Simulation of Lid-driven Cavity Flow by Parallel Implementation of Lattice Boltzmann Method on GPUs

S. Berat Çelik[1], Cüneyt Sert[2], Barbaros ÇETİN[3]

[1,2] METU, Mechanical Engineering, Ankara, TURKEY
[3] METU-NCC, Mechanical Engineering, TRNC Mersin 10 TURKEY
[1] berat.celik@metu.edu.tr, [2]csert@metu.edu.tr, [3]barbaros@metu.edu.tr

Abstract. Lattice Boltzmann Method (LBM) is a computational technique used to solve transport problems in thermo-fluidic sciences. When applied to the solution of incompressible flows it shows a number of advantages compared to classical finite volume or finite element based techniques. Its algorithmic structure is very suitable for parallel programming on GPUs, which are today's state-of-the-art technology for parallel computing. In this study, 2D lid-driven cavity problem, which is a standard benchmark problem for fluid flow simulations, is solved using LBM with D2Q9 model and BGK collision approximation. Numerical computations are performed on a workstation with an NVIDIA Tesla C1060 GPU. Parallelization potential of a typical LBM code is investigated. Performance comparison with CPU- and GPU-based computing is presented.

Keywords: *Lattice-Boltzmann Method, GPU computing, Lid-driven cavity*

1 Introduction

Lattice Boltzmann Method (LBM) is a new generation numerical technique used to study transport problems in thermo-fluidic sciences. It is well suited to simulate single and multi-phase flows of single and multi-component fluids. Although it is commonly used for problems in high Knudsen number regimes for which the continuum assumption of Navier-Stokes solvers breaks down, it is also capable of simulating continuum flows [1-5]. Its simple algorithmic structure is easy to code and very suitable for parallel programming [6].

Today's state-of-the-art technology for parallel computing is based on the use of Graphics Processing Units (GPU). GPUs that are specifically designed for high performance scientific computing can offer tens of times of more FLOP performance and over ten times more memory bandwidth compared to CPUs [6]. In 2007, GPU manufacturer NVIDIA announced a new parallel computing architecture called CUDA, which popularized the use of GPUs for general purpose computing. In the following years, software vendors realized the great potential of GPU based scientific computing, and they started to migrate their CPU based numerical libraries to GPUs. Jacket is such a commercial numerical library that provides GPU-based versions of many built-in MATLAB functions. MATLAB and Jacket software are used in this study to implement LBM.

2D lid-driven cavity problem is solved using LBM with D2Q9 model and BGK collision approximation. Numerical computations are performed on a workstation with a Tesla C1060 GPU. Performance comparison with CPU-based and GPU-based computing is presented. Speedups exceeding 17 times are observed. It is seen that computations using finer grids resulted in better parallel performance.

2 Lattice-Boltzmann Method

Although incompressible flow of a 2D lid-driven cavity, shown in Fig. 1, is commonly used as a benchmark problem for new CFD codes, it is quite challenging considering boundary condition related singularities at its top corners. Classical finite volume or finite element based solvers need excessively fine girds to resolve these regions accurately and provide an oscillation free solution. A major challenge for solving incompressible flow problems with these classical techniques is the weak coupling of pressure and velocity fields. Several remedies to overcome this problem are suggested such as the use of staggered grids for finite volume solvers or the use of different order approximation spaces for velocity and pressure fields for finite element solvers. However, all these extra efforts bring complexities to computer codes. Several stabilization techniques are developed to circumvent these complications, but they suffer from unphysical artificial dissipation.

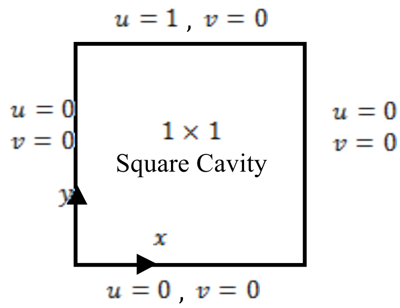


Fig. 1. Definition of 2D lid-driven cavity problem.

In this study LBM is used to solve 2D lid-driven cavity problem. Difficulties of simulating incompressible flows mentioned in the previous paragraph are not of a concern for LBM, which results in a very short and clean code that is easy to understand and implement. A typical LBM code works in a time marching manner, in which the following 4 sections are repeated inside a time loop: **(1)** collision, **(2)** streaming, **(3)** implementation of boundary conditions, and **(4)** calculation of macroscopic quantities. In this study, the popular D2Q9 model is used [7]. D2 stands for 2 space dimensions, and Q9 is the number of lattice directions associated with each node of the mesh, as shown in Fig. 2.

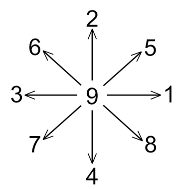


Fig. 2. Lattice directions of D2Q9 model.

2.1 Collision Process

Governing equation of the method is the *Boltzmann Transport Equation (BTE)* [7],

$$\frac{\partial f}{\partial t} + (c \cdot \nabla) f = \Omega \tag{1}$$

where Ω term is a double integral that represents the collisions of particles. f is the distribution function used to model the number density of the particles in a control volume and in a range of velocities at any time t. c is the speed of sound. BTE is an integro-differential equation which is hard to solve. It is commonly simplified using the Bhatnagar-Gross-Krook (BGK) model, according to which the collision operator is replaced with the following expression [8],

$$\Omega = \omega \left(f^{eq} - f\right) \tag{2}$$

where ω is collision frequency and f^{eq} is equilibrium distribution function. BTE equation can now be discretized in space and time as follows,

$$f(x + \Delta x, t + \Delta t) = f(x, t)(1 - \omega) + \omega f^{eq}(x, t) \tag{3}$$

By the use of Chapman-Enskog expansion, i.e. single-relaxation time approach, collision frequency is given by,

$$\omega = \frac{1}{0.5 + \mu\left(\frac{3\Delta t}{\Delta x^2}\right)} \tag{4}$$

where μ is the nondimensional kinematic viscosity which is a function of relaxation time $\tau = 1/\omega$. In the common practice, both the time step Δt and the grid spacing Δx are taken to be unity, which is also the case in the current implementation.

Equilibrium distribution function of Eqn (3) is simplified by Rothman & Zaleski in the following form [9],

$$f_n^{eq} = w_k\, \rho_n \left[1 + \frac{\vec{c_k} \cdot \vec{V_n}}{c_s^2} + \frac{\frac{1}{2}\left(\vec{c_k} \cdot \vec{V_n}\right)^2}{c_s^4} - \frac{\frac{1}{2}\left|\vec{V_n}\right|^2}{c_s^2}\right] \tag{5}$$

which is the equation used to calculate the equilibrium distribution function at node n of the grid. $\vec{c_k}$ are the unit vectors associated with each lattice direction. For example, for the 5th direction of Fig. 2 $\vec{c_5} = \vec{i} + \vec{j}$. $\vec{V_n}$ and ρ_n are the macroscopic velocity and density at node n and c_s is the lattice speed of sound. Weights used in D2Q9 model are,

$$w_k = \begin{cases} 1/9 & \text{for} \quad k = 1, 2, 3, 4 \\ 1/36 & \text{for} \quad k = 5, 6, 7, 8 \\ 4/9 & \text{for} \quad k = 9 \end{cases} \tag{6}$$

2.2 Streaming Process

Streaming (updating) process transfers the recently calculated distribution functions to the neighboring nodes to be used in the next time step. Fig. 3 demonstrates how the nodes communicate information with each other for the simple case of a unidirectional flow from left to right discretized by a grid of 6 nodes (2×3). Arrows represent 9 distribution functions and the one in the x-direction (1st direction) is highlighted and shown before and after streaming.

At the end of streaming, distribution functions of the nodes lying on the left boundary (nodes 1 and 4) are missing and their values should be assigned using a boundary condition. Also see that the functions on the very right side of the figure have left the domain.

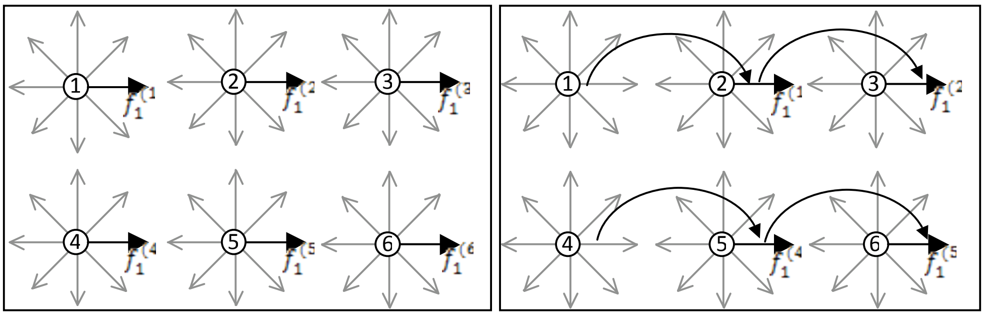


Fig. 3. 1st distribution function before (left) and after (right) streaming.

2.3 Boundary Conditions

For the fixed solid walls of the cavity solved in the current study, first-order accurate bounce-back boundary condition is used. For example at the bottom boundary of the cavity problem 4th , 7th and 8th distribution functions are located outside the fluid domain. To get the required no slip boundary condition their values are assigned to the symmetric ones inside the fluid domain, i.e.

$$f_4 = f_2, \qquad f_7 = f_5, \qquad f_8 = f_6 \tag{7}$$

For the moving top wall of the cavity, a modified version of the procedure described above given by Zhu and He [11] is used.

2.4 Macroscopic Quantity Calculations

Summation of all 9 distribution functions at a node results in the lattice density ρ

$$\rho = \sum_{k=1}^{9} f_k \tag{8}$$

which can further be used to calculate the macroscopic velocity at a node as follows

$$\vec{V} = \frac{1}{\rho}\sum_{k=1}^{9} f_k \overrightarrow{c_k} \tag{9}$$

3 Results and Discussion

LBM described in the previous section is coded in MATLAB and 2D lid-driven cavity problem is solved at 4 different Reynolds numbers. Streamlines and velocity profiles at $x = 0.5$ are shown in Figs. 4 and 5. LBM results are in good agreement with the reference results of Ghia et.al [12].

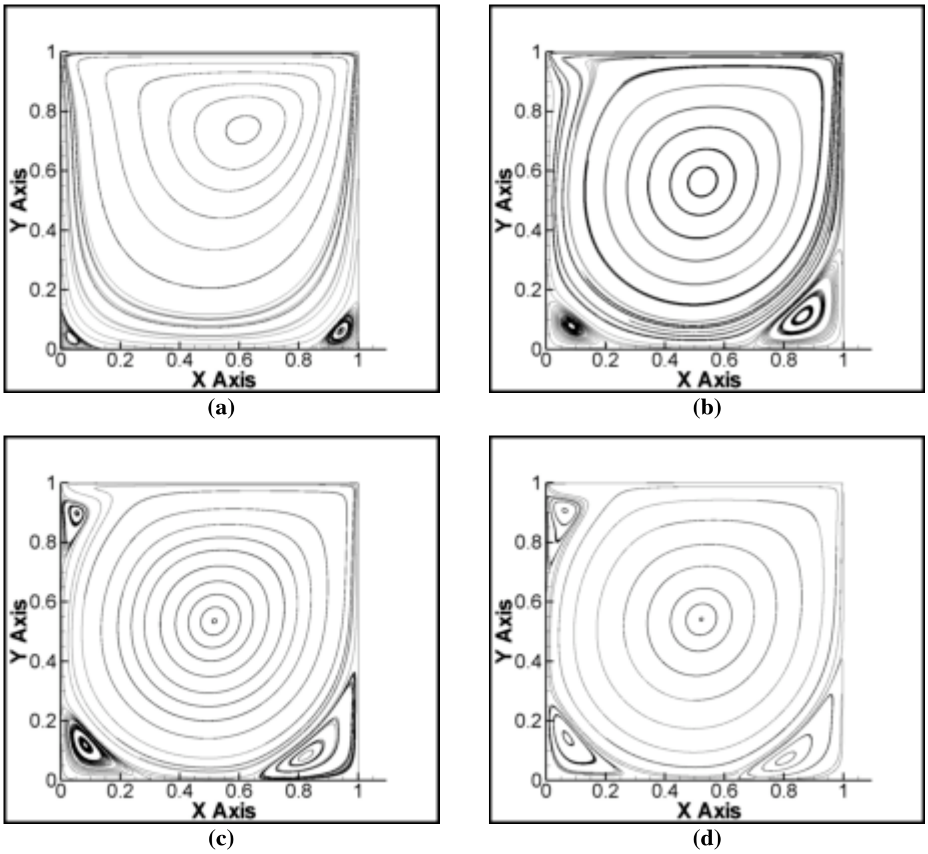
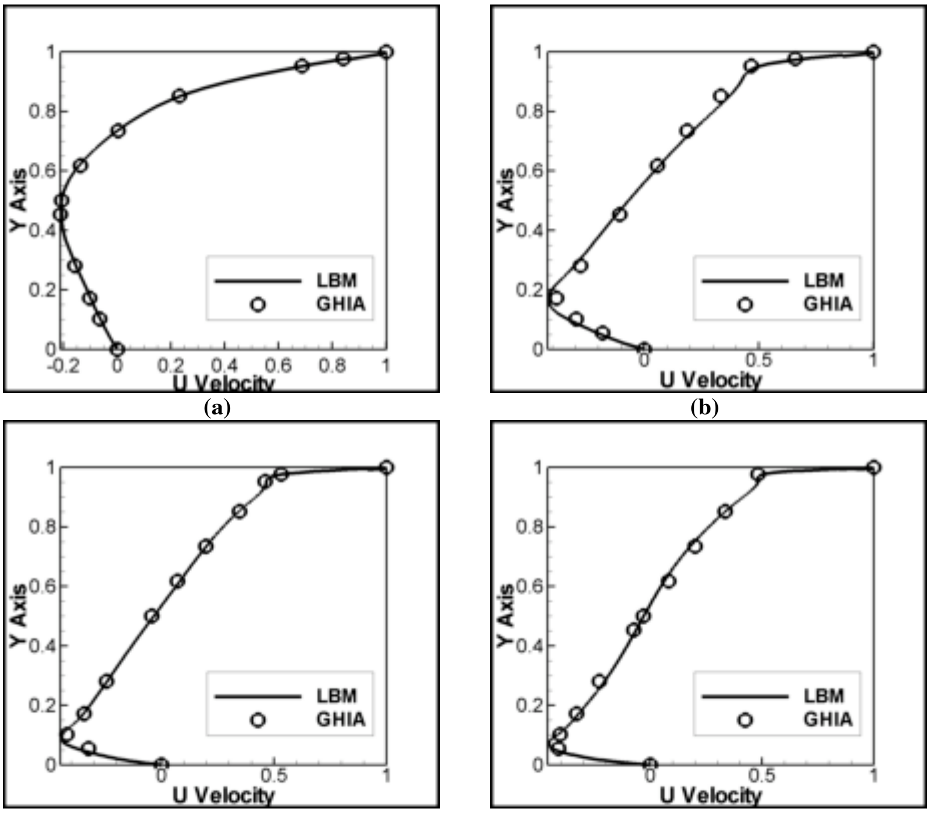


Fig. 4. Streamlines at different Reynolds numbers **(a)** $Re = 100$, **(b)** $Re = 1000$, **(c)** $Re = 3200$, **(d)** $Re = 5000$.

Using MATLAB's add-on library Jacket, LBM code is modified to be run in parallel on GPUs. Lid-driven cavity problem is solved with a number of different uniformly spaced grids using both the serial version of the code running on a single CPU (Intel Xeon E5620 Quad-Core 2.40GHz, 12MB Cache) and the parallel version running on a GPU (NVIDIA Tesla C1060, 4 GB RAM). Results are shown in Table 1 and Fig. 6. Execution times given in seconds in Table 1 are obtained by running the time marching LBM code only for a representative number of time steps that is enough to collect meaningful data. That is actual runs take longer than the amounts presented.



(c) (d)

Fig. 5. Variation of horizontal velocity component at $x = 0.5$ for different Reynolds numbers **(a)** *Re = 100*, **(b)** *Re = 1000*, **(c)** *Re = 3200*, **(d)** *Re = 5000*.

As seen in Fig. 6 for coarse grids, the code runs slower on the GPU. The reason is that the amount of floating point operations resulting from coarse grids are not enough to keep the GPU busy, but rather the memory transfer to and from the GPU becomes the bottleneck and increases the total run time. As the grid gets finer, parallelization on the GPU becomes more and more effective. For the finest mesh of 4 million nodes, which is unnecessarily big for a 2D problem, but can be appropriate for a 3D problem, a speedup of more than 17 times is obtained.

Table 1 - GPU vs CPU Time Comparison (in seconds)

Grid Size	GPU Time	CPU Time	Speedup
40 x 40	0.73	0.08	0.12
80 x 80	0.73	0.21	0.30
100 x 100	0.73	0.32	0.44
200 x 200	0.75	1.20	1.61
400 x 400	0.83	4.69	5.65
800 x 800	1.55	18.99	12.33
1200 x 1200	2.79	42.80	15.33
1600 x 1600	4.54	78.54	17.28
2000 x 2000	6.91	121.39	17.57

Fig. 6. Comparison of GPU and CPU execution time.

Current LBM implementation, based on D2Q9 model has a memory requirement of approximately 32 x *NN* floating point values, where *NN* is the total number of nodes in the mesh. For the finest mesh tried, which has $NN = 4 \times 10^6$, memory requirement is less than 1 GB for the use of single precision, which fits nicely into the 4 GB memory of the NVIDIA Tesla C1060 card used in this study. Even for D3Q15 model, which is commonly used for the solution of 3D problems, memory requirement of our LBM implementation raises only to 52 x *NN* floating point values, which is again smaller compared to the memory need of classical Navier-Stokes solvers. Even though a single card seems to be enough for LBM simulation of large 3D problems, it is logical to use multiple GPUs to increase parallelization potential of LBM and decrease run times. Jacket library has support for multiple GPUs, which will be utilized in future.

4 Conclusion

2D lid-driven cavity benchmark problem is simulated using a MATLAB implementation of Lattice Boltzmann Method. The code is parallelized to be run on GPUs by the use of the Jacket library. LBM is shown to provide accurate results for this continuum flow, without the need of any stabilization technique required for classical incompressible Navier-Stokes solvers. Parallel version of the code provided a speed up of more than 17 times for the finest mesh used with 4 million nodes with a memory usage on the GPU of less than 1GB. LBM has a very simple algorithm with enormous potential for parallelization. Also as shown in this study its memory footprint is very small. Coupled with the rapid coding and testing experience provided by MATLAB and Jacket's easy to use GPU extensions, efficient flow solvers that can simulate realistic 3D flows on personal desktop supercomputers with multiple GPUs can be developed. One disadvantage of LBM is the discretization of complicated flow fields with arbitrary boundaries, which is a current field of active research with already available promising solutions.

Acknowledgements

Financial support from METU–NCC via the Campus Research Project (BAP-EN10) and from the Turkish Scientific and Technical Research Council, Grant No. 110M750 is greatly appreciated.

References

1. Toschi F. and Succi S. (2005). Lattice Boltzmann method at finite Knudsen numbers. *Europhys. Lett* 69(4): 549-555.
2. Nie X., Doolen G. D., and Chen S. (2001). Lattice-Boltzmann simulation of fluid flows. *MEMS J. of Statistical Phys.* 107 (1-2): 279-289.
3. Ghazanfarian J., and Abbasi A. (2010). Heat transfer and fluid flow in microchannels and nanochannels at high Knudsen number using thermal Lattice-Boltzmann method. *Phys. Rev. E.* 82 (2): 026307.
4. Verhaeghe F., Luo L.-S., and Blanpain B. (2009). Lattice Boltzmann modeling of microchannel flow in slip flow regime. *Journal of Computational Physics.* 228 (1): 147-157.
5. Zhou Y., Zhang R., Staroselsky I., Chen H., Kim W. T., and Jhon M. S. (2006). Simulation of micro and nano scale flows via the Lattice Boltzmann method. 362 (1): 68-77.
6. Tubbs K. R. and Tsai F. T.-C. (2011). GPU accelerated Lattice Boltzmann model for shallow water flow and mass transport. *Int. J. for Numerical Methods in Engineering.* 86 (3): 316–334.
7. Mohammad, A. A., (2007). *Applied Lattice Boltzmann Method for Transport Phenomena, Momentum, Heat and Mass Transfer.* Calgary: Sure Printing.
8. Succi S., (2001). *Lattice Boltzmann Equation for Fluid Dynamics and Beyond*, Oxford: Clarendon Press.
9. Rothman D.H,. and Zaleski S. (1997). *Lattice Gas Cellular Automata*, Cambridge: Cambridge University Press.
10. Bhatnagar P. L., Gross E. P., and Krook, M. (1954). Gaseous flow in microconduits with viscous dissipation. *Phys. Rev.,* 94: 511-525.
11. Zou, Q., and He, X., (1997). On pressure velocity flow boundary conditions for the lattice Boltzmann BGK model. *Phys. Fluids* 9 (6): 1591-1598.
12. Ghia U., Ghia K. N., and Shin C. T. (1982). High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method. *Journal of Computational Physics* 48 (3): 387–411.

Biographies

S. Berat Çelik – was born in Turkey (1985). Graduated from mechanical engineering program of Middle East Technical University Northern Cyprus Campus (METU NCC) and currently attending graduate school in METU. Research interests are Lattice Boltzmann Methods, parallel computing and microfluidics.

Cüneyt Sert – was born in Turkey in 1974. He received B.S. and M.S degrees in mechanical engineering from Middle East Technical University (METU), Ankara, Turkey in 1996 and 1998, respectively. He received his Ph.D. degree in mechanical engineering from Texas A&M University, College Station, TX, USA.

He is currently working as a faculty at the Mechanical Engineering Department of METU. His current research interest is in the field of computational fluid dynamics by the use of finite element and spectral element formulations.

Barbaros Çetin – was born in Bursa, Turkey in 1979. He received B.S. and M.S. degree in mechanical engineering from Middle East Technical University, Ankara, Turkey in 2002 and 2005, respectively. He received his Ph.D. degree in mechanical engineering from Vanderbilt University, Nashville, TN, USA.

He is currently working as a faculty at the Mechanical Engineering Program of METU-Northern Cyprus Campus. His research interest lies in the fundamental and experimental investigations on transport phenomena in microchannel networks and development of the LOC technology for biomedical, biological and chemical applications. He is also currently working on the implementation of Lattice-Boltzmann method for fluid and heat transfer problems on graphical process units.