Computer Physics Communications 204 (2016) 64–73

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Efficient parallelization of analytic bond-order potentials for large-scale atomistic simulations

C. Teijeiro [a,*], T. Hammerschmidt [a], R. Drautz [a], G. Sutmann [a,b]

[a] Interdisciplinary Centre for Advanced Materials Simulation (ICAMS), Ruhr-University Bochum, D-44801 Bochum, Germany

[b] Jülich Supercomputing Centre (JSC), Institute for Advanced Simulation (IAS), Forschungszentrum Jülich, D-52425 Jülich, Germany

ARTICLE INFO

Article history:
Received 30 December 2015
Received in revised form
17 March 2016
Accepted 19 March 2016
Available online 31 March 2016

Keywords:
Bond-order potentials
Large-scale simulations
Parallelization
MPI

ABSTRACT

Analytic bond-order potentials (BOPs) provide a way to compute atomistic properties with controllable accuracy. For large-scale computations of heterogeneous compounds at the atomistic level, both the computational efficiency and memory demand of BOP implementations have to be optimized. Since the evaluation of BOPs is a local operation within a finite environment, the parallelization concepts known from short-range interacting particle simulations can be applied to improve the performance of these simulations. In this work, several efficient parallelization methods for BOPs that use three-dimensional domain decomposition schemes are described. The schemes are implemented into the bond-order potential code BOPfox, and their performance is measured in a series of benchmarks. Systems of up to several millions of atoms are simulated on a high performance computing system, and parallel scaling is demonstrated for up to thousands of processors.

1. Introduction

Atomistic simulations have been proven successful in describing soft and hard materials to explain or predict their behavior under given thermodynamic conditions. These properties are thereby fully determined by the interatomic interactions and the dynamical behavior of a system can be accessed by exploring the system's phase space. In the simplest case, approximations based on pair interactions between particles are applied, and they can be linearly superposed to obtain the total force acting on a particle. However, the high computational complexity of $\mathcal{O}(N^3)$ with increasing number of degrees of freedom N limits its applicability to systems of $N \approx 1000$ atoms, while linear scaling $\mathcal{O}(N)$ implementations of density functional theory (DFT) start to become more efficient than the $\mathcal{O}(N^3)$ implementations also at about 1000 atoms. Therefore, if collective effects such as structural transitions, defect formation or non-equilibrium processes are to be studied on longer time and length scales, accurate approximations to describe the features of interest have to be introduced.

Analytic bond-order potentials are based on a coarse graining of DFT and compute atomic energies and forces by a moments expansion of length m of the local density of states, $n_{i\alpha}$ [1–3]. Thus, these potentials allow to compute atomistic interactions with high accuracy at a computational complexity of $\mathcal{O}(N)$. In the BOP formalism, the total binding energy of the system, U_B, is given as $U_B = U_{bond} + U_{prom} + U_{rep}$, where the promotion energy U_{prom} can be calculated as sums over occupied electronic states and the repulsive energy U_{rep} is represented as a classical pair or many-body interaction. The calculation of the bond energy takes most of the computational cost, and is given by

$$U_{bond} = 2 \sum_{i\alpha} \int_{-\infty}^{E_F} dE\,(E - E_{i\alpha})\, n_{i\alpha}(E) \tag{1}$$

where the local density of states $n_{i\alpha}$ is represented as a moments expansion, with the moments

$$\mu_{i\alpha}^N = \int E^N n_{i\alpha}(E) dE = \sum_{j\beta k\gamma \ldots q\kappa} \langle i\alpha|\hat{H}_{i\alpha j\beta}|j\beta\rangle\langle j\beta|\hat{H}_{j\beta k\gamma}|k\gamma\rangle \ldots \langle q\kappa|\hat{H}_{q\kappa i\alpha}|i\alpha\rangle. \tag{2}$$

In Eq. (1) E_F represents the Fermi energy and $E_{i\alpha}$ the energy of the occupied orbital α of atom i. The density of states $n_{i\alpha}$ is constructed by sampling a neighborhood of each atom i, which is accomplished by building self-returning paths as a chain of links between nearest neighbors (e.g. $\langle i\alpha|\hat{H}_{i\alpha j\beta}|j\beta\rangle$). The final

* Corresponding author.
E-mail addresses: carlos.teijeirobarjas@rub.de (C. Teijeiro), thomas.hammerschmidt@rub.de (T. Hammerschmidt), ralf.drautz@rub.de (R. Drautz), g.sutmann@fz-juelich.de (G. Sutmann).

http://dx.doi.org/10.1016/j.cpc.2016.03.008
0010-4655/

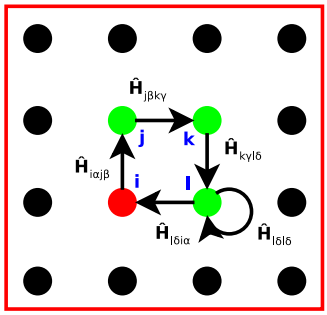

Fig. 1. Sample path of length 5 (contribution to the 5th moment for atom i and orbital α).

result is obtained by summing up the contributions of all path combinations from atom i of given length. The length of paths is determined by the order of the moment expansion, m, where the accuracy converges to the tight binding result with increasing number of moments. A sample path from an atom i is depicted in Fig. 1. For a more detailed description of bond-order potentials, see e.g. Refs. [4,5].

In order to exploit the locality of the energy calculation, it is natural to transfer well established concepts from the parallelization of short range interacting systems [6]. Efficient implementations for simple interaction models have proven their applicability to system sizes of billions of atoms [7,8] and beyond [9]. For simulations with local interactions, the domain decomposition method has become a standard approach. It consists in the distribution of both particles and space onto the processors, which allows to increase the overall size of the system since (1) the workload is shared between the processors and (2) the size of memory is shared, both of which are limiting factors in single core computations.

In the present article we propose different approaches based on domain decomposition for the parallelization of the bond-order potential code BOPfox [10]. The paper is organized in the following parts: Section 2 presents a general motivation and basic information about the sequential simulation code. Section 3 describes and compares the implementation of several parallel algorithms. Section 4 presents the performance evaluation of the implementations in terms of benchmarks for different number of threads, moment expansions and problem sizes. Finally, Section 5 gives the main conclusions of this work.

2. Motivation and code description

The BOPs provide a rigorous expansion for the density of states and the binding energy. The algorithmic problem is the construction of all closed paths of a given length (cf. Eq. (1) and Fig. 1) between neighboring atoms. As shown in Ref. [11], the complexity is confirmed to be $\mathcal{O}(N)$ when increasing the number of atoms N, but the prefactor per atom increases as $m^{9/2}$ with the moment expansion m. This strong increase of the prefactor for energy and force computations poses a significant computational load for multi-million atom systems, which are necessary, e.g., to describe the development of dislocation networks. Therefore an efficient parallelization is necessary to reduce the overall wall-clock time of large-scale simulations.

An initial analysis of the simulation code BOPfox reveals that the large number of path combinations that need to be evaluated for every atom clearly dominates the CPU time compared with the rest of tasks. Each path is computed as the result of a chain of multiplications of small pairwise Hamiltonian matrices between neighbor atoms (cf. Fig. 1), and all together contribute to the computation of energies and forces in the system. This task is accomplished in three steps for a given moment expansion m:

- Paths of length 1 from every atom are directly obtained from the bond information (i.e. pairwise Hamiltonians) associated with its nearest neighbors, including itself.
- Paths of length between 2 and $\lceil m/2 \rceil$ (from now on referred to as *half-length paths*) are computed.
- Paths of length between $\lceil m/2 \rceil + 1$ and m are obtained by merging (i.e. multiplying) pairs of the paths computed previously.

A rough profiling of the BOPfox code indicates that these multiplications consume more than 90% of the total execution time of a sample execution [11]. Therefore, the computation of paths is the target for the parallelization presented in this work.

3. Domain decomposition algorithms

A parallelization based on the concept of independent domains implies a tradeoff between two factors: (1) the use of local computations, which imply the definition of an overlap area with a certain level of redundancy, and (2) the use of communications, in order to transfer partial computations of paths from one domain to the other. Communications are required when a given domain does not have all necessary information to compute all self-returning paths for local atoms, which is usually the case for atoms close to the domain boundary. It is therefore required to balance the cost associated with data transfer between domains and redundant computations within the overlap region.

An additional consequence of this tradeoff is that, when some extreme cases are considered, the nature of the simulation hampers a really efficient parallelization. Considering moment expansions of length m, each atom is accessing all atoms in a spherical volume of radius $\lceil m/2 \rceil \times R_{cut}$, which should be of the order of a domain size and considerably smaller than the system size for an efficient parallel execution. According to this, a minor assumption has been considered for the present implementations of domain decomposition, i.e. the computation of paths requires information from only the local domain $\mathcal{D}$ and its neighbor domains, with which $\mathcal{D}$ shares a common border. The idea is to restrict the local-area information in every process to its associated domain and a relatively small portion of its neighbor domains, so that the number of domains in which a system is subdivided is coherent with the system size.

Previous simulations of pair potentials or BOPs with low-order expansions (available in widely used simulation codes like LAMMPS [12,13]) can benefit from highly efficient parallel approaches such as the neutral-territory [14] and the eighth-shell approach [15,16], which may also be used for analytic BOPs. However, here the intense computation in a volume around each atom suggests the definition of overlap zones according to the moment expansion m, in order to allow a mainly local computation of half-length paths for each domain. Consequently, the size of the domain is preferably defined large enough with respect to the overlap zone in order to keep communications as simple and reduced as possible.

Following these premises, three types of algorithms are now presented in order to illustrate different levels of the tradeoff between communications and computations. Starting from a scenario with redundant path computations with no additional communications, the tradeoff is applied to get an approach with no redundancy and complementary communications. According to this, the order of presentation of the algorithms is as follows: (1) minimum redundancy (*MIN-REDUND*), (2) path communication algorithms (*FULL-SHELL* and *HALF-SHELL*) and (3) force communications (*BOND-COMMS*). Fig. 2 illustrates the differences between these approaches in terms of the tradeoff between local computations and the amount of required communications with the type of exchanged information (either path matrices or force contributions). All these algorithms have been implemented using the MPI library, which is a de-facto standard approach for parallel computing on distributed systems.