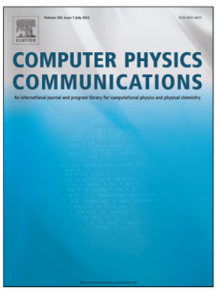
Accepted Manuscript



A multilevel-skin neighbor list algorithm for molecular dynamics simulation

Chenglong Zhang, Mingcan Zhao, Chaofeng Hou, Wei Ge

PII: S0010-4655(17)30304-1
DOI: https://doi.org/10.1016/j.cpc.2017.09.017
Reference: COMPHY 6330

To appear in: *Computer Physics Communications*

Received date : 17 March 2016
Revised date : 15 September 2017
Accepted date : 21 September 2017

Please cite this article as: C. Zhang, M. Zhao, C. Hou, W. Ge, A multilevel-skin neighbor list algorithm for molecular dynamics simulation, *Computer Physics Communications* (2017), https://doi.org/10.1016/j.cpc.2017.09.017

A Multilevel-skin Neighbor List Algorithm for Molecular Dynamics Simulation

Chenglong Zhang[1,2], Mingcan Zhao[1,2], Chaofeng Hou[1], Wei Ge*[1,2]

[1] State Key Laboratory of Multi-phase Complex Systems (MPCS),
Institute of Process Engineering (IPE), Chinese Academy of Sciences (CAS), Beijing 100190, China

[2] College of Chemistry and Chemical Engineering, University of Chinese Academy of Sciences (UCAS), Beijing 100049, China

* Corresponding author, email: wge@ipe.ac.cn

Abstract

Searching of the interaction pairs and organization of the interaction processes are important steps in molecular dynamics (MD) algorithms and are critical to the overall efficiency of the simulation. Neighbor lists are widely used for these steps, where thicker skin can reduce the frequency of list updating but is discounted by more computation in distance check for the particle pairs. In this paper, we propose a new neighbor-list-based algorithm with a precisely designed multilevel skin which can reduce unnecessary computation on inter-particle distances. The performance advantages over traditional methods are then analyzed against the main simulation parameters on Intel CPUs and MICs (many integrated cores), and are clearly demonstrated. The algorithm can be generalized for various discrete simulations using neighbor lists.

Keywords high performance computing; molecular dynamics simulation; neighbor-list-based algorithm; searching efficiency.

1 Introduction

Molecular dynamics (MD) simulation is one of the most commonly used discrete simulation methods, which was first proposed by Alder & Wainwright in 1956 [1] based on deterministic Newtonian mechanics. With various potential functions, it has been applied in physics, chemistry, biology, materials and medical science [2-6]. All the applications for short-range MD require calculation of the inter-particle (atoms or molecules) potential energy and its gradients (forces) where two-body interactions are commonly used, such as the Lennard-Jones (LJ) potentials [7] widely used in the simulation of simple gases and liquids. A significant feature of these potentials is that they decay very fast and become negligible when the distance is larger than several characteristic particle diameters (σ). Therefore, a cut-off radius (R_c) is usually applied to define the computed range of interactions [8]. Afterwards, the velocities (**v**) and positions (**r**) of the particles can be updated by the integration based on Newton's Second Law.

The computational cost at each time step is concentrated in the search for neighboring particles and processing of the interactions. To avoid unnecessary inter-particle distance calculation, two mainstream techniques, the linked cell list (LCL) [9] and the Verlet list (VL) [10], are widely used. In the LCL algorithm, all particles are mapped to cubic cells according to their positions, the edge size of the cells (L_c) is equal to or slightly larger than R_c, so the inter-particle distances are only computed for each particle among its neighbors in the same cell and adjacent cells. However, a straightforward implementation of the LCL algorithm is still of low efficiency, in which the majority of the computed pairwise distances are not within R_c. Mason [11] improved the algorithm with arbitrary shaped cells to increase the ratio of effective neighbors, and Gonnet [12] added a sorting procedure according to the particle positions to reduce unnecessary searching. Nevertheless, the efficiency of the LCL algorithm [13,14] is inferior to the VL algorithm in many cases when the particle motion is not violent and the cost for updating the neighbor lists is high.

The VL is a neighbor list valid for several steps, which is constructed by including the currently interacting