Computer Physics Communications 219 (2017) 246–254

Contents lists available at ScienceDirect

Computer Physics Communications

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

A derivation and scalable implementation of the synchronous parallel kinetic Monte Carlo method for simulating long-time dynamics

Hye Suk Byun [a], Mohamed Y. El-Naggar [a,b,c], Rajiv K. Kalia [a,d,e,f], Aiichiro Nakano [a,b,d,e,f,*], Priya Vashishta [a,d,e,f]

[a] Department of Physics & Astronomy, University of Southern California, Los Angeles, CA 90089-0242, USA
[b] Department of Biological Sciences, University of Southern California, Los Angeles, CA 90089-0242, USA
[c] Department of Chemistry, University of Southern California, Los Angeles, CA 90089-0242, USA
[d] Department of Computer Science, University of Southern California, Los Angeles, CA 90089-0242, USA
[e] Department of Chemical Engineering & Materials Science, University of Southern California, Los Angeles, CA 90089-0242, USA
[f] Collaboratory for Advanced Computing and Simulations, University of Southern California, Los Angeles, CA 90089-0242, USA

ARTICLE INFO

Article history:
Received 3 February 2017
Received in revised form 23 April 2017
Accepted 31 May 2017
Available online 27 June 2017

Keywords:
Kinetic Monte Carlo simulation
Divide-and-conquer algorithm
Parallel computing

ABSTRACT

Kinetic Monte Carlo (KMC) simulations are used to study long-time dynamics of a wide variety of systems. Unfortunately, the conventional KMC algorithm is not scalable to larger systems, since its time scale is inversely proportional to the simulated system size. A promising approach to resolving this issue is the synchronous parallel KMC (SPKMC) algorithm, which makes the time scale size-independent. This paper introduces a formal derivation of the SPKMC algorithm based on local transition-state and time-dependent Hartree approximations, as well as its scalable parallel implementation based on a dual linked-list cell method. The resulting algorithm has achieved a weak-scaling parallel efficiency of 0.935 on 1024 Intel Xeon processors for simulating biological electron transfer dynamics in a 4.2 billion-heme system, as well as decent strong-scaling parallel efficiency. The parallel code has been used to simulate a lattice of cytochrome complexes on a bacterial-membrane nanowire, and it is broadly applicable to other problems such as computational synthesis of new materials.

1. Introduction

Ever-increasing processing power of parallel computers [1] is continuously extending the spatiotemporal scales of particle simulations [2], where each particle could represent an atom in molecular dynamics (MD) simulations or a human in agent-based social simulations. Essential to extending the spatial scale is linear-scaling algorithms based on spatial locality principles, in which the computational complexity scales linearly with the number of particles N [3–6]. A harder problem is to increase the time scale of processes that can be simulated, due to the inherently sequential nature of time as a result of causality [7]. In some cases, temporal locality principles alleviate the sequential-time bottleneck in MD simulations [2,4,6,8]. Namely, a many-particle system tends to stay near a local minimum-energy configuration over a long duration of time, which is bounded by a rare transition over short time period to another minimum. In such a case, the transition state theory (TST) [9,10] uses a local equilibrium assumption to reformulate the sequential long-time dynamics as computationally more efficient parallel search for low-activation transition events [11–14], where the rates of the events are computed from their activation barriers.

The most widely used simulation method based on TST is kinetic Monte Carlo (KMC) [15–20]. In KMC simulations, an event to occur is stochastically selected from a database of events. The simulated time progresses according to Poisson statistics, where the time increment t at each KMC step is inversely proportional to the sum of the rates of all possible events. Since the summed rate grows as at least $O(N)$, KMC simulations progress much more slowly for larger systems, *i.e.*, $t = O(1/N)$. With an $O(N)$ implementation of the computation for a single KMC step, therefore, the computational complexity to simulate a physical time duration of τ is $O(N \times \tau/t) = O(N \times \tau N) = O(N^2)$. The conventional KMC algorithm is thus not scalable for large N. Here, it should be noted that single-step KMC computation can be performed in $O(\log N)$ time instead of $O(N)$ [21], or even faster if the types of possible events are bounded as $O(1)$ [22].

More efficient long-time simulation is possible because of the spatiotemporal locality of activated events. Namely, they are usually localized not only temporally but also spatially [23]. This leads

* Corresponding author at: Collaboratory for Advanced Computing and Simulations, University of Southern California, Los Angeles, CA 90089-0242, USA. Fax: +1 213 821 2664.
E-mail address: anakano@usc.edu (A. Nakano).

http://dx.doi.org/10.1016/j.cpc.2017.05.028
0010-4655/

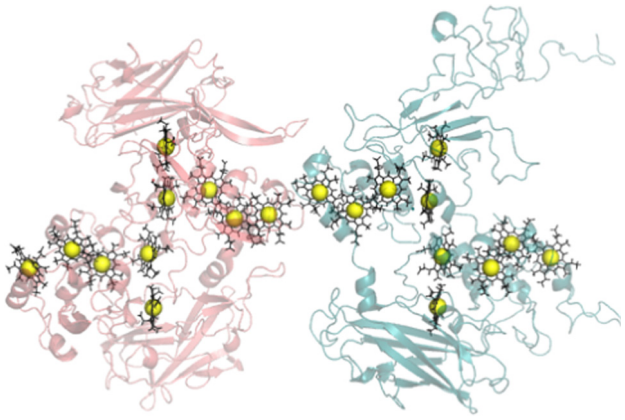

Fig. 1. Heme groups in a dimer of decaheme cytochromes, where the Fe atom within each heme is represented by a yellow sphere. Each of the two cytochromes (colored magenta and cyan) contains 10 hemes.

to computationally more efficient simulation methods that concurrently sample multiple, spatially localized events [24–28]. In such cases, event sampling may be enhanced by an ensemble-mean field approach. For example, a time-dependent Hartree (TDH) approximation has been employed to sample the dynamics of a small key subsystem within a large molecule [29]. Here, an ensemble of MD simulations for the subsystem is embedded in an MD simulation of the entire molecule, where the subsystem and the rest of the molecule interact via an ensemble-mean field [29]. In the light of the localized nature of atomistic events [23], we here introduce a local transition-state (LTS) approximation, in conjunction with the TDH approximation, into the KMC simulation method. This leads to a divide-and-conquer strategy that simulates multiple events concurrently based on spatial decomposition. The resulting simulation method is equivalent to the synchronous parallel KMC (SPKMC) method [25–28]. Furthermore, we introduce a dual linked-list cell (DL^2C) method to further reduce the computational cost. This paper is organized as follows. Section 2 describes a derivation and implementation of the SPKMC algorithm. Numerical results are presented in Section 3, and Section 4 contains conclusions.

2. Methods

In this section, we present a new derivation of the synchronous parallel kinetic Monte Carlo (SPKMC) algorithm and its implementation on parallel computers using a dual linked-list cell (DL^2C) method. As a concrete example, we use electron-transfer (ET) dynamics between heme groups in a large network of cytochromes [30,31]. (Heme is an organic compound called porphyrin that contains an iron (Fe) atom at its center.) Fig. 1 shows a dimer of decaheme cytochromes and the heme groups in it. The Fe atom in each heme can exist in either of the two valence states, Fe^{2+} or Fe^{3+}. Conversion of irons between Fe^{2+} and Fe^{3+} allows for the hopping of electrons between adjacent hemes. KMC simulation treats electron-hopping events in a network of N hemes, where a heme site at position $\mathbf{q}_i$ is labeled by index $i \in \{1, \ldots, N\}$. The ith heme is either occupied by an electron ($n_i = 1$ or reduced, corresponding to Fe^{2+}) or unoccupied ($n_i = 0$ or oxidized, corresponding to Fe^{3+}), where n_i is the electron occupation number of the ith heme. The system dynamics are characterized by (i) electron hopping rates W_{ji} from the ith heme to the jth heme for adjacent (i, j) pairs, (ii) electron injection rate W_{red} into a selected entrance heme i_{ent}, and (iii) electron-ejection rate W_{ox} from an exit heme i_{exit}. In our work, W_{ji} is computed from the positions of hemes, $\mathbf{q}_i$ and $\mathbf{q}_j$, and their precomputed free energies, G_i and G_j [30]. In addition, W_{ji} depends on the occupation numbers, n_i and n_j, since an electron can hop from i to j only when $n_i = 1$ and $n_j = 0$. Due to the exponential decay of the electron-hopping rate with respect to the heme-pair distance, $W_{ji} = 0$ when $|\mathbf{q}_i - \mathbf{q}_j| > q_{\text{cut}}$, where $q_{\text{cut}} \sim 1$ nm is a cutoff distance. The following discussion also applies to other applications such as photoexcitation dynamics in solar cells [20] and computational synthesis of new materials based on chemical vapor decomposition and other techniques [19], as long as an event is spatially localized within a cutoff distance.

2.1. Kinetic Monte Carlo simulation

To introduce a notation necessary for the derivation of the SPKMC algorithm, Appendix A describes the conventional KMC simulation method [15–20]. We initialize a KMC simulation by emptying all heme sites and resetting the time to 0. At each KMC step, one of the following events occurs: (i) an electron is injected with rate W_{red} if the entrance heme is unoccupied; (ii) an electron is ejected with rate W_{ox} if the exit heme is occupied; or (iii) an electron hops from heme i to one of its nearest-neighbor hemes, j, with rate W_{ji} if heme i is occupied and heme j is unoccupied. The method for calculating the rates for the ET dynamics can be found in Ref. [30]. KMC simulation consists of a time-stepping loop. Let e be one of the possible events listed above, with W_e being its rate, E be the total number of possible events, and

$$W = \sum_{e=1}^{E} W_e \tag{1}$$

be the sum of the rates of all possible events. At each KMC step, the time is incremented by

$$t = -\ln(\xi_1)/W, \tag{2}$$

where ξ_1 is a uniform random number in the range [0,1]. The probability of choosing a particular event is proportional to its rate, and specific event e^* is chosen such that

$$e^* = \min_e \left\{ \sum_{c=1}^{e} W_c > W\xi_2 \right\}, \tag{3}$$

where ξ_2 is another uniform random number in [0,1].

2.2. Synchronous parallel KMC algorithm

The standard KMC method in Section 2.1 is not scalable to larger system sizes. The cumulative event rate W grows as $O(N)$, and accordingly the time scale of the simulation determined by its inverse becomes progressively smaller in larger systems, as is seen in Eq. (2). To overcome this scaling problem, we parallelize KMC simulations in a divide-and-conquer (DC) fashion, using a synchronous formulation and graph coloring to avoid conflicting events [25–28]. To do so, we first introduce a local transition-state (LTS) approximation, in which events outside a cutoff distance are assumed to be statistically independent. We then introduce a time-dependent Hartree (TDH) approximation, *i.e.*, the simulated system is subdivided into spatially localized domains and local events in a domain are sampled independently of those in the other domains. Appendix B provides a formal derivation of the resulting SPKMC algorithm.

Domain decomposition: The SPKMC algorithm partitions the 3-dimensional space $\Re^3$ into spatially localized domains $\Re^3_d$ that are mutually exclusive,

$$\Re^3 = \bigcup_d \Re^3_d; \quad \Re^3_d \cap \Re^3_{d'} = \varnothing. \tag{4}$$

For simplicity, we consider a simple mesh decomposition, in which the total rectangular space is subdivided into domains of equal