EAI Endorsed Transactions
on Pervasive Health and Technology
Research Article EAI.EU

Parallelization of the numerical simulation of motion of deformable objects within fluid domain on a GPU device

T. Djukic[1,*] and N. Filipovic[1,2]

[1]BioIRC R&D Bioengineering Center, Prvoslava Stojanovica 6, 34000, Kragujevac, Serbia
[2]Faculty of Engineering, University of Kragujevac, Sestre Janjic 6, Kragujevac, 34000, Serbia

Abstract

Computationally demanding numerical simulations can be significantly accelerated using GPU (Graphics Processing Unit) devices. This way, the results of the simulation can be observed in real time. In this paper, the principles of GPU programming are used to simulate the movement of deformable objects within fluid domain. Lattice Boltzmann (LB) method is used to simulate fluid flow. The solid-fluid interaction is modeled using the Immersed boundary method. The developed software was tested on a Tesla GPU device; the execution time of parallelized version and sequential version of the software are compared and significant speed-up is obtained. Fluid flow simulations in the field of biomedicine that needed up to several hours to be performed, can now be completed in just a few minutes.

Keywords: CUDA architecture, NVIDIA, lattice Boltzmann method, solid-fluid interaction, parallelization speed-up.

Received on 13 July 2017, accepted on 06 October 2017, published on 28 February 2018

doi: 10.4108/eai.28-2-2018.154143

1. Introduction

Modern computer scientific simulations of diverse problems, such as fluid flow simulations, can be extremely complex and require large computational resources. GPU devices (Graphics Processing Unit) represent a good choice for this type of applications. GPU devices have already been successfully applied in several existing programs, such as in processing and analysis of medical images [1], molecular dynamics [2], DNA sequence alignment [3], bioinformatics pairwise sequence alignment [4], block decompositioning [5], graph component labelling [6], computation of shortest paths [7], etc. GPU devices execute hundreds or even thousands of threads simultaneously and thus accelerate calculations. One of the greatest benefits is the fact that only small modifications have to be introduced in the existing computer programs, thanking to a specially designed CUDA architecture (Compute Unified Device Architecture). In this paper, the basic principles of GPU programming are applied to the numerical simulation of motion of deformable objects within fluid domain.

The paper is organized as follows. In Section 2 the numerical model is explained in detail. Execution time of sequential and parallelized version was compared for a specific test simulation. The details of this test simulation are given in Section 3. Section 4 shows the results and the obtained speed-up. Section 5 concludes the paper.

2. Description of the Numerical Model

Numerical model consists of three main parts: fluid flow simulation, modeling deformation of the solid and solid-fluid interaction.

Numerical simulation of fluid flow can be performed using many different methods, such as a continuum based finite element method [8] or discrete methods like dissipative particle dynamics [9]. In this paper, a discrete method called lattice Boltzmann (LB) is used [10, 11].

*Corresponding author. Email:tijana@kg.ac.rs

This method observes fluid as a set of fictitious particles and by studying the dynamics of motion of these particles (the collisions between them and further propagation), the fluid flow is modelled on the macroscopic level. A distribution function has an identical form for all the particles and its value for each particle depends on the state of neighboring particles. The entire fluid domain is represented by a large number of identical quadrilateral elements, and this representation is called a lattice mesh.

In the numerical simulation, the basic equation of LB method is solved in two in two steps that represent the two characteristic phases of motion of particles – collision and propagation. These two equations/steps can be written as:

Collision step:

$$\bar{f}_i^{out}(\mathbf{x},t) = \bar{f}_i^{in}(\mathbf{x},t) - \frac{1}{\bar{\tau}}\left(\bar{f}_i^{in}(\mathbf{x},t) - f_i^{(0)}(\rho,\mathbf{u})\right) + \left(1 - \frac{1}{2\bar{\tau}}\right)F_i \,. \quad (1)$$

Propagation step:

$$\bar{f}_i^{in}\left(\mathbf{x} + \bar{\xi}_i, t+1\right) = \bar{f}_i^{out}(\mathbf{x},t) \,. \quad (2)$$

These two steps are repeated the predefined number of times (iterations). Components of the distribution function are hence recalculated for all nodes of the lattice mesh in each of these interations.. In the above equation, $\bar{f}_i^{in}$ and $\bar{f}_i^{out}$ denote values of the discretized distribution function before and after collision, $\bar{\tau}$ denote the relaxation time, F_i denotes the external force term and with $\bar{\xi}_i$ the vectors defining the abscissae of the lattice structure are represented. In this paper, the three-dimensional isothermal flow of incompressible fluid is considered and the abscissae for lattice structure denoted by D3Q27 are used.

In this paper, the deformable object is observed as a part of the fluid that is separated from the surrounding fluid by a membrane whose thinkness can be considered negligible. The discretization of the membrane is perfomed by dividing the entire membrane into a predefined number of triangles. It is considered that the deformation of the immersed object is influenced by four different parameters: volume within membrane, surface area of the membrane, surface strain of the membrane and bending of the membrane.

The surrounding fluid causes the deformation of the object and this deformation causes a reaction force in all nodes of the membrane. The reaction forces caused by the change of all four mentioned parameters are calculated using the equations that are presented in literature by Dupin et al. [12]. Calculated reaction forces are transferred to the fluid points and fluid then again causes new deformation of the object. This process is repeated in iterations.

The solid-fluid interaction is modeled using the strong coupling approach, more precisely, using the immersed boundary method [13]. The main idea is to transfer the reaction forces to the fluid domain using an external force field, while the new deformation of the object is calculated by interpolating the velocities of the boundary points. Interpolation over the neighboring points is used to calculate all quantities that are necessary for the simulation of solid-fluid interaction. This means that for each boundary point of the membrane of the deformable object, the influence of several points from the fluid lattice mesh is considered and vice versa. Interpolation is performed using the Dirac delta function, as it is defined in literature [13]. Additional details of IBM are given in literature [11,14,15].

3. Test Simulation Setup

During the testing, the simulation of a particle deformation under the influence of shear flow is executed. In this simulation, the fluid domain is modeled such that the velocities of upper and lower walls are prescribed and these velocities have opposite directions, like it is illustrated in Fig. 1. Dimensions of the domain in all three directions are equal – $L_x = L_y = L_z$. On the boundaries of the domain along x and z axis the periodical boundary condition is defined. Diameter of the spherical particle is equal to one tenth of the dimension of fluid domain – $r = L_x/10$. The particle is placed in the center of fluid domain at the beginning of the simulation.

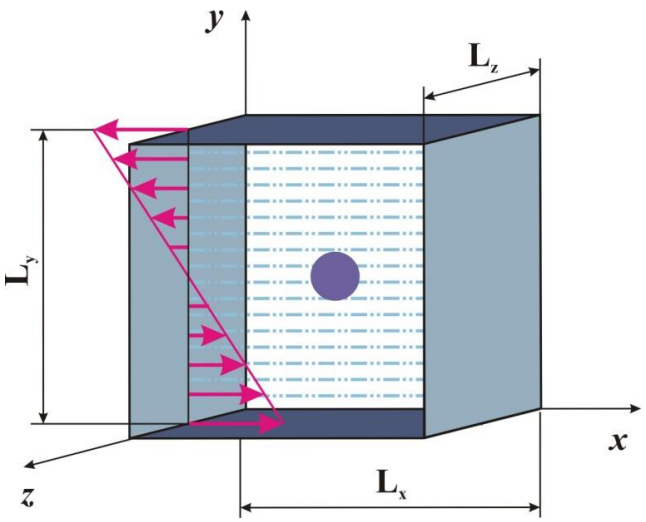

Figure 1. Geometry of the fluid domain used in the test simulation

Number of iterations of the numerical simulation is fixed at 2000, because this number is enough for the simulation to reach certain deformation of the object and at the same time, the program is executed in a reasonable time interval. This is a reasonable assumption since the same amount of computer resources and same execution time are required for each iteration. This implies that the results of this test example are relevant to determine the speed-up of the parallelized version.

When the speed-up is compared, it is necessary to consider the number of nodes, i.e. the dimension of

matrices that are processed. Thus, the overall number of nodes in lattice mesh is varied.

4. Results of Comparison of Execution Time

Fig. 2 shows the change of the speed-up of the parallelized version compared to the sequential version, when the number of nodes in the lattice mesh was varied. It should be noted that during these measurements, the number of nodes in the mesh describing the membrane of the deformable object was set to 100. As the number of nodes in the lattice mesh increases, the speed-up also increases. In this case, the maximum value of speed-up is 12.5 and it is achieved for 80000 nodes. Further increase of the number of nodes has no influence on the speed-up.

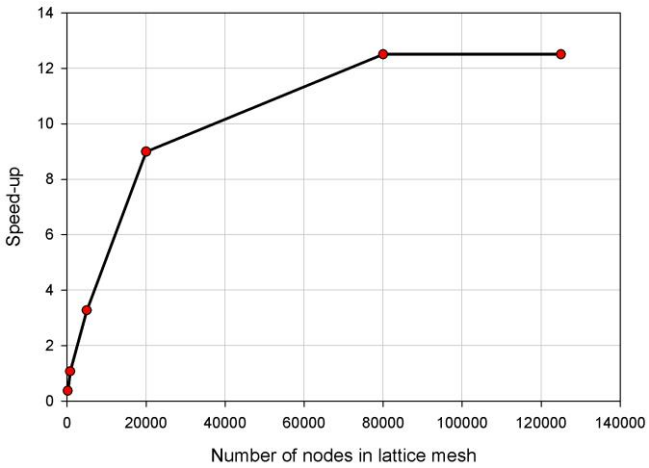

Figure 2. Diagram of variation of speed-up of parallelized software depending on the number of nodes of lattice mesh

The number of nodes in the mesh for membrane of the deformable object was also varied while the number of lattice nodes was kept constant at 80000. In this case the change of speed-up was also noticed, like it is shown in Fig. 3. For smaller number of nodes the speed-up is similar, but a larger increase in number of nodes emphasizes the fact that all solid nodes are processed synchronously in the parallelized version, while in the sequential version all these processing is for one node after the other. Further increase in the number of nodes of the membrane does not drastically increase the speed-up of parallelized version.

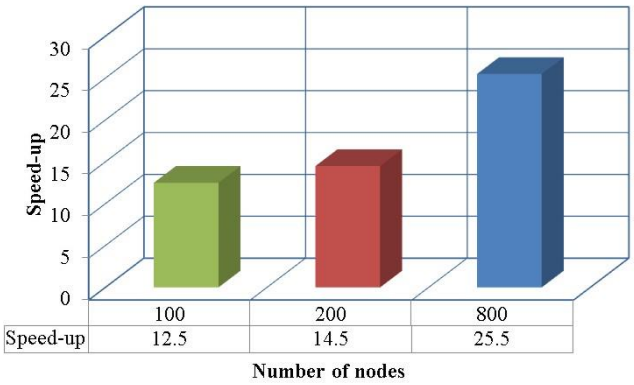

Figure 3. Comparison of speed-up of parallelized software, for different number of nodes of the mesh for the membrane of the deformable object

4. Conclusion

The parallelization of the software that simulates motion and deformation of object immersed in fluid is presented in this paper. The obtained speed-up demonstrates that the presented software for numerical simulations provides the possibility to perform simulations in a reasonable amount of time. Using the applied parallelization techniques, the software can be successfully applied to track the motion of particles in real-time.

Acknowledgements.

This paper is supported by grants from Ministry of Education, Science and Technological Development of the Republic of Serbia (projects number III41007 and ON174028). This paper is also supported by the SCOPES project (JRP: IZ73Z0_152454/1).

References

[1] VINTACHE, D., HUMBERT, B., BRASSE, D. (2010) Iterative Reconstruction for Transmission Tomography on GPU Using Nvidia CUDA. *Tsinghua Science & Technology* **15**: 11-16.

[2] ANDERSON, J.A., LORENZ, C.D., TRAVESSET, A. (2008) General purpose molecular dynamics simulations fully implemented on graphics processing units. *Journal of Computational Physics* **227**: 5342-5359

[3] TRAPNELL, C., SCHATZ M.C. (2009) Optimizing data intensive GPGPU computations for DNA sequence alignment. *Parallel Comput.* **35**: 429-440.

[4] DÍAZ, D., ESTEBAN, F.J., HERNÁNDEZ, P., CABALLERO, J.A., DORADO, G., GÁLVEZ, S. (2011) Parallelizing and optimizing a bioinformatics pairwise sequence alignment algorithm for many-core architecture. *Parallel Comput.* **37**: 244-259.

[5] MAURER, D., WIENERS, C. (2011) A parallel block LU decomposition method for distributed finite element matrices. *Parallel Comput.* **37**: 742-758.

[6] HAWICK, K.A., LEIST, A., PLAYNE, D.P. (2010) Parallel graph component labelling with GPUs and CUDA. *Parallel Comput*. **36**: 655-678.

[7] BULUÇ, A., GILBERT, J.R., BUDAK, C. (2010) Solving path problems on the GPU. *Parallel Comput.* **36**: 241-253.
[8] KOJIC, M., FILIPOVIC, N., STOJANOVIC, B., KOJIC, N. (2008) *Computer modeling in bioengineering: Theoretical Background, Examples and Software* (Chichester, England: John Wiley and Sons).
[9] FILIPOVIC, N., KOJIC, M., DECUZZI, P., FERRARI, M. (2010) Dissipative Particle Dynamics simulation of circular and elliptical particles motion in 2D laminar shear flow. *Microfluidics and Nanofluidics* **10**: 1127-1134.
[10] MALASPINAS, O.P. (2009) *Lattice Boltzmann Method for the Simulation of Viscoelastic Fluid Flows*, Ph.D. dissertation (Switzerland: École polytechnique fédérale de Lausanne).
[11] DJUKIC, T. (2015) *Modeling motion of deformable body inside fluid and its application in biomedical engineering (in Serbian)*, PhD dissertation. (Serbia: Faculty of Engineering, University of Kragujevac).
[12] DUPIN, M., HALLIDAY, I., CARE, C., ALBOUL, L., MUNN, L. (2007) Modeling the flow of dence suspensions of deformable particles in three dimensions. *Phys. Rev. E Stat. Nonlin. Soft. Matter Phys.* **6**: 066707.
[13] PESKIN, C.S. (1977) Numerical analysis of blood flow in the heart. *Journal of Computational Physics* **25**: 220-252.
[14] WU, J., SHU, C. (2010) Particulate flow simulation via a boundary condition-enforced immersed boundary-lattice Boltzmann scheme. *Commun. Comput. Phys.* **7**: 793-812.
[15] FENG, Z., MICHAELIDES, E. (2004) The immersed boundary-lattice Boltzmann method for solving fluid-particles interaction problem. *Journal of Computational Physics* **195**: 602-628.