The First International Symposium on Optimization and Systems Biology (OSB'07)
Beijing, China, August 8–10, 2007
Copyright © 2007 ORSC & APORC, pp. 451–458

Parallel Simulation of Compressible Fluid Dynamics Using Lattice Boltzmann Method*

Bing He[1,†] Wei-Bing Feng[1,‡] Wu Zhang[1,§]
Yu-Ming Cheng[2,¶]

[1] School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China
[2] Shanghai Institute of Applied Mathematics and Mechanics, Shanghai University,
Shanghai 200072, China

Abstract The Lattice Boltzmann Method (LBM) has numerous computational advantages, such as the simplicity of programming, the ability to incorporate microscopic interactions, and the easy parallelization of algorithms. The traditional lattice Boltzmann model has a constraint of small mach number(the velocity of fluid must be less than 0.3 mach). This paper presents a novel lattice Boltzmann method to simulate compressible flows, Appling the model in computing the velocity of flows between 0.3 mach to 0.7 mach. The numerical experiments show that our algorithm is more numerical stable than traditional LBM methods when the Reynolds number is large. The parallel performance of this algorithm on workstation clusters is presented in the end.

Keywords Lattice Boltzmann method; compressible fluid dynamics; parallel computing; cache optimization.

1 Introduction

In recent years the lattice Boltzmann method (LBM) has attracted much attention in the physics and engineering communities as a possible alternative approach for solving complex fluid dynamics problems. In particular, the inherent parallelism, the simplicity of programming, and the capability of incorporating complex microscopic interactions have made LBM a very attractive simulation method for fluid flow in complex physical systems. Unfortunately, as a CFD tool, the general LB method developed in the past suffered from the constraint of small Mach number.

Some models have been made to increase the allowable Mach number range and to incorporate the effects of temperature into lattice Boltzmann simulations. Choosing a modified equilibrium distribution, Alexander, Chen, and Doolen[1] replicated the Burger's equation with a controllable sound speed. Yu and Zhao[11] introduced

[*] This research is supported by Natural Science Foundation of shanghai under 05ZR14050

[†] Email: hebing@shu.edu.cn.
[‡] Email: wbfeng@staff.shu.edu.cn
[§] Email: wzhang@shu.edu.cn
[¶] Email: ymcheng@staff.shu.edu.cn

an attractive force to reduce the sound speed and to alleviate the small Mach number restriction; however, the energy equation was not recovered in their formulation. Palmer and Rector[6] formulated a thermal model that can simulate temperature variations in a flow, but high Mach number effects were not included in that study.

Recently, we apply a novel Lattice Boltzmann methods to simulate cavity flows that the velocity of flows is great than 0.3 Ma.

2 Lattice Boltzmann equations

This study simulates the compressible flow using the 12-velocity LBM model[7] with a 2D square lattice, In this model, $c = \delta x/\delta t$ is the lattice streaming speed and δx and δy are the grid spacing in the x and y directions, respectively, and correspond to the distance which a particle moves in each time step of the LBM simulation. The discrete velocities for the model are defined as:

$$e_{A\alpha} = (\sqrt{6},0)_{FS}, \qquad W_A = \frac{1}{36}; \tag{1}$$

$$e_{B\alpha} = (\pm\frac{\sqrt{9-3\sqrt{5}}}{2},\pm\frac{\sqrt{9-3\sqrt{5}}}{2}), \qquad W_B = \frac{5+2\sqrt{5}}{45}; \tag{2}$$

$$e_{C\alpha} = (\pm\frac{\sqrt{9+3\sqrt{5}}}{2},\pm\frac{\sqrt{9+3\sqrt{5}}}{2}), \qquad W_C = \frac{5-2\sqrt{5}}{45}. \tag{3}$$

The subscript FS denotes a fully symmetric set of points.W_A, W_B, W_C is weight.

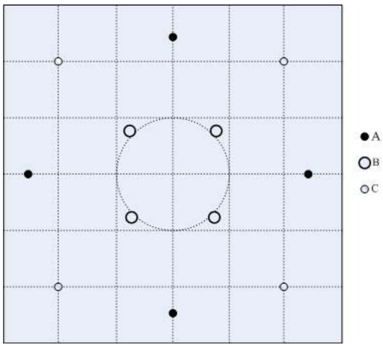

Figure 1: Abscissae of two degree-7 Gauss-Hermit quadrate formulate in two-dimensions. the unit circle in the middle depicts the sound speed.

The governing equation for the density distribution function is given by:

$$f_{i\alpha}(x+e_{i\alpha}\delta t, t+\delta t) - f_{i\alpha}(x,t) = -\frac{1}{\tau}(f_{i\alpha}(x,t) - f_{i\alpha}^{eq}(x,t));$$
$$(i = A,B,C), (\alpha = 1,2,3,4). \tag{4}$$

where τ characterizes the relaxation time of the density distribution function towards the local equilibrium f_{α}^{eq} . The equilibrium density distribution is expressed as:

$$\begin{aligned} f_{i\alpha}^{eq} &= W(e_{i\alpha})\rho\{1+e_{i\alpha}\cdot v+\frac{1}{2}((e_{i\alpha}\cdot v)^2 - v\cdot v+(\theta-1)(e_{i\alpha}\cdot e_{i\alpha}-D)); \\ &+\frac{e_{i\alpha}\cdot v}{6}[(e_{i\alpha}\cdot v)^2-3(v\cdot v)+3(\theta-1)(e_{i\alpha}\cdot e_{i\alpha}-D-2)]\}. \end{aligned} \tag{5}$$

The macroscopic density, temperature and velocity are calculated from:

$$\rho = \sum_{i=A}^{C} \sum_{\alpha=1}^{4} f_{i\alpha}; \tag{6}$$

$$\rho v = \sum_{i=A}^{C} \sum_{\alpha=1}^{4} f_{i\alpha} e_{i\alpha}; \tag{7}$$

$$\rho(D\theta + v \cdot v) = \sum_{i=A}^{C} \sum_{\alpha=1}^{4} f_{i\alpha} e_{i\alpha} \cdot e_{i\alpha}. \tag{8}$$

3 Numerical simulation

3.1 Validation of results for square cavity

Lid-driven flow in a two-dimensional square cavity is a classical, benchmark problem with which LBE simulations have been extensively validated [5][4][2]. Hou [5] presented a detailed analysis of this flow to demonstrate the capabilities of the LBE method. They made extensive comparison of LBE results with NS-solution results of Ghia [3]. Except for high Reynolds numbers (*Re* >5000), the LBE simulation results are better than the conventional (NS) results.

Following the literature [8] for the square-cavity problem, Dirichlet boundary conditions are used: Left and right walls: $u, v = 0$ at $x = (0, N)$ Bottom wall: $u, v = 0$ at $y = 0$ Top wall: $u = u_{\text{ref}}, v = 0$ at $y = N$ These physical boundary conditions may be implemented within the LBE code in different ways.

In the present simulations, the walls coincide with a corresponding line of lattice nodes. On the three stationary walls, the on-grid bounce-back model [5][9] is implemented because its numerical implementation is straightforward and yet sufficient for LBE simulations of fluid flows in simple bounded domains.

For the moving lid, the equilibrium scheme [5] is used. In all the simulations presented here, we choose a lid-velocity of $u_{\text{ref}} = 0.5$.

In this section, first we present a comparison of the steady results for a square cavity with the conventional benchmark results of Ghia [3]. Herein. Reynolds number was increased beyond a value of 3200, numerical instabilities were observed in D2Q9 model. But my model is numerical stable.

A comparison of the predicted location of the primary vortex at different Reynolds numbers is shown Fig 2. The present results are compared with the NS-results of Ref.[3] and the LBE results of Ref[5]. It is seen that the comparison result is excellent at all Reynolds numbers. From these comparisons, we are confident in the predictions of our LBE code.

Table 1. Vortex Location

Re	Primary eddy	Bottom-left corner eddy	Bottom-right corner eddy
	(x,y)	(x,y)	(x,y)
400[a]	(0.5547,0.6055)	(0.0508,0.0469)	(0.8906,0.1250)

400[b]	(0.5608,0.6078)	(0.0549,0.0510)	(0.8902,0.1255)
400[c]	(0.5513,0.5972)	(0.0570,0.0523)	(0.8857,0.1225)
1000[a]	(0.5313,0.5625)	(0.0859,0.0781)	(0.8594,0.1094)
1000[b]	(0.5333,0.5647)	(0.0902,0.0784)	(0.8667,0.1137)
1000[c]	(0.5318,0.5621)	(0.0868,0.0824)	(0.8644,0.1105)
3200[a]	(0.5165,0.5469)	(0.0859,0.1094)	(0.8125,0.0859)
3200[c]	(0.5314,0.5331)	(0.0860,0.1175)	(0.8340,0.0800)

[a]Ghia[4]. [b]Hou[5].[c]Present results.

In Table 1, the results extracted from the present simulations are all summarized and compared with the results of Ghia[3] and Hou[5]. Specifically, listed in Table 1 are the location of primary eddies. It is observed from Table 1 that the present LBE simulation results for the square cavity agree very well with previous results of Refs[5][3].

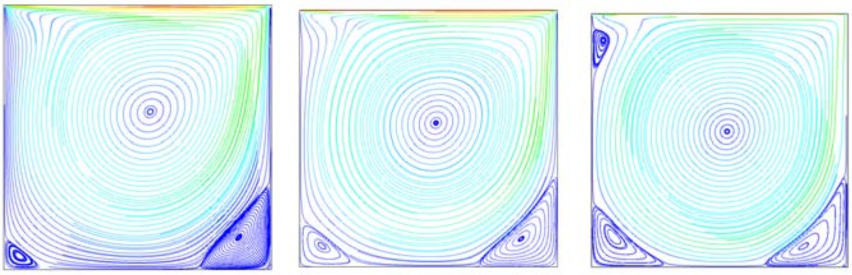

Figure 2: lid cavity flows

3.2 Validation of results for Poiseuille flow

The novel LB model was verified by implementing a steady fluid flow in a channel with a width of 30 nodes. No slip boundary conditions are imposed through the introduction of walls at the top and bottom of the lattice. The parabolic velocity boundary condition was applicable to inlet. This method provides a way to verify the result of velocity of the fluid flow.

$$\begin{aligned} V_x(y) &= \frac{3}{2}V_{avg}(1-\frac{y^2}{h^2}) \qquad (9) \\ V_{max} &= \frac{3}{2}V_{avg}. \qquad (10) \end{aligned}$$

V_{avg} is a average velocity. V_{max} is a maximum velocity got from eq(10).

Fig.3 shows the velocity profile of the fluid by taking a vertical cut of the channel at a length of 150. X-velocity component of the results was divided to V_{max} . The solid line is the analytical solution, and the points are the data results obtained from the simulation. The analytical solution shows that the velocity flow profile in the channel is parabolic. The flow velocity will be highest at the middle of the channel. The simulation shows excellent agreement with this behavior.

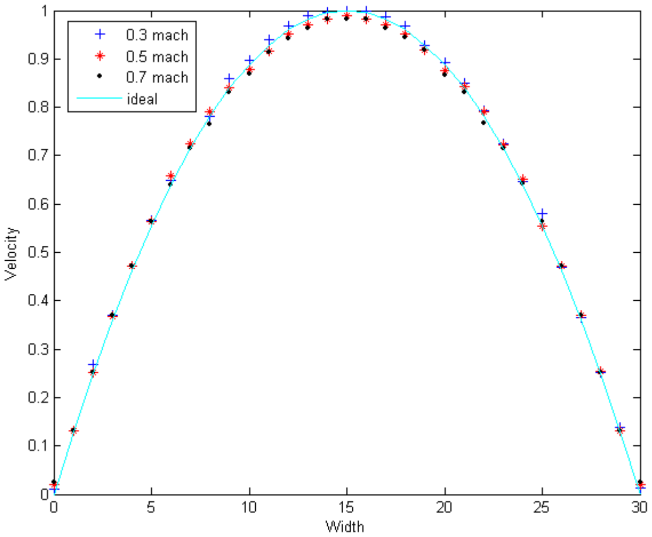

Figure 3: X-Velocity component profile of channel flow at length 150. Solid line is the analytical solution. Points are my simulation results.

Testing for velocity using the Poiseuille Channel benchmark, the model was executed in a channel with a length of 300 and width of 30 lattice points under no-slip boundary conditions. The time relaxation parameter was set to 1.2. The model was executed for 8000 time steps and the data was extracted at the end of the simulation.

4 Parallel Implementation of the LB method

The computations of the LB method can be summarized as follows:

1. Setup an initial fluid velocity field;
2. Calculate the equilibrium particle distribution from velocity, density and temperature field based on equation (5);
3. Perform the collision operation at each node;
4. Communicate the post collision distribution data from appropriate nodes to neighboring processors;
5. Calculate particle distribution function at each node from the post collision distribution using second order accurate interpolation;
6. Update the particle distribution on physical boundaries based on boundary conditions;
7. Calculate the fluid velocity, density and temperature at each node using equations eq(6),eq(7),eq(8);
8. Repeat from (2) to (7) until the flow field reaches a steady state or until enough cycles of the flow field are obtained.

From a computational perspective, steps(2),(3),(5)and (7) are extremely important as they account for nearly all the computations. In the actual implementation, steps(7),(2)and(3)can be done simultaneously as the result data can be reused, hence, these data can be put into the cache to improve the computing speed. Mapping the arrays into the L1 and L2 cache properly also results in further improvement of speed. All computational kernels have been written in C.

5 Cache optimization

Obtaining data from the main memory in every computational cycle is time consuming, while CPU remains idle during this process. If the data were accessed from cache in every computational cycle, the time consumed by data transfer would be much less. To obtain good performance on cache-based architectures, the numerical algorithms would have to divide the data (computational domain) into blocks (subdomains), such that the blocks can fit into cache and then be utilized repeatedly. Not all algorithms are applicable to this kind of cache optimization, as data dependencies will disallow updating subdomains separately. While in lattice Boltzmann algorithm, only the nearest neighbors have data dependencies, so it's much applicable to cache optimization.

The LBM simulates the two-dimensional fluid using twelve distribution functions. Therefore the largest grid size that can be accommodated within an 4MB size cache for a square grid is 200×200,

12 arrays×200^2 elements×8 bytes/elements=3.75MB.

To accommodate larger grid sizes in the cache, the grids need to be divided into subsections that can be fitted in cache. Cache optimization[10] can be obtained by performing computations separately for each subsection in several time steps. Cache optimization is possible for LBM , as the computations are completely local in nature, such as the computation of the RHS of Eq11, where the computation of the equilibrium distributions are included. They do not involve any data dependencies between the subsections:

$$\tilde{f}_{i\alpha}(x,t+\delta t)-f_{i\alpha}(x,t)=-\frac{1}{\tau}(f_{i\alpha}(x,t)-f_{i\alpha}^{eq}(x,t)); \\ (i=A,B,C),(\alpha=1,2,3,4) \tag{11}$$

The propagation substep is an assignment operation of an almost local nature due to nearest-neighbor dependencies

$$f_{i\alpha}(x+e_{i\alpha}\delta t,t+\delta t)=\tilde{f}_{i\alpha}(x,t+\delta t); \quad (i=A,B,C),(\alpha=1,2,3,4). \tag{12}$$

Due to the nearest-neighbor data dependencies, a limit is placed on the number of time steps, in which computations are performed separately for each subsection of the grid [10]. A parallel lattice Boltzmann algorithm is implemented by decomposing the domain into vertical strips and assigning each strip to a processor.

6 Results

To test the computational efficiency and performance of LBM, parallel speedup on single and multiple processors are measured. The results were computed on a workstation cluster, which has 64 processors and 4MB cache, each processor has 1GB main memory. MPI is used as the message-passing library.

We measure the parallel speedup to access the performance of the parallel LBM. The number of grid points range from 1000×1000 to 6000×6000. The computations are performed for 2000 time steps. In this paper, parallel speedup is defined as:

Parallel speedup=compute time for serial case/compute time for parallel case.

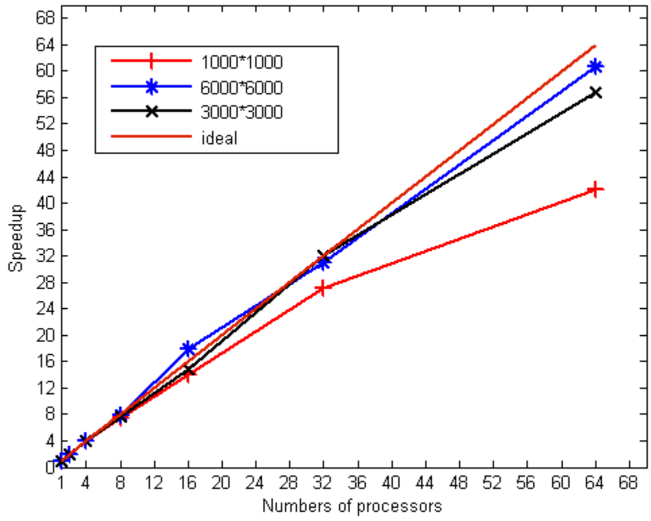

Figure 4: Parallel speedup of Lattice Boltzmann method.

Fig.4 shows the parallel speedup of the cache-optimized LBM. These speedups are linear in almost all cases. The 1000×1000 case is an exception because there is not much computation to be done when the number of processors increases.

7 Conclusions

LBM takes significantly less compute time for unsteady flows when it can take the same time step as traditional finite difference methods and not violate any stability criteria. Parallel speedup for LBM is linear. It is expected that LBM will perform far better for unsteady Navier-Stokes simulations because conventional methods are required to solve a Poisson equation at every time step for computing pressure whereas LBM can explicitly calculate pressure from the density (sum of distribution functions). However, for solving the Navier–Stokes equations, the LBM model requires twelve distribution functions at each grid point, which would mean greater memory requirements.

References

[1] F. J. Alexander, H. Chen, and G. D. Doolen. Lattice Boltzmann model for compressible fluids. *Phys. Rev. A*, 46(4):1967–1970, 1992.

[2] Y.T. Chew, C. Shu, and X.D. Niu. A new differential lattice boltzmann equation and its application to simulate incompressible flows on non-uniform grids. *J. Stat. Phys. (USA)*, 107(1-2):329 – 42, 2002/04.

[3] U. Ghia, K.N. Ghia, and C.T. Shin. High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. *J. Comput. Phys. (USA)*, 48(3):387 – 411, 1982/12.

[4] Zhaoli Guo, Baochang Shi, and Nengchao Wang. Lattice bgk model for incompressible navier-stokes equation. *J. Comput. Phys. (USA)*, 165(1):288 – 306, 2000/11/20.

[5] S. Hou, Q. Zou, S. Chen, and G. Doolen. Simulation of cavity flow by the lattice boltzmann method. *Journal of Computational Physics*, 118(2):329 –, 1995.

[6] B.J. Palmer and D.R. Rector. Lattice boltzmann algorithm for simulating thermal flow in compressible fluids. *J. Comput. Phys. (USA)*, 161(1):1 – 20, 2000/06/10.

[7] Xiaowen Shan, Xue-Feng Yuan, and Hudong Chen. Kinetic theory representation of hydrodynamics: a way beyond the navier-stokes equation. *J. Fluid Mech. (UK)*, 550:413 – 41, 2006/03/10.

[8] P.N. Shankar and M.D. Deshpande. Fluid mechanics in the driven cavity. *Annual Review of Fluid Mechanics*, 32:93 – 136, 2000.

[9] SAURO SUCCI. The lattice boltzmann equation for fluid dynamics and beyond. *Oxford University Press*, pages 1 – 265, 2001.

[10] A.C. Velivelli and K.M. Bryden. A cache-efficient implementation of the lattice boltzmann method for the two-dimensional diffusion equation. *Concurrency Computation Practice and Experience*, 16(14):1415 – 1432, 2004.

[11] Huidan Yu and Kaihua Zhao. Lattice boltzmann method for compressible flows with high mach numbers. *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top. (USA)*, 61(4):3867 – 70, 2000/04.