Event-chain Monte Carlo algorithms for hard-sphere systems

Etienne P. Bernard,†† Werner Krauth,† and David B. Wilson†‡

1 CNRS-Laboratoire de Physique Statistique, Ecole Normale Supérieure, 24 rue Lhomond, 75231 Paris Cedex 05, France
2 Microsoft Research, One Microsoft Way, Redmond, Washington 98052, USA

(Dated: May 01, 2009)

In this paper we present the event-chain algorithms, which are fast Markov-chain Monte Carlo methods for hard spheres and related systems. In a single move of these rejection-free methods, an arbitrarily long chain of particles is displaced, and long-range coherent motion can be induced. Numerical simulations show that event-chain algorithms clearly outperform the conventional Metropolis method. Irreversible versions of the algorithms, which violate detailed balance, improve the speed of the method even further. We also compare our method with a recent implementations of the molecular-dynamics algorithm.

In this paper, we propose a class of Monte Carlo algorithms for hard-sphere systems: the “event-chain” algorithms. In contrast to the Metropolis algorithm, these methods are rejection-free. In a single move, they displace an arbitrary long chain of spheres, each advancing until it strikes the next one. Event-chain algorithms are generically faster than other Markov-chain algorithms, in part because the mean-square displacements of individual particles are larger. In addition, one of the event-chain algorithms moves coherently over long distances. This further improves equilibration times. Finally, the absence of rejections allows us to consider irreversible versions, which violate detailed balance, but preserve the correct stationary distribution. These versions accelerate the algorithm even further. The event-chain algorithms clearly outperform the traditional Metropolis algorithm for hard-disk and hard-sphere systems.

In the local Metropolis algorithm, the move of a sphere is accepted if it induces no overlaps, and is rejected otherwise (see Fig. 1). This algorithm is notoriously slow at high density because, although a particle can move back and forth in the “cage” formed by its neighbors, it cannot easily escape from it. To overcome the limitations of the local Metropolis algorithm, coordinated particle moves have been considered: When the displacement of one sphere generates overlaps with other spheres, the latter are in turn moved out of the way. The rejection-free pivot cluster algorithm, for example, works extremely well for binary or for polydisperse mixtures, but it breaks down for the high densities of interest in two-dimensional melting.

In Jaster’s algorithm, overlapping spheres forming a chain are displaced, all of them by a fixed vector, until a configuration without overlaps is obtained (see Fig. 1). If a sphere branches out to more than one other sphere during the chain construction, the move is rejected (see...
This happens frequently, so the expected chain length is short and Jaster’s algorithm barely faster than the local Metropolis algorithm.

In the algorithms presented here, each move consists in a deterministic chain of “events”: a disk advances until it strikes another one, which then in turn is displaced. The move starts with a randomly chosen disk, and stops when the lengths of all displacements add up to a total-displacement parameter \( \ell \) (see Fig. 2). This parameter allows the move to be reversible without rejections. To satisfy detailed balance, the move must also conserve configuration-space volume. This implies that when a disk strikes a neighbor, the latter may be displaced either in the original direction (“straight event-chain” (SEC) algorithm) or in the direction reflected with respect to the symmetry axis of the collision (“reflected event-chain” (REC) algorithm) (see Fig. 2). In a periodic box, and with the initial direction \( \theta \) sampled uniformly in \([0, \pi]\), both versions, which we call “reversible”, preserve the uniform measure because of detailed balance.

Let us define the “free path” \( \lambda = \lambda(\theta) \) of a disk as the distance it must move in direction \( \theta \) to strike another particle. The ensemble average of \( \lambda \) yields the mean-free path \( \lambda_0 \). The distribution of the free path \( \pi(\lambda/\lambda_0) \) is well approximated by an exponential

\[
\pi(\lambda/\lambda_0) \simeq \exp(-\lambda/\lambda_0),
\]

even in the solid phase (see Fig. 4). This exponential ansatz allows us to estimate the mean-square displacement for the local Metropolis algorithm, supposing,

\[
\int_0^{\infty} \pi(|\Psi|) d|\Psi|
\]
for simplicity, that the proposed moves have fixed step size \( \ell \) in random directions. The acceptance probability

\[ p_{\text{acc}}(\ell) = \exp(-\ell/\lambda_0) \]

yields \( \langle \Delta^2_X(\ell) \rangle = \ell^2 \exp(-\ell/\lambda_0) \), which is maximized when \( \ell = 2\lambda_0 \),

\[ \max_{\ell} \langle \Delta^2_X(\ell) \rangle = \langle \Delta^2_X(2\lambda_0) \rangle = 4\lambda_0^2/e^2. \quad (3) \]

To estimate the mean-square displacement for the event-chain algorithms, we suppose that the lengths of subsequent displacements in the chain are independent. In this case, the expected number of particles in the chain equals \( \ell/\lambda_0 + 1 \). We index the displacement during one event-chain move through a time-like variable \( s \) with \( 0 \leq s \leq \ell \). The mean-square displacement of an event-chain move (the expected sum of the squares of the individual displacements) can be expressed through the probability \( \pi(s,s') \) that two times \( s \) and \( s' \) belong to the same particle movement:

\[ \langle \Delta^2_X(\ell) \rangle = \int_0^\ell \int_0^\ell ds \, ds' \, \pi(s,s'). \]

With the ansatz of Eq. (2), we have \( \pi(s,s') = \exp\left(-|s-s'|/\lambda_0\right) \). This yields the mean-square displacement per particle, which can be viewed as a short-time (local) diffusion coefficient:

\[ D^\text{loc}_\text{loc}(\ell) = \frac{\langle \Delta^2_X(\ell) \rangle}{\langle M(\ell) \rangle} = 2\lambda_0^2 \exp\left(-\ell/\lambda_0 + \ell/\lambda_0 - 1\right) \frac{\ell}{\lambda_0 + 1} \to 1 \text{ for } \ell/\lambda_0 \to \infty. \quad (4) \]

This tends to \( 2\lambda_0^2 \) for large \( \ell/\lambda_0 \), that is, to a value \( e^2/2 \sim 4 \) times larger than what we found in Eq. (3) for the local Metropolis algorithm. This factor corresponds to the efficiency gain we might expect for a generic event-chain algorithm with large \( \ell/\lambda_0 \), even though we will obtain considerably larger factors for the SEC algorithm.

In a finite system, the expressions in Eq. (4) must be corrected for the center-of-mass displacement. For the SEC algorithm, the corrected mean-square displacement per particle, \( D^\text{soc}(\ell) \), drops to zero for \( \ell/\lambda_0 \to \infty \) because in that limit, for a finite box, all disks participate in the chain, and the system ends up moving as a solid block. The REC algorithm, in contrast, saturates to a constant mean-square displacement per particle for large chains.

To further analyze the event-chain algorithms, we determined the auto-correlation time of the orientational order parameter \( \Psi \) for hard-disk systems at densities near the melting transition. The orientational order parameter \( \Psi \) averages the complex-valued local bond order \( \psi_j \) for each disk \( j \), where

\[ \Psi = 1/N \sum_j \psi_j \quad (5) \]

and

\[ \psi_j = 1/n_j \sum_{k,j} \exp(i6\varphi_{j,k}). \quad (6) \]

In Eq. (6), the sum is over the \( n_j \) neighbors of \( j \), and \( \varphi_{j,k} \) is the angle of the shortest vector equivalent to \( \mathbf{x}_k - \mathbf{x}_j \) [14]. Probable values of \( \Psi \) form an irregular ring around the origin (see the scatter plot in Fig. 3), \( \Psi \to \Psi + \pi \) symmetry in a square box imposes \( (\Psi(t)) = 0 \).

We suppose that the correlation time of this system is proportional to the time the order parameter takes to wander around the ring, that is, the auto-correlation time of \( \Psi \). This global measure of the overall speed of a Monte Carlo simulation is more appropriate than, for example, single-particle diffusion constants. However \( \Psi \) is very long to decorrelate at the interesting densities (see Fig. 4), and we have to limit ourselves to small systems. The auto-correlation function \( C(\Delta t) \) of the orientational order parameter is defined by the ensemble average

\[ C(\Delta t) \propto \langle \Psi(t)\Psi^*(t + \Delta t) \rangle, \]

normalized so that \( C(0) = 1 \). In our square box, this function decays to zero exponentially for large \( \Delta t \) (see Fig. 5), and the decay constant \( \tau \) and the speed \( 1/\tau \) are obtained by a fit, for each value of the parameters \( (N,\eta,\ell) \), from one single very long simulation (with \( t \gg \tau \)). The local Metropolis algorithm, for its optimal step size, is as fast as the event-chain algorithms with \( \ell/\lambda_0 \approx 1 \). (Our implementation moves \( 3 \times 10^{10} \) particles per hour on a 2.8GHz single-processor workstation for \( N = 4096 \).

For small total displacements \( \ell/\lambda_0 \ll 1 \), the speed of all the algorithms (reversible and irreversible SEC, REC, and local Metropolis algorithm) is proportional to \( D^\text{loc}_\text{loc}(\ell) \), as given by Eq. (4), that is, they all follow the single-particle behavior (see Fig. 6). For larger \( \ell/\lambda_0 \), the event-chain algorithms realize a considerable speed-up compared to the local Metropolis algorithm (also in Fig. 6). Moreover, both versions of the SEC algorithm set up coherent motion across the system and are clearly better than the REC algorithm, whose particle chains meander through the system (as shown in Fig. 2), so that the disks move incoherently. For large \( \ell/\lambda_0 \), the irreversible
SEC algorithm is faster than the reversible version: it is of advantage to break detailed balance. Figure 6 also illustrates that the SEC algorithm becomes more efficient (as compared to the local Metropolis algorithm) as one approaches the transition from the liquid phase (at density $\eta \sim 0.708$). The optimal speed-up increases with the system size, as shown in Table I. This suggests that the speed-up of the SEC algorithm may well increase with the correlation length of the system, and may, in the transition region, have a more favorable scaling than the local Metropolis algorithm.

Let us finally discuss the relationship between the Monte Carlo method and the molecular-dynamics algorithm. All these approaches describe the same equilibrium state. Unlike the Monte Carlo method, the molecular dynamics follows the physical time-evolution of the system. The first implementations of the molecular dynamics algorithm were very time consuming, with a complexity of $O(N)$ per event (collision), slower than the Metropolis algorithm ($O(1)$ per move). The complexity of modern implementations has improved to $O(\log N)$ per event and even $O(1)$. A closer look is thus needed to choose between the two methods.

FIG. 6: **Left:** Efficiency of SEC and REC algorithms for 1024 disks at $\eta = 0.71$ (all speeds normalized by the speed of the reversible SEC algorithm at $\ell/\lambda_0 = 1$). The speed of the local Metropolis algorithm and the mean-square displacement per particle from Eq. (3) are also shown. **Right:** Density dependence of the optimal speed-up factor.

| $N$  | Optimal speed-up | Reversible | Irreversible |
|------|------------------|-------------|--------------|
| 64   | $\sim 6$         | $\sim 8$    |              |
| 256  | $\sim 8$         | $\sim 11$   |              |
| 1024 | $\sim 9$         | $\sim 15$   |              |
| 4096 | $\sim 10$        | $\sim 20$   |              |

TABLE I: Optimal speed-up reached by the SEC algorithm (with respect to the reversible SEC algorithm for $\ell/\lambda_0 = 1$) at density $\eta = 0.71$ as a function of particle number.

FIG. 7: System-size dependence of the correlation time of the orientational order parameter for two densities. What can be achieved in approximately one hour with our implementation of the Metropolis algorithm, irreversible SEC, and the implementation of the molecular dynamics algorithm of [15].

In conclusion, we have in this paper proposed a class of algorithms for hard spheres and related systems, which clearly outperform the local Metropolis algorithm. We discussed three aspects of our algorithms, which all contributed to improve their speed. First, we showed that event-chain algorithms have a larger effective step size than the local Metropolis algorithm, because spheres move until they strike one of their neighbors. We computed mean-square displacements per particle (local diff-
fusion constants) to quantify this point. Nevertheless, local diffusion constants are not clearly related to the speed of the algorithm: they merely describe the short-time rattling of a particle in its cage (only for small $\ell/\lambda_0$ is the local diffusion constant directly proportional to the algorithm’s speed). Second, we performed numerical simulations of two variants of the method, and carefully analyzed the auto-correlation function of the orientational order parameter. One of them, the SEC algorithm, induces coherent motion of a long chain of spheres, and it allows the different parts of the system to communicate with each other. We witnessed considerable performance gains of this algorithm in the critical region, in the same way than the molecular dynamics. This suggests the exciting possibility that the speed-up of the event-chain algorithm grows with the correlation length of the system, and may have a more favorable scaling than the local Metropolis algorithm in the critical region. This speed-up, which is shared by both the molecular-dynamics algorithm and the SEC algorithm, is not understood and should be further investigated. Third, we noticed that the absence of rejections permitted to conceive an irreversible version of the SEC algorithm which improves the performances.

Our implementation of the SEC algorithm approaches equilibrium (for large systems at $\eta \approx 0.70$) about 40 times faster than our local Metropolis algorithm, not only because of the speed-up evidenced in Fig. 6 but also because the $xy$ version of the algorithm computes no scalar products and uses very few random numbers. It also equilibrates about five times faster than the best molecular-dynamics implementation and preserves certainly a large potential for improvement.

Nevertheless, CPU times needed for convergence remain extremely large, and even with our algorithm, full convergence of systems with $10^6$ particles at high densities comes barely into reach. The irreversible SEC algorithm not only appears to be the fastest currently known simulation method for dense hard-disk and hard-sphere systems, but it also provides a telling example of the benefits of breaking detailed balance in Monte Carlo algorithms going beyond the “lifting” Markov chains.

URL: [http://dbwilson.com](http://dbwilson.com)

[1] B. J. Alder and T. E. Wainwright, Phase transition in elastic disks, *Phys. Rev.* **127**, 359–361 (1962).
[2] B. J. Alder and T. E. Wainwright, Decay of the velocity autocorrelation function, *Phys. Rev. A* **1**, 18–21 (1970).
[3] N. Metropolis et al., Equation of state calculations by fast computing machines, *J. Chem. Phys.* **21**, 1087 (1953).
[4] B. J. Alder and T. E. Wainwright, Phase transition for a hard-sphere system, *J. Chem. Phys.* **27**, 1208–1209 (1957).
[5] R. H. Swendsen, J. S. Wang, Nonuniversal critical-dynamics in Monte-Carlo simulations, *Phys. Rev. Lett.* **58**, 86–88 (1987).
[6] U. Wolff, Collective Monte-Carlo updating for spin systems, *Phys. Rev. Lett.* **62**, 361–364 (1989).
[7] C. H. Mak, Large-scale simulations of the two-dimensional melting of hard disks, *Phys. Rev. E* **73**, 065104 (2006).
[8] B. I. Halperin and D. R. Nelson, Theory of 2-dimensional melting, *Phys. Rev. Lett.* **41**, 121 (1978); D. R. Nelson and B. I. Halperin, Dislocation-mediated melting in 2 dimensions, *Phys. Rev. B* **19**, 2457 (1979); A. P. Young, Melting and the vector Coulomb gas in 2 dimensions, *Phys. Rev. B* **19**, 1855 (1979).
[9] W. Krauth, Statistical Mechanics: Algorithms and Computations (Oxford University Press, Oxford, U.K., 2006).
[10] C. Dress and W. Krauth, Cluster Algorithm for hard spheres and related systems, *J. Phys. A* **28**, L597 (1995).
[11] A. Buhot and W. Krauth, Numerical solution of hard-core mixtures, *Phys. Rev. Lett.* **80**, 3787 (1998).
[12] L. Santen and W. Krauth, Absence of thermodynamic phase transition in a model glass former, *Nature* **405**, 550–551 (2000).
[13] A. Jaster, An improved Metropolis algorithm for hard-core systems, *Physica A* **264**, 134 (1999); A. Jaster, The hexatic phase of the two-dimensional hard disks system, *Phys. Lett. A*, 330, 120 (2004).
[14] K. J. Strandburg, Two-dimensional melting, *Rev. Mod. Phys.* **60**, 161 (1988).
[15] M. Isobe, Simple and Efficient Algorithm for Large Scale Molecular Dynamics Simulation in Hard Disk System, *International Journal of Modern Physics C* **10**, 1281 (1999); Long-time tail of the velocity autocorrelation function in a two-dimensional moderately dense hard-disk fluid, *Phys. Rev. E* **77**, 021201 (2008).
[16] G. Paul, A Complexity O(1) Priority Queue for Event Driven Molecular Dynamics Simulations, *J. Comp. Phys.* **221**, 615–625 (2007).
[17] M. Isobe, private communication (2009).
[18] P. Diaconis, S. Holmes, and R. M. Neal, Analysis of a nonreversible Markov chain sampler, *Ann. Applied Probab.* **10**, 726 (2000).