Journal of Computational Physics 281 (2015) 864–875

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Parallelized event chain algorithm for dense hard sphere and polymer systems

Tobias A. Kampmann, Horst-Holger Boltz, Jan Kierfeld *

Physics Department, TU Dortmund University, 44221 Dortmund, Germany

ARTICLE INFO

Article history:
Received 2 July 2014
Received in revised form 9 October 2014
Accepted 29 October 2014
Available online 4 November 2014

Keywords:
Parallelization
Monte Carlo
Event chain algorithm
Semiflexible polymers
Filament bundles

ABSTRACT

We combine parallelization and cluster Monte Carlo for hard sphere systems and present a parallelized event chain algorithm for the hard disk system in two dimensions. For parallelization we use a spatial partitioning approach into simulation cells. We find that it is crucial for correctness to ensure detailed balance on the level of Monte Carlo sweeps by drawing the starting sphere of event chains within each simulation cell with replacement. We analyze the performance gains for the parallelized event chain and find a criterion for an optimal degree of parallelization. Because of the cluster nature of event chain moves massive parallelization will not be optimal. Finally, we discuss first applications of the event chain algorithm to dense polymer systems, i.e., bundle-forming solutions of attractive semiflexible polymers.

1. Introduction

Since its first application to a hard disk system [1], Monte Carlo (MC) simulations have been applied to practically all types of models in statistical physics, both on-lattice and off-lattice. MC samples microstates of a thermodynamic ensemble statistically according to their Boltzmann weight. It requires knowledge of microstate energies rather than interaction forces. In its simplest form, the Metropolis MC simulation [1], a MC simulation is easily implemented for any system by offering a new microstate to the system and accepting or rejecting according to the Metropolis rule, which is based on the Boltzmann factor. The Metropolis rule guarantees detailed balance. Typical local MC moves, such as single spin flips in spin systems or single particle moves in off-lattice systems of interacting particles, are often motivated by the actual dynamics of the system. Sampling with local moves can become slow, however, under certain circumstances, most notably, close to a critical point, where large correlated regions exist, or in dense systems, where acceptable moves become rare.

There have been two routes for major improvement of MC simulations to address the issues of critical slowing down and dense systems, namely cluster algorithms and parallelization:

(i) One route for improvement is *cluster algorithms*, which go beyond local Metropolis MC. Such methods construct MC moves of large *non-local* clusters. Ideally, clusters are generated in a way that the MC move of the cluster is performed *rejection-free*. Clusters have to be sufficiently large and cluster building has to be sufficiently fast to gain performance. For lattice spin systems, the Swendsen–Wang [2] and Wolff [3] algorithms are the most notable cluster algorithms with enormous performance gains close to criticality, where they reduce the dynamical exponent governing the critical slowing down.

* Corresponding author.
E-mail addresses: tobias.kampmann@tu-dortmund.de (T.A. Kampmann), jan.kierfeld@tu-dortmund.de (J. Kierfeld).

http://dx.doi.org/10.1016/j.jcp.2014.10.059
0021-9991/

For off-lattice interacting particle systems, the simplest of which are dense hard spheres, cluster algorithms have been proposed based on different types of cluster moves. In Ref. [4], a cluster algorithm based on pivot moves has been proposed, which was applied to different hard core systems [5,6], and variants formulated for soft core systems [7]. In Ref. [8], the event chain (EC) algorithm has been proposed, which generates large clusters of particles in the form of a chain of particles, which are moved simultaneously and rejection-free. ECs become long in the dense limit, which significantly reduces autocorrelation times.

(ii) The other route is *parallelization*, as multi-processor computing has become widely available both in the form of multiple CPUs and, in recent years, in the form of graphic processing units (GPUs). On GPUs, "massively" parallel algorithms can significantly improve performance of simulations. For massively parallel computation on GPUs, the algorithm has to be data-parallel to gain performance, i.e., the simulation system has to be dividable into pieces, which can be updated independently accessing a limited shared memory. This has been achieved very efficiently for molecular dynamics (MD) simulations [9,10] and MC simulations [11]. It is an ongoing effort to massively parallelize other simulation algorithms.

It seems attractive to combine both strategies and search for parallelization options for cluster algorithms. For conventional Metropolis MC based on single spin/particle moves or MD simulations with finite range of interactions, the parallelization strategy typically consists in spatial partitioning of the system into several domains, on which the algorithm works independently, i.e., in a data-parallel manner. Such algorithms can also be massively parallelized for GPUs. For cluster algorithms the suitable parallelization strategy is less clear. If a spatial decomposition strategy is to be used, it must be applied to the cluster selection and cluster identification. This has been achieved for Swendsen/Wang and Wolff algorithms for lattice spin models recently [12,13], and these algorithms have been implemented with efficiency gains on GPUs.

The EC algorithm for dense hard sphere system [8] relies on a sequential selection of a chain of particles as the cluster to be updated (and will be explained in more detail below), which makes massive parallelization difficult. In this article, we want to investigate a strategy to apply spatial partitioning into independent simulation cells as parallelization technique to the EC algorithm for hard sphere systems in order to combine performance gains from cluster algorithm and parallelization. A similar approach has been proposed in Ref. [14]. Here, we also systematically test parallelized EC algorithms for correctness and efficiency using the well-studied example of two-dimensional hard disks. As a result, we find that for the parallel EC algorithm to work correctly it is crucial how the starting points of ECs are selected during a sweep in a simulation cell.

Moreover, the most efficient parallelization will not be massive; the scalability will be limited by the nature of the EC algorithm itself. The EC algorithm is most efficient if EC clusters have an optimal size [8], which is related to the particle density. If simulation cells become too small compared to the typical size of EC clusters, parallelization becomes inefficient. The parallel EC algorithm will, therefore, be best suited for multicore CPUs with shared memory.

We present and systematically test the parallel EC algorithm in detail in the context of the hexatic to liquid transition in two-dimensional melting of a system of hard disks and give an outlook to dense polymeric systems in the end.

1.1. Two-dimensional melting

Two-dimensional melting, especially in the simplest formulation of a system of hard (impenetrable) disks, is a fascinating example of a classical phase transition. Hard disks are an epitome of a system that is easily described and quickly implemented in a (naive) simulation, but very hard to tackle analytically. Hard disks have been subject of computational studies ever since the seminal works of Metropolis et al. [1], which can be regarded as a starting point for MC simulations and of the area of computational physics as a whole.

For two-dimensional melting, there has been a long debate on the nature of the phase transitions leading from the liquid to the solid phase (see, e.g., Ref. [15] for a review). In two dimensions genuine long-range positional order is not possible because of thermal fluctuations, but a two-dimensional fluid with short-range interactions can only condense into a solid phase with algebraically decaying positional correlations. The KTHNY-theory [16–18] describes two-dimensional melting as defect-mediated two-step melting process: Starting from the quasi-ordered solid in a first transition dislocations unbind, which destroys the translational order [16] resulting in a so-called hexatic liquid, which remains orientationally ordered. In a subsequent second transition, disclinations unbind destroying also the remaining orientational order [17,18]. Both transitions are continuous phase transitions of Kosterlitz–Thouless type. Alternatively, a weak first order phase transition has been discussed, where a liquid phase with lower free energy appears before the instability of the ordered phase with respect to defect-unbinding sets in. Both phases then coexist in a region in parameter space. Simulations on hard disks in two dimensions gave indecisive results regarding this issue for many years (see Ref. [19] for a discussion).

The EC algorithm helped to settle this issue for the two-dimensional hard disk system. In Ref. [20] it was shown that the transition from the hexatic to the liquid phase is a weak first order transition by identifying a region of phase coexistence and a characteristic pressure loop if the pressure P is measured as a function of the particle density. No such loop was detected between the hexatic and solid phase with quasi-long-range positional order indicating that the hexatic to solid transition is continuous. In Ref. [21], these results were corroborated by massively parallel local MC and MD simulations.