

# Development of a Lattice Boltzmann Model for the Solution of Partial Differential Equations, A Performance Comparison Study with that of the Finite Difference Method



Mahmud Ashrafizaadeh and A. Ghavaminia

**Abstract** The lattice Boltzmann method (LBM) has attracted much attention in recent years as a recent efficient solution method for fluid flow simulations as well as general PDEs. Due to the local nature of the computations in the lattice Boltzmann method and its ease of programming, the LBM is an ideal candidate for developing efficient parallel PDE solvers suitable for recent computer hardware. In the present study, we have used the lattice Boltzmann method for solving the transient heat diffusion equation. The performance of this method is compared with that of the traditional finite difference based PDE solver. All these solvers have been developed using the Julia programming language, which is a recent player amongst the scientific computing languages. Several benchmark problems in the field of transient heat transfer described by parabolic PDEs are solved, and the results obtained from the aforementioned methods are compared with each other. It is shown that by using the lattice Boltzmann method, it is possible to solve these partial differential equations more efficiently while maintaining the accuracy of the solution.

**Keywords** Lattice Boltzmann method · Finite difference · Partial differential equation · Numerical performance comparison

## 1 Introduction

The lattice Boltzmann method (LBM) is a rather young and promising method for simulating complex fluid flow physics. In comparison with the conventional methods in computational fluid dynamics (CFD), LBM is easy for programming, intrinsically

---

M. Ashrafizaadeh (✉) · A. Ghavaminia  
Department of Mechanical Engineering, Isfahan University of Technology,  
Isfahan 841568311, Iran  
e-mail: [mahmud@cc.iut.ac.ir](mailto:mahmud@cc.iut.ac.ir)

A. Ghavaminia  
e-mail: [a.ghavami@alumni.iut.ac.ir](mailto:a.ghavami@alumni.iut.ac.ir)

© Springer Nature Switzerland AG 2021

D. M. Kilgour et al. (eds.), *Recent Developments in Mathematical, Statistical and Computational Sciences*, Springer Proceedings in Mathematics & Statistics 343, [https://doi.org/10.1007/978-3-030-63591-6\\_24](https://doi.org/10.1007/978-3-030-63591-6_24)

255

parallel [1, 19], and it is easy to incorporate complicated boundary conditions such as those encountered in porous media problems [6, 10].

In the past years, in addition to solving the fluid flow problems, more applications have been introduced for the lattice Boltzmann method. It has been shown that by modifying a typical LBM, it is possible to solve partial differential equations as well [3, 9, 14, 18, 20]. Solving partial differential equations is required in a vast verity of applications varying from image denoising using the Laplace equation [15] to simulating the heat transfer phenomena in solid or fluid media.

With advances in the high-performance supercomputers and the invention of new GPU acceleration methods, the LBM has gained even more attention due to its intrinsic parallelism and locality of calculations, which is an advantage of the LBM over the conventional partial differential equation solvers [1, 19].

In this study, the accuracy and computational performance of a LBM PDE solver have been investigated and compared with those of a traditional FD PDE solver by solving several benchmarks in the field of transient heat transfer which are described by parabolic PDEs.

## 2 Description of the Numerical Methods

The transient heat diffusion equation can be written as:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad (1)$$

In which  $T$  represents temperature,  $\alpha$  is the heat diffusion coefficient,  $t$  is time, and  $x$  is the spatial direction of the diffusion [8]. One of the well known traditional FD solutions for this PDE is to use the Forward Euler method and the second-order central difference scheme to discretize the Eq. (1) as it is presented in Eq. (2).

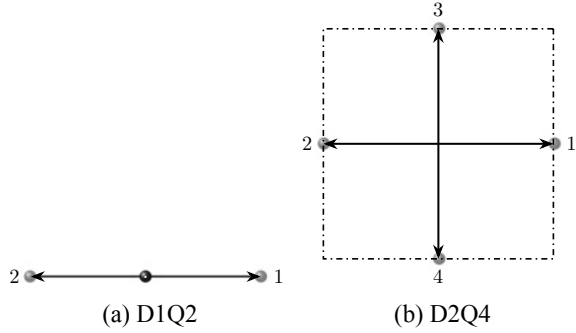
$$T_i^{n+1} = T_i^n (1 - \omega) + \omega(0.5T_{i+1}^n + 0.5T_{i-1}^n) \quad (2)$$

where  $\omega = (2\alpha \Delta t)/(\Delta x^2)$ . It is necessary to mention that this discrete formulation has a stability criterion ( $\Delta t \leq \Delta x^2/2\alpha$ ), which limits the maximum possible time step size for a fixed  $\Delta x$ . The stability criterion may also vary according to the employed boundary condition type [8]. This method only requires information from the nearest neighboring sites at each time step. Therefore, it is very similar to the LBM in this regard.

The lattice Boltzmann general formulation is:

$$\frac{\partial f}{\partial t} + c \cdot \nabla f = \Omega \quad (3)$$

**Fig. 1** Lattice configuration and naming convention used for discretization of the LBM



where  $f$  is the population distribution function (PDF),  $c$  is the lattice speed,  $\nabla$  is the gradient operator, and the  $\Omega$  is the collision operator [13]. The  $\Omega$ , in general, is a sophisticated integral, which is hard to compute. There have been several approximations for this integral which up to now, the most widely used and known of them is the Bhatnagar-Gross-Krook (BGK) model [11]. Among the different LBM methods, the BGK collision method [16] is mathematically the simplest. However, it has some deficiencies; for example, the BGK LBM suffers from numerical instability, especially for the simulation of low viscosity fluids [12]. To overcome these deficiencies, several advanced methods have been proposed, among which the multi-relaxation LBM [4] and cascaded central moments method [5] are two of the most known ones.

For discretizing the LBM formulation, Eq. (1), over a domain, there is a need to choose a lattice configuration. In this work, we have chosen the D1Q2 and D2Q4 lattice configuration for one-dimensional and two-dimensional simulations as they are shown in Fig. 1a, b, respectively.

By discretizing the LBM formulation, we will have:

$$\frac{f_k(x, t + \Delta t) - f_k(x, t)}{\Delta t} + c_k \cdot \frac{f_k(x + \Delta x, t + \Delta t) - f_k(x, t + \Delta t)}{\Delta x} = \Omega_k \quad (4)$$

The BGK approximation is:

$$\Omega_k = \frac{1}{\tau} [f_k(x, t) - f_k^{eq}(x, t)] \quad (5)$$

By substitution of the BGK approximation, Eq. (5), into Eq. (4) the BGK LBM becomes:

$$f_k(x + \Delta x, t + \Delta t) - f_k(x, t) = -\frac{\Delta t}{\tau} [f_k(x, t) - f_k^{eq}(x, t)] \quad (6)$$

where  $f^{eq}$  is the equilibrium distribution function and  $\tau$  is the relaxation time factor. By performing the Chapman-Enskog expansion [17] the relation between macroscopic values and the  $f^{eq}$  and  $\tau$  can be determined as:

$$f_k^{eq} = w_i T(x, t) \quad (7)$$

where  $w_i$  is the lattice weight factor. For the D1Q2 lattice configuration  $w_i = 0.5$ ,  $i = 1, 2$  and for D2Q4 lattice configuration  $w_i = 0.25$ ,  $i = 1, \dots, 4$  [13]. Additionally, the temperature  $T$  is calculated using the Eq. (8) [13].

$$T(x, t) = \sum_{i=1}^q f_i(x, t) \quad (8)$$

It is more convenient to separate the local and nonlocal part of Eq. (6) and perform the calculations in the collision, Eq. (9), and the streaming, Eq. (10), processes separately.

$$f_k^{post} = f_k(x, t) - \frac{\Delta t}{\tau} [f_k(x, t) - f_k^{eq}(x, t)] \quad (9)$$

$$f_k(x + \Delta x, t + \Delta t) = f_k^{post}(x, t) \quad (10)$$

Boundary Cconditions:

To implement the Dirichlet boundary condition, we should write the flux balance at the considered boundary. For example, at the left boundary, we have:

$$f_1^{eq}(x, t) - f_1(x, t) + f_2^{eq}(x, t) - f_2(x, t) = 0 \quad (11)$$

In Eq. (11) the only unknown is the  $f_1$ .

In the case of the Neumann boundary condition on the left side:

$$q' = -k \frac{T(1) - T(0)}{dx} \quad (12)$$

Substituting  $T(1) = f_1(1) + f_2(1)$  and  $T(0) = f_1(0) + f_2(0)$  in Eq. (12) and solving for  $f_1(0)$  gives:

$$f_1(0) = f_1(1) + f_2(1) - f_2(0) + \frac{q' dx}{k} \quad (13)$$

Finally, in the case of the Robin boundary condition, we have:

$$-\lambda \frac{\partial T(x, t)}{\partial x} = \beta [T(x, t) - T_a] \quad (14)$$

where  $\lambda$  is the solid medium thermal conductivity,  $\beta$  is the convection coefficient in lattice units, and  $T_a$  is the ambient temperature [13]. Expanding the Eq. (14) for the right boundary condition yields to:

$$-\lambda \frac{T_n^{p+1} - T_{n-1}^{p+1}}{\Delta x} = \beta [T_n^{p+1} - T_a] \quad (15)$$

This means:

$$f_{1,n}^{post} + f_{2,n}^{post} = \frac{\lambda}{\lambda + \beta \Delta x} (f_{1,n-1}^{post} + f_{2,n-1}^{post}) + \frac{\beta \Delta x}{\lambda + \beta \Delta x} T_a \quad (16)$$

On the right side boundary, the only unknown distribution function is  $f_{2,n}$ , which can be calculated using Eq. (16).

For a more detailed derivation of boundary conditions, interested readers may refer to references [7, 9, 13, 20].

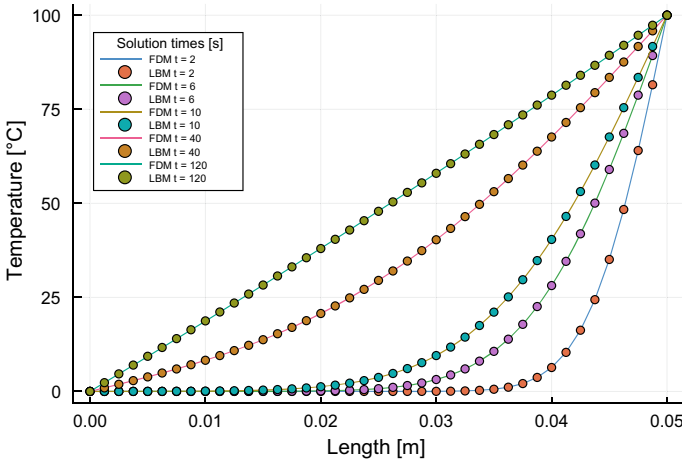
### 3 Results

The results presented here are obtained using codes which have been implemented in the Julia programming language [2]. The simulations are performed on a computer system with the following configuration: CPU model: AMD Opteron(tm) Processor 6174, CPU MHz: 2200, L1 cache: 64K, L2 cache: 512K, L3 cache: 5118K, and 94 GB of RAM.

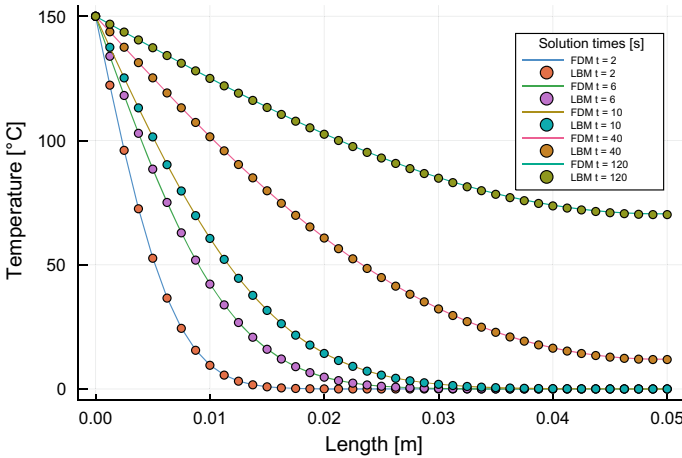
To compare the performance and accuracy of the methods, we used several common one-dimensional and two-dimensional benchmarks introduced in [7, 9, 13]. In the first case a one-dimensional layer of steel with a thermal conductivity of,  $\lambda$ , 35 [W/mK], thermal diffusivity of,  $\alpha$ ,  $7.1795 \times 10^{-6}$  [m<sup>2</sup>/s], and thickness of  $L = 0.05$  [m] has been considered. The initial condition, the left boundary condition, and the right boundary condition are specified by  $T(x, 0) = 0$  [°C],  $T(t, 0) = 0$  [°C], and  $T(t, L) = 100$  [°C] respectively. To obtain the results illustrated in Fig. 2, the mesh size and time step have been set to  $\Delta x = 0.00125$  [m] (*node numbers* ( $N$ ) = 40) and  $\Delta t = 0.001$  [s] respectively. Figure 3 shows the same geometry as the first case (Fig. 2) with the exception of the Robin boundary condition ( $\beta = 10$  [W/m<sup>2</sup>K],  $T_a = 20$  [°C]) at the right hand side of the geometry and  $T = 150$  [°C] at the left-hand side of the geometry. Figures 2 and 3 clearly show that the results obtained from the LBM and the FD codes are in excellent agreement with each other.

For the two-dimensional formulation, two geometries are selected to compare the performance and the accuracy of the lattice Boltzmann method versus that of the finite difference method. Fig. 4a, b show the geometries, boundary conditions, and initial conditions of the chosen benchmarks.

Figure 5 shows the temperature distribution along the centerline of the case 3 geometry for different dimensionless times. Fig. 6, which is a contour display of the temperature distribution of case 4, shows that the results from both the finite difference method and lattice Boltzmann method are in an excellent agreement with each other. To demonstrate this better, Fig. 7 shows the temperature distributions along the X and Y oriented centerlines of the case 4 geometry. As it is shown in Fig.



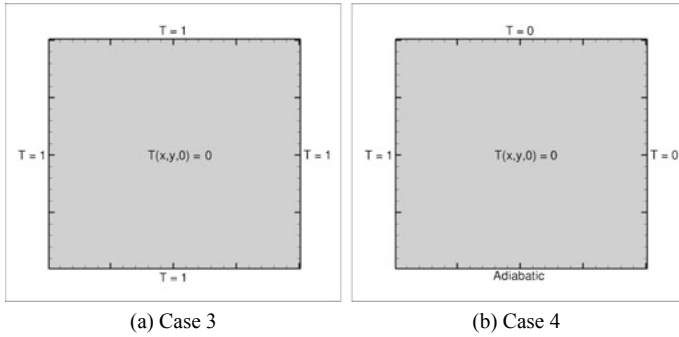
**Fig. 2** Temperature distribution at different times for the one-dimensional transient heat transfer with Dirichlet boundary conditions at the left and right sides



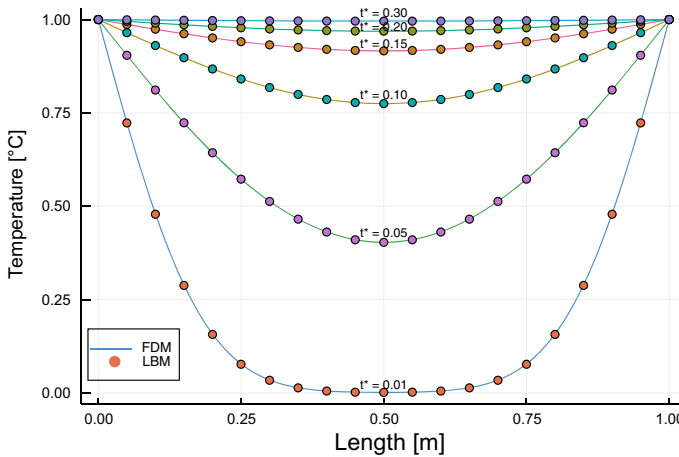
**Fig. 3** Temperature distribution at different times for the one-dimensional transient heat transfer with a Dirichlet boundary condition at the left side and a Robin boundary condition at the right side

6 to Fig. 7, the results from the lattice Boltzmann method and the finite difference method are in excellent agreement with each other.

The primary motivation of using the LBM over conventional methods such as the FD for solving differential equations (in this case, the heat diffusion problem) is to benefit from the LBM's capabilities, particularly the computational performance of the LBM. Another test has been conducted to study the performance of the LBM and the FD, which compares wall clock times required by each method to achieve



**Fig. 4** Boundary conditions and initial condition values for the two-dimensional transient heat transfer case 3 and case 4



**Fig. 5** Temperature distribution along the centerline of the case 3 geometry (Fig. 4a) at different dimensionless times ( $t^*$ ). Circles and lines represent the LBM and the FD results respectively

a particular result. For this test, we used case 3 geometry to benchmark the results. Times are measured with the benchmarking tool provided by the Julia language (*BenchmarkTools.jl*), which is a tool that can repeat each benchmark and processes the result to eliminate system load noises and to provide a consistent, reliable answer. To conduct a fair comparison, the FD time step has been pushed to its maximum possible value concerning the FD stability limits ( $Fo = 0.25$ ). Then the LBM relaxation time factor has been changed to measure the changes in the performance (Fig. 8).

As it is shown in Fig. 9, the resolution study reveals that even in lower mesh resolutions, the solution is mesh-independent. Nevertheless, we have chosen a finer mesh because we wanted to ensure the accuracy of the study through different simulations,

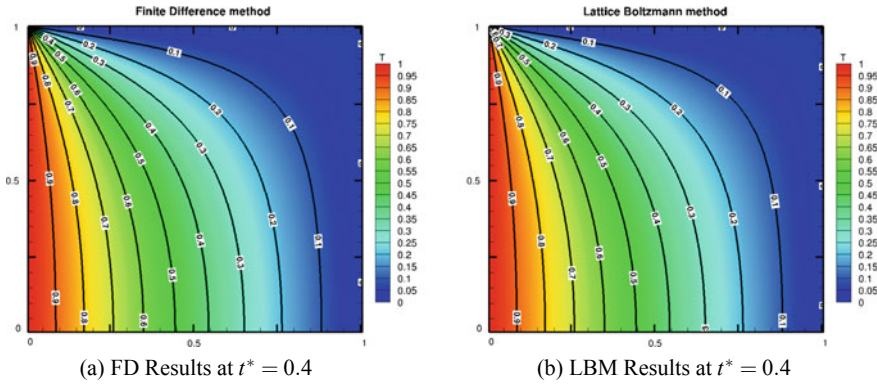


Fig. 6 Case 4 results: Temperature distribution over the solution domain at a dimensionless time ( $t^* = 0.4$ )

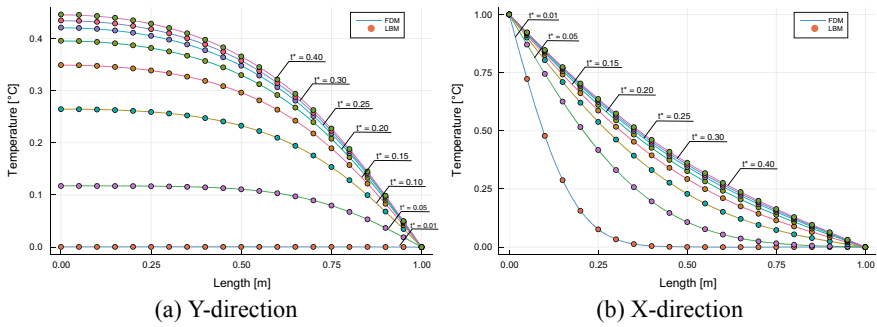


Fig. 7 Temperature distribution along the X and Y oriented centerlines at different dimensionless times

and more importantly, we needed a more time-consuming calculation so that we can provide a more accurate time comparison.

Figure 8 depicts the time ratio of the FD over the LBM calculations, which clearly shows the LBM's superiority over the FD regarding their performances. It is worth mentioning that in this study, we have just implemented a serial code for both the LBM and the FD methods. Due to the parallel nature of the LBM, it is expected that the parallel implementation of these codes would result in even more performance gain in favor of the LBM approach. It is known that by increasing the relaxation time factor of the BGK LBM, its numerical accuracy may deteriorate [12]. To measure this error, the difference between the LBM solution and that of the FD solution for a variety of dimensionless times and LBM relaxation factors are calculated. The results are presented in Fig. 10. The maximum measured difference in our studies has been less than 0.6%, which could be neglected in most common applications. On the other hand, the performance gain is very high. Nearly 0.1 of the computational time is required for the LBM to achieve the same results as that of the FD method. Also, the difference between the LBM and the FD results vanishes by advancing



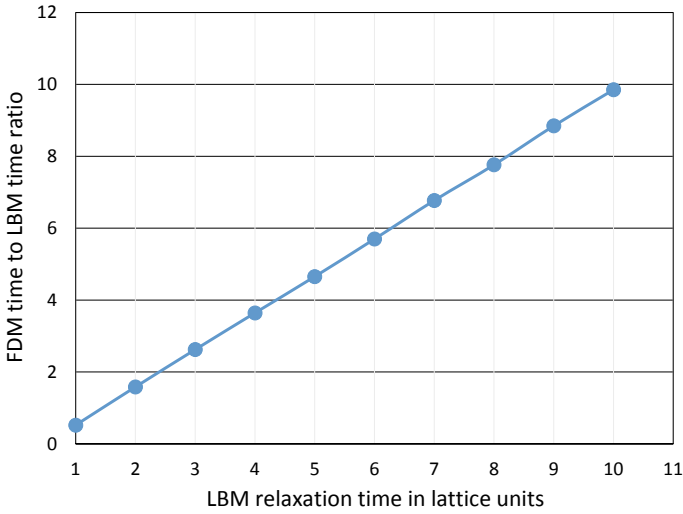


Fig. 8 The time ratio of the FD over the LBM solution.  $r = \frac{Time_{FD}}{Time_{LBM}}$

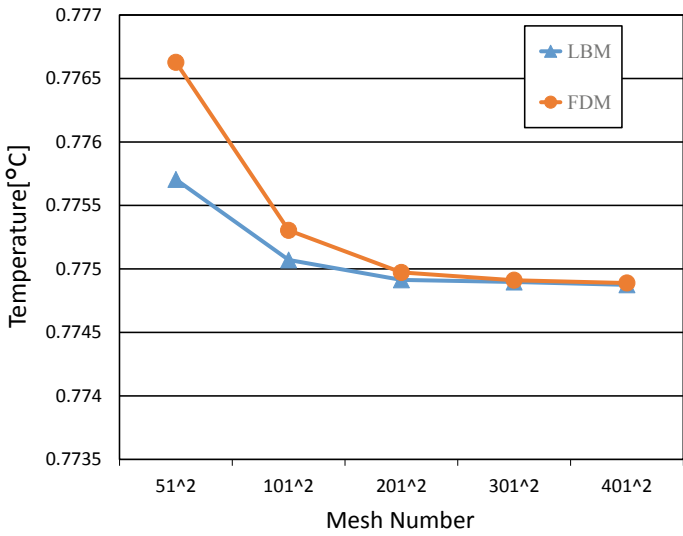
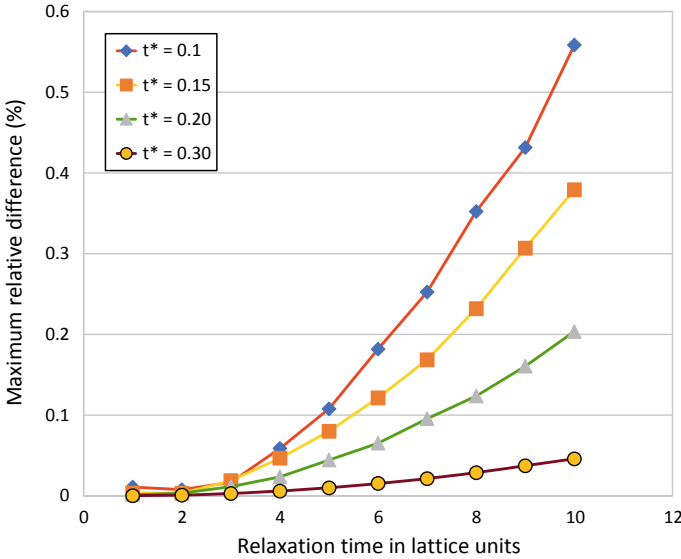


Fig. 9 Mesh study with case 3, LBM relaxation fact 2 and dimensionless time 0.1. The graph shows changes in the temperature values in the center of the geometry for LBM and FD method



**Fig. 10** Maximum relative difference between the FD and LBM solutions through the centerline of the geometry for different LBM relaxation times at different dimensionless times  $difference = \max(\frac{T(x,y,t)_{LBM} - T(x,y,t)_{FDM}}{T(x,y,t)_{FDM}}) \times 100$

through time. So it is possible to even achieve the same steady-state results while spending significantly less computational time using the LBM.

### 4 Conclusion

In this study, the lattice Boltzmann method has been employed to solve the transient heat diffusion equation with different boundary conditions in one and two dimensions. To study the accuracy and performance of the lattice Boltzmann method, several benchmarks have been implemented in the Julia programming language. The results show that the lattice Boltzmann method solution for the transient heat diffusion problems would be as accurate as those obtained by the finite difference method. However, using the lattice Boltzmann method, it is possible to achieve better computational performance and significantly reduce the computational cost. In our benchmarks, the required solution time for the lattice Boltzmann method is an order of magnitude less than that required by the finite difference method for the same level of accuracy.

**Acknowledgements** The authors gratefully acknowledge the Sheikh Bahaei National High Performance Computing Center (SBNHPCC) for providing computing facilities and time. SBNHPCC

is supported by scientific and technological department of presidential office and Isfahan University of Technology (IUT).

## References

1. Ashrafizaadeh, M., Zadehghol, A., Safi, A.: GPU implementation of a lattice boltzmann flow solver. In: 18th Annual International Conference on Mechanical Engineering-ISME2010, number May, Tehran (2010)
2. Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: a fresh approach to numerical computing. *SIAM Rev.* **59**(1), 65–98 (2017)
3. Chai, Z., He, N., Guo, Z., Shi, B.: Lattice Boltzmann model for high-order nonlinear partial differential equations. *Phys. Rev. E* **97**(1), 013304 (2018)
4. D’Humières, D.: Multiple-relaxation-time lattice Boltzmann models in three dimensions. *Philos. Trans. R. Soc. Lond. Ser. A Mathe. Phys. Eng. Sci.* **360**(1792), 437–451 (2002)
5. Geier, M., Greiner, A., Korvink, J.G.: Cascaded digital lattice Boltzmann automata for high Reynolds number flow. *Phys. Rev. E* **73**(6), 066705 (2006)
6. Gharibi, F., Jafari, S., Rahnama, M., Khalili, B., Jahanshahi Javaran, E.: Simulation of flow in granular porous media using combined Lattice Boltzmann method and smoothed profile method. *Comput. Fluids* **177**, 1–11 (2018)
7. Guo, Z., Shu, C.: Lattice Boltzmann method and its applications in engineering. In: *Advances in Computational Fluid Dynamics*, vol. 3. World Scientific (2013)
8. Incropera, F.P., DeWitt, D.P., Bergman, T.L., Lavine, A.S.: *Fundamentals of Heat and Mass Transfer*.pdf, 7 edn. Wiley (2002)
9. Kałuza, G.: The numerical solution of richards equation using the Lattice Boltzmann method. *Appl. Mech. Mater.* **11**(1), 23–30 (2012)
10. Khalili, B., Rahnama, M., Jafari, S., Gharibi, F., Jahanshahi Javaran, E.: Lattice Boltzmann simulation of solid particles motion in a three dimensional flow using smoothed profile method. *J. Appl. Fluid Mech.* **10**(4), 1091–1103 (2017)
11. Krüger, T., Kusumaatmaja, H., Kuzmin, A., Shardt, O., Silva, G., Viggien, E.M.: *The Lattice Boltzmann Method*. Graduate Texts in Physics. Springer International Publishing, Cham (2017)
12. Luo, L.-S., Liao, W., Chen, X., Peng, Y., Zhang, W.: Numerics of the lattice Boltzmann method: effects of collision models on the lattice Boltzmann simulations. *Phys. Rev. E* **83**(5), 056710 (2011)
13. Mohamad, A.A.: *Lattice Boltzmann Method*. Springer, London (2011)
14. Otomo, H., Boghosian, B.M., Dubois, F.: Two complementary lattice-Boltzmann-based analyses for nonlinear systems. *Physica A Stat. Mechan. Appl.* **486**, 1000–1011 (2017)
15. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. In: *ACM SIGGRAPH 2003 Papers on—SIGGRAPH ’03*, p. 313. ACM Press, New York, NY, USA (2003)
16. Qian, Y.H., D’Humières, D., Lallemand, P., Search, H., Journals, C., Contact, A., Iopscience, M., Address, I.P.: Lattice BGK models for Navier-Stokes equation. *Europhys. Lett. (EPL)* **17**(6), 479–484 (1992)
17. Rosenboum, E.J.: The mathematical theory of non-uniform gases (Chapman, S.; Cowling, T. G.). *J. Chem. Educ.* **18**(1), 48 (1941)
18. Zergani, S., Aziz, Z.A., Viswanathan, K.K.: Exact solutions and lattice Boltzmann method modelling for shallow water equations. *Glob. J. Pure Appl. Math.* **12**(3), 2243–2266 (2016)
19. Zhao, Y.: Lattice Boltzmann based PDE solver on the GPU. *Vis. Comput.* **24**(5), 323–333 (2008)
20. Zhou, Z., Ma, J.: Lattice Boltzmann methods for solving partial differential equations of exotic option pricing. *Front. Math. China* **11**(1), 237–254 (2016)