M$, where *M* is the number of neighboring points around *P* and it should be larger than 5. At each point, we can define an error in terms of equation (11), that is,

$$err_i = g_i - \sum_{j=1}^{6} s_{i,j} V_j \ , \quad i = 0,1,2,...,M \tag{13}$$

The square sum of all the errors are defined as

$$E = \sum_{i=0}^{M} err_i^2 = \sum_{i=0}^{M} \left(g_i - \sum_{j=1}^{6} s_{i,j} V_j \right)^2 \tag{14}$$

To minimize the error *E*, we need to set $\partial E / \partial V_k = 0, k = 1,2,...,6$, which leads to

$$[S]^T [S]\{V\} = [S]^T \{g\} \tag{15}$$

where [*S*] is a $(M+1) \times 6$ dimensional matrix, which is given as

$$[S] = \begin{bmatrix} 1 & \Delta x_0 & \Delta y_0 & (\Delta x_0)^2/2 & (\Delta y_0)^2/2 & \Delta x_0 \Delta y_0 \\ 1 & \Delta x_1 & \Delta y_1 & (\Delta x_1)^2/2 & (\Delta y_1)^2/2 & \Delta x_1 \Delta y_1 \\ - & - & - & - & - & - \\ - & - & - & - & - & - \\ - & - & - & - & - & - \\ 1 & \Delta x_M & \Delta y_M & (\Delta x_M)^2/2 & (\Delta y_M)^2/2 & \Delta x_M \Delta y_M \end{bmatrix}_{(M+1)\times 6}$$

and $\{g\} = \{g_0, g_1, ..., g_M\}^T$

The Δx and Δy values in the matrix [*S*] are given as

$$\Delta x_0 = e_{\alpha x}\delta t \, , \; \Delta y_0 = e_{\alpha y}\delta t \tag{16a}$$

$$\Delta x_i = x_i + e_{\alpha x}\delta t - x_0 \, , \Delta y_i = y_i + e_{\alpha y}\delta t - y_0 \, , \text{for } i = 1,2,...,M \tag{16b}$$

Clearly, when the coordinates of mesh points are given, and the particle velocity and time step size are specified, the matrix [*S*] is determined. Then from equation (15), we obtain

$$\{V\} = \left([S]^T [S]\right)^{-1} [S]^T \{g\} = [A]\{g\} \tag{17}$$

Note that [*A*] is a 6×(*M*+1) dimensional matrix. From equation (17), we can obtain

$$f_\alpha(x_0, y_0, t + \delta t) = V_1 = \sum_{k=1}^{M+1} a_{1,k}^\alpha g_{k-1}^\alpha \tag{18}$$

where $a_{1,k}^\alpha$ are the elements of the first row of the matrix [*A*], which are pre-computed before the LBM is applied. Therefore, little computational effort is introduced as compared with the standard LBE. Note that the function g is evaluated at the time level *t*. So, equation (18) is actually an explicit form to update the distribution function at the time level $t + \delta t$ for any mesh point. In the above process, there is no requirement for the selection of neighboring points. In other words, equation (18) is nothing to do with the mesh structure. It only needs to know the coordinates of the mesh points. Thus, we can say that equation (18) is basically a meshless form.

3. Some Applications of TLLBM

Like the standard LBM, TLLBM only involves algebraic operation. But it can be applied to complex geometry with the use of non-uniform mesh. In the past two years, we have applied TLLBM to simulate many incompressible flow problems. Some simulation results are shown below.

The lid-driven flow in a square cavity is first tested by TLLBM with the use of non-uniform mesh. Numerical experiments showed that the TLLBM results with a mesh size of 145×145 for the case of Re=5000, 10000 agree very well with results of a Navier-Stokes solver using a mesh size of 257×257. This can be clearly observed in Fig. 2, which displays the velocity profiles along vertical and horizontal central lines at Re=10000. The streamlines of Re=5000, 10000 are shown in Fig. 3. The primary and tiny vortices are well captured by TLLBM.

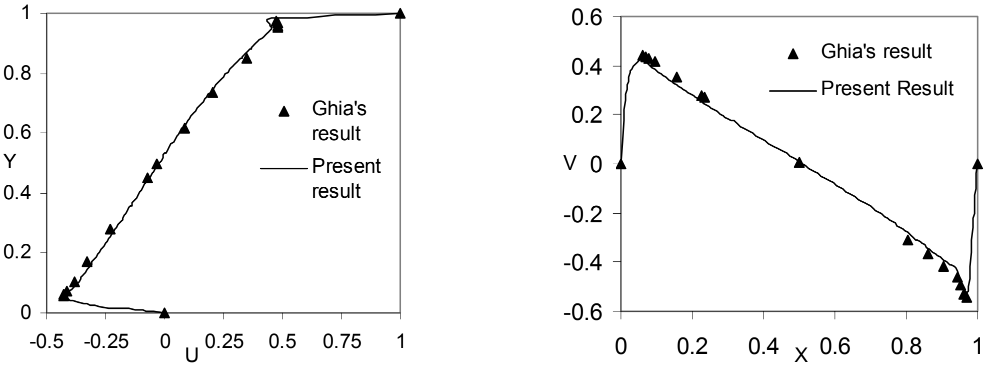
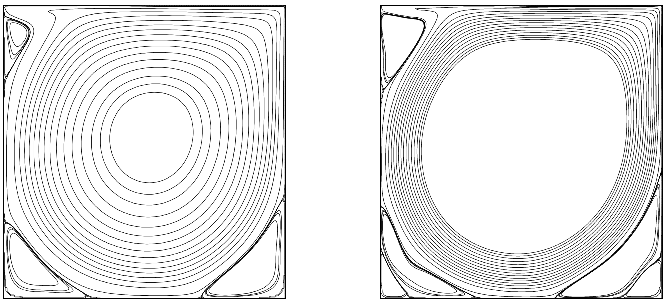


Fig. 2 U (left) and V (right) velocity profiles along vertical and horizontal central lines for a driven cavity flow (Re=10000, mesh size 145×145)



(a) Re=5000, 145×145 (b) Re=10000, 145×145

Fig. 3 Streamlines for the flow in a lid-driven square cavity

To test the ability of TLLBM for the treatment of curved boundary, the flow past a circular cylinder is simulated. The simulation includes the steady case and the unsteady case. Again, the TLLBM results are in good agreement with those obtained by experiment and Navier-Stokes solvers. The streamlines of a steady case (Re=20) are shown in Fig. 4, while the streamlines of an unsteady case (Re=100) are depicted in Fig. 5.

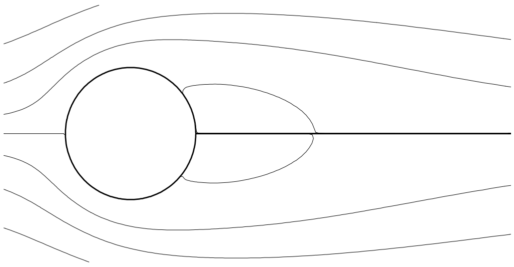
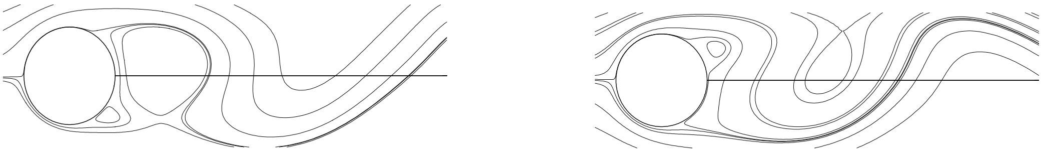


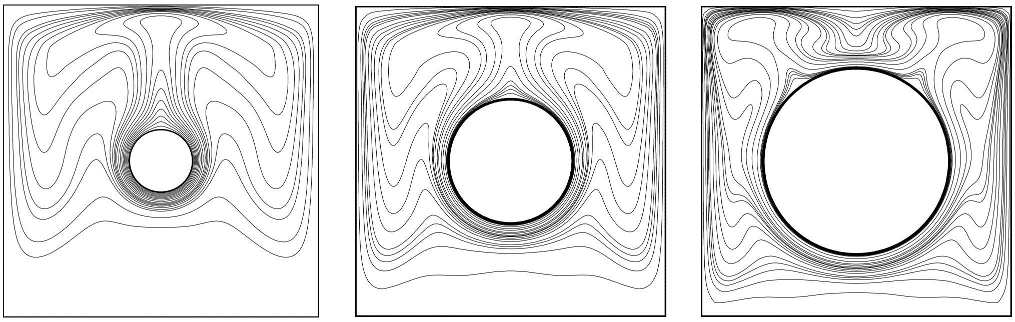
Fig. 4 Streamlines of a steady flow past a circular cylinder (Re=20)



(a) t=T/8 (b) t=7T/8

Fig. 5 Streamlines of an unsteady flow past a circular cylinder (Re=100)

Apart from the simulation of isothermal flows, TLLBM has also been applied to simulate thermal flow problems with curved boundary and the use of non-uniform mesh. An example of such application is natural convection in an annulus between a square outer cylinder and a circular inner cylinder, in which the inner cylinder is hot while the outer cylinder is cold. It was found that the computed Nusselt numbers by TLLBM agree very well with those from Navier-Stokes solvers for all simulated cases. Fig. 6 displays isotherms of three typical cases.



(a) *rr*=5 (b) *rr*=2.5 (c) *rr*=1.67

Fig. 6 Isotherms of 3 typical cases (Ra=10^6)

4. Conclusions

This paper demonstrates that the Taylor series expansion- and least squares-based lattice Boltzmann method (TLLBM) is an efficient approach for simulation of isothermal and thermal incompressible flows. The beauty of the TLLBM is that it still keeps the local and explicit features of the standard lattice Boltzmann method. Therefore, it is able to exploit fully the power of parallel computing. The other advantage of the present method is that it is easy for application to flow problems with complex geometry. Furthermore, the computational efficiency of the present method is competitive as compared with that of the standard LBE and of the conventional CFD solvers.

References

1. Chen S. and Doolen GD, *Annu. Rev. Fluid Mech.*, 30, p329, 1998.
2. Koelman JMVA, *Europhys. Lett.*, 15, p603, 1991.
3. Bhatnagar PL, Gross EP and Krook M., *Phys. Rev.,* 94, p511, 1954.
4. He X., Luo L-S and Dembo M., *J. Comp. Phys.*, 129, p357, 1996.
5. He X. and Doolen GD., *Phys. Rev. E, 56*, p434, 1997
6. Mei R. and Shyy W., *J. Comp. Phys.*, 134, p306, 1997.
7. Chen H, Phys. Rev. E 58: 3955 (1998).
8. Peng G., Xi H., Duncan C. and Chou S. H., *Phys. Rev. E*, 59, p4675, 1999.
9. Shu C, Chew YT and Niu XD, *Phys. Rev. E.,* vol. 64, 045701, p1, 2001.
10. Shu C, Niu XD and Chew YT, *Phys. Rev. E.*, vol. 65, 036708, p1, 2002.
11. Lim CY, Shu C, Niu XD and Chew YT, *Phys. Fluids,* vol. 14, No. 7, p2299,2002.
12. Niu XD, Shu C and Chew YT, *Int. J. Mod. Phys. B*, vol.17, No. 1&2, p161, 2003.
13. Peng Y., Chew Y. T. and Shu C., *Phys. Rev. E* 67, 026701, 2003.