414

SIMULATION OF CHANNEL FLOW BY PARALLEL IMPLEMENTATION OF THERMAL LATTICE–BOLTZMANN METHOD ON GPU

S. Berat Çelik[1], Cüneyt Sert[1], Barbaros Çetin[2]*
[1]Dept. Mechanical Engineering, METU, Ankara 06531 TURKEY
[2]Microfluidics & Lab-on-a-chip Research Group, Mechanical Engineering
METU–Northern Cyprus Campus, Guzelyurt, TRNC Mersin 10, TURKEY
Correspondence author: Fax: +90-661-2999 Email: barbaros@metu.edu.tr

ABSTRACT

Lattice-Boltzmann method (LBM) is a new generation numerical technique used to solve fluid flow and heat transfer problems. It has certain advantages over the classical solvers based on Navier-Stokes equations. LBM is well suited to simulate single and multi-phase flows of single and multi-component fluids around complex geometries, and is suitable for parallel programming. One of the state-of-the-art technologies to perform parallel computing is the use of Graphics Processing Units (GPU). In this study, heat transfer analysis of single-phase laminar flow of a viscous fluid in a slit-channel with a constant wall temperature is presented using thermal Lattice–Boltzmann method. Both the flow field and the temperature field are determined by solving the associated Lattice–Boltzmann equations. Computing is performed on GPUs in parallel using JACKET which is an add-on package for MATLAB to handle GPU computing. Performance comparison with CPU-based and GPU–based computing is presented.

NOMENCLATURE

c_s	speed of sound
f	distribution function
f^{eq}	equilibrium distribution function
g	thermal distribution function
g^{eq}	thermal equilibrium distribution function
w	weight function
ω	collision frequency
Ω	collision operator
τ	relaxation time
τ_t	thermal relaxation time
ξ	lattice velocity

INTRODUCTION

The problem considered in this study is a 2D benchmark problem known as thermal slit–channel flow, which is a non–isothermal flow of an incompressible fluid between two stationary walls as shown in Fig 1. The top and the bottom walls of the channel are heated, and kept at constant temperature. Both the temperature and the flow is developing which is knows as combined entrance problem [1]. Hydrodynamically developed, thermally developing form of this problem together with some additional effects such as axial conduction and viscous dissipation has also been studied in the literature [2, 3]. In this study, this benchmark problem is solved by using Lattice Boltzmann Method. Computations are performed both on CPU (serial) and GPU (parallel). Performance of CPU and GPU computing is compared.

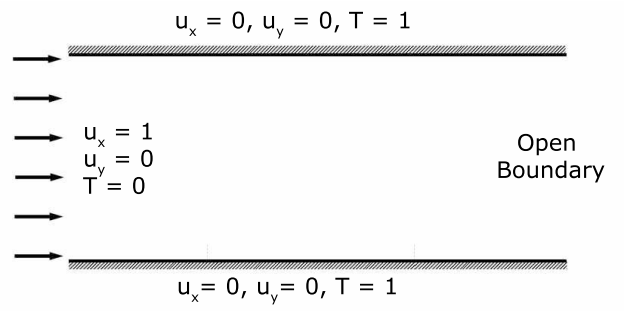

Figure 1
Computational Domain

LATTICE BOLTZMANN METHOD

For the numerical investigation of the 2D Slit-channel benchmark problem Lattice Boltzmann Method (LBM) is used. LBMs basically compose of 4 parts; collision, streaming, implementation of boundary conditions and macroscopic quantity calculations. LBM uses non-dimensional quantities for velocity, pressure, temperature etc. In this study, a popular D2Q9 model which is suitable for flow problems [4] is used (see Fig 2).

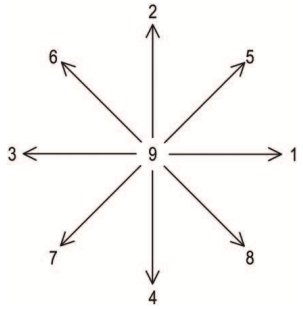

Figure 2
D2Q9 Model

Collision: The governing equation of LBM is the Boltzmann Transport Equation (BTE) [4],

$$\frac{\partial f}{\partial t} + c\nabla f = \Omega, \tag{1}$$

where f is the distribution function, Ω term is a double integral. Evaluation of BTE at every node addresses the collision part. BTE is an integro-differential equation which is hard to solve. Instead, a simplified and commonly used model is applied for the collision operator. The model is named Bhatnagar-Gross-Krook [5]. According to the model, the collision operator is replaced with the following expression,

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

where f^{eq} is the equilibrium distribution function, ω is collision frequency. Hence, the Eq. (1) becomes

$$\frac{\partial f}{\partial t} + c\nabla f = \omega(f^{eq} - f). \tag{3}$$

Eq. (3) can be discretized when first order forward Euler scheme in time and first order for-

ward upwind scheme in space are applied [6],

$$f(x+\Delta x, t+\Delta t) = f(x,t)(1-\omega) + \omega f^{eq}(x,t). \quad (4)$$

In LBM, it is common that $\Delta t = \Delta x = 1$ [4]. By using Chapman-Enskog expansion, i.e. single-relaxation time, ω can be expressed as [7],

$$\omega = \frac{1}{0.5 + \mu\left(\frac{3\Delta t}{\Delta x^2}\right)}, \quad (5)$$

where μ is kinematic viscosity. Note that, the viscosity is a function of the relaxation time, $\tau = 1/\omega$.

The equilibrium distribution function f^{eq} can be defined as [4],

$$f^{eq} = \frac{\rho}{(2\pi RT)^{D/2}} exp\left[-\frac{(\xi-u)^2}{2RT}\right] \quad (6)$$

where R, D, ξ, ρ, u and T are gas constant, dimension of the space, lattice velocity, macroscopic density of mass, velocity, and temperature, respectively. The weight functions in D2Q9 model are [8]:

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

Streaming: Streaming (updating) process delivers the recently calculated distribution functions to the neighbouring nodes. For instance, 1^{st} distribution functions of all nodes point $+x$ direction. Hence, all 1^{st} distribution functions are shifted 1 node length on the $+x$ direction. After all, 8 distribution functions undergo a streaming process according to their directions, some functions on the boundaries will be missing and the boundary conditions are needed to be assigned to find those values. Details of the streaming process can be found elsewhere [9, 4].

Implementation of Boundary Conditions: For the walls of the channel, first–order accurate bounce–back boundary condition is used. This method equalizes the unknown distribution functions to the ones leave the domain. In other words, reverses the direction of the distribution functions to obtain the unknown ones.

For the top wall boundary, for instance, the unknown functions can be determined as follows, [9]

$$f_k = \begin{cases} f_2 & \text{for } k = 4, \\ f_7 & \text{for } k = 5, \\ f_8 & \text{for } k = 6. \end{cases} \quad (8)$$

This approach provides no-slip boundary condition []. The procedure is same for the other walls.

Uniform inlet velocity can be implemented as [4],

$$f_1 = f_3 + \frac{2}{3}\rho_i u_i, \quad (9)$$

$$f_5 = f_7 - 0.5(f_2 - f_4) + \frac{1}{6}\rho_i u_i + 0.5\rho_i v_i, \quad (10)$$

$$f_8 = f_6 + 0.5(f_2 - f_4) + \frac{1}{6}\rho_i u_i - 0.5\rho_i v_i, \quad (11)$$

For the open boundary which is the outlet of the channel can be obtained by extrapolation as [4],

$$f_{3,n} = 2f_{3,n-1} - f_{3,n-2}, \quad (12)$$

$$f_{6,n} = 2f_{6,n-1} - f_{6,n-2}, \quad (13)$$

$$f_{7,n} = 2f_{7,n-1} - f_{7,n-2}. \quad (14)$$

Macroscopic Quantity Calculations: Once the distribution functions have been determined, the macroscopic quantities can be obtained. Lattice density at a given node can be determined by summing the distribution functions in each lattice direction as,

$$\rho = \sum_{k=1}^{9} f_k. \tag{15}$$

Macroscopic velocity can be calculated by taking the 1^{st} moment of the lattice density ρ with velocity u

$$\rho u = \sum_{k=1}^{9} f_k c_k. \tag{16}$$

Thermal LBM: Beside the flow of a fluid, LBM is also capable to solve temperature distribution in fluids. The procedure is to solve the fluid flow and the temperature respectively in the same time step. The advection-diffusion equation is obtained by equating the thermal distribution function, g^{eq}, to a more simplified function [4].

$$\frac{\partial g}{\partial t} + c\nabla g = \omega(g^{eq} - g) \tag{17}$$

$$g_i^{eq} = w_k \rho(x,t) \cdot \left[1 + \frac{c_k \cdot u}{c_s^2}\right] \tag{18}$$

While inserting the thermal distribution function into The Boltzmann Transport equation, the relaxation time is needed to be determined as a function of thermal diffusivity, α.

$$\tau_t = \frac{\alpha \Delta t D}{\Delta x^2} + 1/2 \tag{19}$$

Where D is the dimension of the domain. The sum of the distribution functions at a node results the non-dimensional temperature.

$$T = \sum_{k=1}^{9} g_k \tag{20}$$

Parallelization: One of the main advantages of LBM is the resulting short and clean computer code. Its algorithmic structure is also known to be very suitable for parallelization. In this study this nature of LBM is coupled with the state of the art parallel hardware, namely the Graphical Processing Unit (GPU). With the advent of the CUDA programming architecture by NVIDIA, use of GPUs for scientific computing become very popular in the last couple of years [10]. By proper utilization of 100s of parallel streaming cores and large memory bandwidth provided by todays GPUs, speed ups that are not possible to achieve using CPUs are reported in the literature [11].

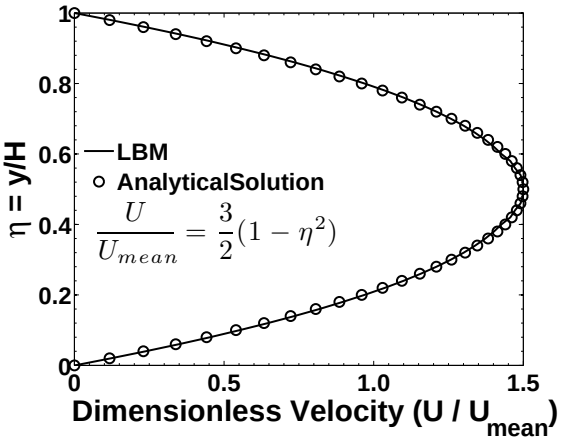

Figure 3
Fully–developed velocity profile

RESULTS AND DISCUSSION

The velocity and temperature field inside the channel is determined by solving appropriate LBM equations, Eqs (4) and (17). The resulting fully-developed velocity on a 50x200 grid

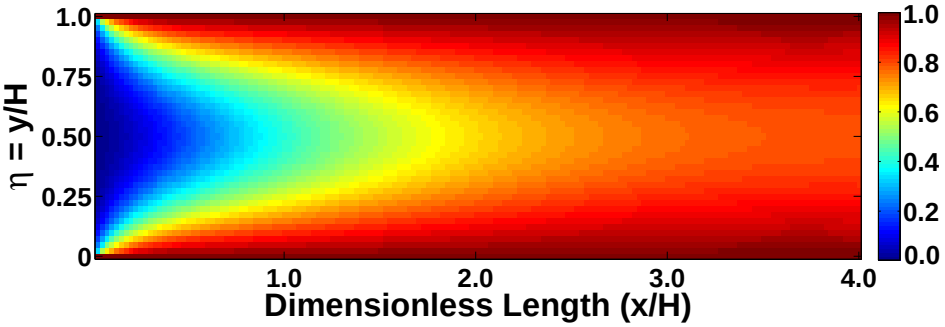

Figure 4
Temperature field

is shown in Fig 3. Analytical solution for the fully–developed velocity profile [12] is also included in the figure. As seen from the figure, perfect match is obtained between the LBM results and the analytical solution. Temperature field obtained on the same grid for a flow with $Re = 10$ is given in Fig4. As seen from the figure, the fluid is heated by the effect of the heated wall. The thermal boundary layers can be clearly seen. Heat diffuses from the wall to the fluid and convected in the longitudinal direction.

Developed LBM code is ran serial on CPU (Intel Xeon E5620 Quad-Core 2.40GHz, 12MB Cache) and parallel on GPU (Tesla C1060, 4GB RAM). Although the problem of interest is time independent, LBM results in a time marching numerical solution by its own nature. For a runtime comparison of CPU and GPU implementations of LBM it is not necessary to wait for the converged solution. Instead only a representative number of time steps are computed and the results given in Table 1 are obtained. Graphical representation of the results is provided in Figure 5. As seen, unnecessarily large node numbers for the 2D problem of interest are tried to see how the speed up scales with the size of the mesh. For node numbers less than about 26000 it is not efficient to work on a GPU. The reason is the overhead of memory transfers between the CPU and GPU being larger than the gain obtained by parallelization. However, increasing the node number increases the time spent by CPU dramatically whereas the time spent by GPU is comparatively stable.

For meshes of 500x2000 (1 million) nodes or finer, which are actually too large for this 2D problem, but could be used for a 3D one, it is possible to obtain more than 10 fold increase. The finest mesh tried has 4 million nodes and provides a speed up of almost 14 times. It is clearly seen from Table 1 that as the problem size increases the speed up also increases, which makes GPU computing attractive especially for 3D problems.

CONCLUSIONS

In this study, 2D thermal slit-channel flow is simulated using LBM. It is shown that LBM has a very high parallelization potential with

Table 1
Comparison of GPU and CPU execution time*

Grid Size	GPU time	CPU time	Speedup
20x80	2.22	0.45	0.20
40x160	2.24	1.03	0.46
60x240	2.24	1.60	0.71
80x320	2.26	2.23	0.99
100x400	2.24	3.16	1.41
120x480	2.26	4.31	1.90
140x560	2.28	5.59	2.44
160x640	2.35	7.13	3.03
180x720	2.43	8.89	3.66
200x800	2.53	10.8	4.28
300x1200	3.26	23.7	7.26
400x1600	4.26	40.4	9.49
500x2000	5.63	62.5	11.1
600x2400	7.44	90.0	12.1
700x2800	9.63	122	12.7
800x3200	12.2	160	13.2
900x3600	15.0	202	13.5
1000x4000	18.2	251	13.8

* in seconds

the use of GPUs. For grids suitable for large 3D problems, GPU computing provided up to 14 times of speed up compared to serial version of the code ran on CPU. However, for node numbers less than 26000 parallel version of the code performs slower than the CPU counterpart. It is also experienced that the use of MATLAB and JACKET results in tremendous savings in code development time. JACKET library provides a very convenient way to parallelize a serial MATLAB code for GPU computing with very minimal source code changes. Even though coding in a compiled language such as Fortran or C++ may result in a different and possibly better speed up performance, the mentioned advantages of MATLAB and JACKET makes this combination a valuable alternative.

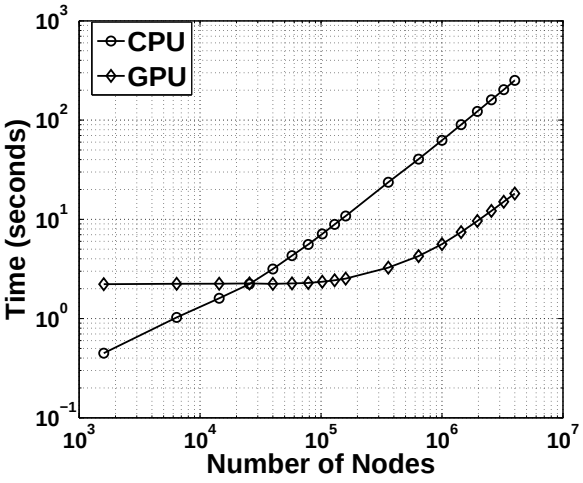

Figure 5
Comparison of GPU and CPU execution time

ACKNOWLEDGMENTS

Financial support from the METU–NCC via the Campus Research Project (BAP-FEN10) and from the Turkish Scientific and Technical Research Council, Grant No. 110M750 is greatly appreciated.

REFERENCES

1. Incorpera, F. P., and DeWitt, D. P., 1996. *Fundamentals of Heat and Mass Transfer*, 4 ed. John Wiley & Sons, New York, pp. 439–454.

2. Çetin, B., Yazicioglu, A., and Kakac, S., 2006. Gaseous flow in microconduits with viscous dissipation. *Int J Transport Phenomena,* **8**(4), pp. 297–315.

3. Jeong, H. E., and Jeong, J. T., 2006. Extended Graetz problem including streamwise conduction and viscous dissipation in

microchannels. *Int. J. Heat Mass Transfer,* **49**, pp. 2151–2157.

4. Mohammad, A. A., 2007. *Applied Lattice Boltzmann Method for Transport Phenomena, Momentum, Heat and Mass Transfer*. Sure Printing, Calgary, pp. 21, 86, 110, 114, 115, 143.

5. P. L. Bhatnagar, E. P. G., and Krook, M., 1954. Gaseous flow in microconduits with viscous dissipation. *Phys. Rev.,* **94**(511).

6. Yong Shi, T. S. Z., and Guo, Z. L., 2004. Thermal lattice bhatnagar-gross-krook model for flows with viscous heat dissipation in the incompressible limit. *Phys. Rev. E,* **70**, p. 066310.

7. Chapman, S., and Cowling, T. G., 1990. *The Mathematical Theory of Non-Uniform Gases: An Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases*. Cambridge University Press, U.K.

8. Yong Shi, T. S. Z., and Guo, Z. L., 2008. Simplified model and lattice boltzmann algorithm for microscale electro-osmotic flows and heat transfer. *Int. Journal of Heat and Mass Transfer,* **51**, pp. 586–596.

9. Succi, S., 2001. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*. Oxford University Press, New York, p. 84.

10. Sanders, J., and Kandrot, E., 2010. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Edwards Brothers in Ann Arbor, U.S.A.

11. AccelerEyes, 2011. *Getting Started Guide Jacket v1.7*. AccelerEyes.

12. Cetin, B., 2005. Analysis of single phase convective heat transfer in microtubes and microchannels. Master's thesis, Middle East Technical University.