Quick search of neighbour particles in molecular dynamics simulations

P L Novikov¹,², K V Pavsky¹,³ and A A Baranov¹,²

¹ Rzhanov Institute of Semiconductor Physics, Pr. Lavrent'eva, 13, Novosibirsk, Russia
² Novosibirsk State University, Pirogova str., 2, Novosibirsk, Russia
³ Siberian State University of Telecommunications and Information Sciences, Kirov str., 86, Novosibirsk, Russia

Abstract. Quicksort algorithm as an assistant preliminary tool is proposed for acceleration of neighbour search procedure in the frame of molecular dynamics simulations. Simple estimations are made showing that the number of operations required to determine all the neighbours within a system of particles can be reduced by 2-3 orders of magnitude due to the preliminary sorting. Test molecular dynamics simulations carried out for virtual crystal structure containing atoms manifest 9-fold acceleration achieved by the modified algorithm.

1. Introduction

Molecular dynamics (MD) simulations [1] are widely used to study various phenomena in terms of elementary processes in the nanometre scale with time duration from $10^{-13}$ to $10^{-8}$ s [2-6]. The essence of the method is the numerical solution of the dynamics equations in a manybody system, where the interaction is described by an empirical potential [7]. Typically, the system contains up to $10^5$ particles and only the nearest neighbour particles join into interaction. For this reason, in order to determine the force which acts on a certain particle, one has first to mark the group of its nearest neighbours among all the mesh of particles. For the entire system it takes ~$N^2$ operations, where $N$ is the total number of particles. It is available to reduce the number of operation to ~$N^{3/2}$ using the Verlet list [8]. However, the search of neighbours remains the most time-spending stage of calculations. Thus, any way to speedup this procedure is of great interest.

In the present paper we suggest using the quicksort algorithm [9-11] for the significant reduction of time spent for the search of neighbours in MD simulations.

2. Method

The structure studied by MD simulations is described by an array of elements. Each element of the array keeps the index $i$ of an atom, its coordinates, the components of velocity vector, etc. The simulated process includes a sequence of time steps. At a current time step one has to determine the forces acting on each atom. Initially, the order of atoms in the array is not associated with their location in space. For this reason the only way to determine the neighbours of an atom is to ask the coordinates of all other atoms and mark those located in the vicinity. This procedure, being applied to the whole system, takes ~$N^2$ operations. However, this number can be significantly reduced, if the atoms in the array are ordered so that their indices increase with one of their Cartesian coordinate (particularly, in the direction of the longest size of a system). In the ordered array the search of the neighbours for the $i$-th atom can be limited (in one direction) within the range from $j=i+1$ to $j=i+m$. Once the difference between the Cartesian coordinates of $i$-th and $j$-th atoms (in a cycle by $j$) becomes greater than the critical value (typically, 2.4 Å), the search is stopped. Analogously, the search in the opposite direction from $j=i-1$ down...
to \(j=i-m_2\) is stopped, just as the \(j\)-th atom occurs too far from the \(i\)-th one. To perform the space arrangement of the atomic array we used the quicksort algorithm. The number of operations required for sorting is estimated as

\[ n_1 = C_1 \cdot N \cdot \ln N, \]  

where \(C_1\) is a coefficient depending on the choice of the reference elements in the quicksort algorithm. The number of operations after the preliminary sorting is equal to

\[ n_2 = C_2 \cdot \frac{r}{L} N^{3/2}, \]  

where \(r\) is a cut-off radius of interaction, \(L\) is the maximum size of the structure, \(C_2\) is a coefficient equal to 0.5 approximately. The second process is expected to be longer than the first one by the factor of

\[ \frac{n_2}{n_1} = \frac{C_2 N^{3/2}}{C_1 L \ln N}. \]  

If this ratio is much less than 1, we may neglect \(n_1\) and estimate the expected gain from using the quicksort algorithm as follows

\[ \frac{n_2}{n_1} = \frac{C_2 N^{3/2}}{C_1 L \ln N}. \]  

where it is assumed that the neighbour search takes the most part of the calculation time.

3. Results and discussion

The efficiency of the modified neighbour search method was checked on the virtual crystalline structure. The slab was \(35 \times 35 \times 7\) nm\(^3\) in size and contained \(N = 2.68 \times 10^5\) Si atoms. All atoms were restricted to the nodes of the diamond-like crystal structure. The process simulated included the evaluation of temperature from 10 K up to 500 K, followed by its lifting down to 10 K. The duration of the process was \(5 \times 10^{12}\) s. For this regime no desorption occurs, so that the total number of atoms, as well as their neighbours, do not change during all the process.

Two algorithms were used: with and without the preliminary sorting, respectively. In both cases the neighbour search was made using Verlet list. The calculations were carried out at the segment of distributed computer system (CS) of SibSUTIS [12]. The test results are presented in Table 1.

| Algorithm                  | Processing time, s |
|----------------------------|--------------------|
| No sorting                 | 185.15             |
| Preliminary Quicksort      | 21.22              |

It follows from these results, that the modified algorithm gives the 9-fold gain approximately. The estimations (1-4) predict even stronger effect for the parameters used: \(L/r = 340/2.4 \approx 1.42 \times 10^2\). However, one should take into account, that the search of neighbours is not the only procedure performed during MD simulations. Much time is also spent to calculate force, acceleration and space shift for all atoms in the system.

4. Conclusion

We have proposed a tool for significant acceleration of neighbour search procedure used in molecular dynamics simulations. For this purpose, prior to the neighbour search itself, we carry out a sorting of
elements in the array using the quicksort algorithm. The sorted elements are put in ascending order of their coordinate. This allows searching potential candidates for the neighbours of a particle within the reduced range rather than the whole system. The test MD simulations for the crystal structure heating/cooling process showed a 9-fold speedup when using the modified algorithm.

Acknowledgments
This work was partially supported by the Ministry of Science and Higher Education of the Russian Federation (Project 0306-2019-0019).

References
[1] Frenkel D, Smit B 1996 “Understanding molecular simulation” Academic Press
[2] Zhong Z, Bauer G 2004 Appl. Phys. Lett. 84 1922
[3] Schmidt O G, Eberl K 2001 IEEE Trans. Electron Devices 48 1175
[4] Stangl J, Holy V, Bauer G 2004 Rev. Mod. Phys. 76 725
[5] Novikov P, Atovullaev T, Smagina Zh, Dvurechenskii A, Pavskii K 2017 Phys. Stat. Sol. (c) 14 1700200(3)
[6] Smagina Zh V, Zinovyev V A, Rudin S A, Novikov P L, Rodyakina E E, Dvurechenskiiaand A V 2018 Journal of Applied Physics 123 165302
[7] Tersoff J 1968 Phys. Rev. B 37 6991
[8] Verlet L 1967 Phys. Rev. 159 98
[9] Dean B C 2006 Discrete Applied Mathematics 154 1
[10] Martinez C, Roura S 2001 SIAM J. Comput. 31 683
[11] Sedgewick R 1978 Comm ACM 21 847
[12] Computing resources of SibSUTIS Retrieved from: https://cpct.sibsutis.ru/index.php/Main/Resources