

## Event-chain Monte Carlo algorithms for hard-sphere systems

Etienne P. Bernard,<sup>1,\*</sup> Werner Krauth,<sup>1,†</sup> and David B. Wilson<sup>2,‡</sup><sup>1</sup>CNRS–Laboratoire de Physique Statistique, Ecole Normale Supérieure, 24 rue Lhomond, 75231 Paris Cedex 05, France<sup>2</sup>Microsoft Research, One Microsoft Way, Redmond, Washington 98052, USA

(Received 19 March 2009; revised manuscript received 15 October 2009; published 18 November 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.

DOI: 10.1103/PhysRevE.80.056704

PACS number(s): 05.10.Ln, 05.20.Jj

Hard spheres in three and in two dimensions (hard disks) occupy a special place in statistical mechanics. Indeed, many fundamental concepts, from the virial expansion (by van der Waals and Boltzmann), to two-dimensional melting [1], to long-time tails [2], were first discussed in these extraordinarily rich physical systems. These models have also played a crucial role in the history of computation: both the Metropolis algorithm [3] and molecular dynamics [4] were first implemented for monodisperse hard disks in a box. In contrast with the spectacular algorithmic developments for lattice spin models [5,6], Monte Carlo algorithms for hard spheres have changed little since the 1950s, especially for high densities. Recent sophisticated implementation have reduced the complexity of the molecular-dynamics algorithm to a value comparable to that of the Monte Carlo method. Nevertheless, one can today still not equilibrate sufficiently large systems [7] to clarify whether the melting transition in two-dimensional hard disks, at density (occupied volume fraction)  $\eta \approx 0.70$ , is weakly first order, or whether it is of the Kosterlitz-Thouless type [8], with a narrow hexatic phase in between the liquid and the solid.

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 [9].

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 [10], for example, works extremely well for binary [11] or for polydisperse [12] mixtures, but it breaks down for the high densities of interest in two-dimensional melting. In Jaster’s algorithm [13], 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 Fig. 1). 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

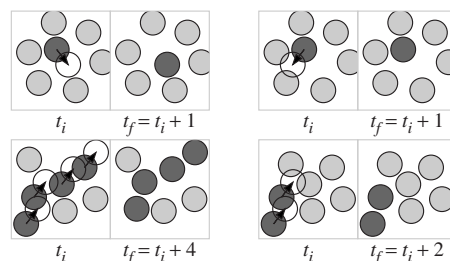


FIG. 1. Upper panels: Accepted (left) and rejected (right) local Metropolis moves of a disk in the cage formed by its neighbors. Lower panels: Accepted and rejected moves in Jaster’s chain algorithm.

\*etienne.bernard@ens.fr

†werner.krauth@ens.fr

‡http://dbwilson.com

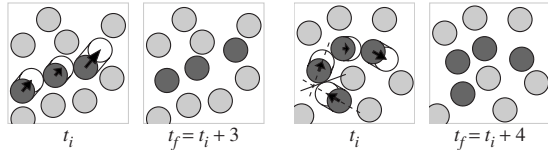


FIG. 2. *Left two panels*: Move of the straight event-chain (SEC) algorithm. The individual displacements add up to a distance  $\ell$ . *Right two panels*: Reflected event-chain (REC) move.

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  $[0, 2\pi]$ , both versions, which we call “reversible,” preserve the uniform measure because of detailed balance.

The detailed balance condition is allowed to be broken in the SEC algorithm. Indeed, for a given direction  $\theta$ , a configuration  $\Gamma$  of  $N$  disks can reach  $N$  other configurations in one move. By applying to  $\Gamma$  the  $N$  possible moves in direction  $\theta$ , one finds the  $N$  configurations that reach  $\Gamma$ . Therefore, the SEC algorithm satisfies global balance for any distribution of  $\theta$ . Algorithms breaking detailed balance induce probability flows in the configuration space and potentially speed-up equilibration [18]. We study such an irreversible version of the SEC algorithm where  $\theta$  is uniformly distributed in  $[0, \pi]$ . To preserve ergodicity, at least two independent directions are needed. By far our fastest implementation (the “xy version” of the SEC algorithm) alternates moves in the positive  $x$  and  $y$  directions ( $\theta=0, \pi/2$ ). A version of the SEC algorithm, but with rejections and which cannot break detailed balance, was also mentioned in [13].

In Fig. 3 we show the integrated distribution of  $|\Psi|$  of Eq. (5),

$$\int_0^x \pi(|\Psi|) d|\Psi|, \quad (1)$$

for the xy version of the SEC algorithm, for the Metropolis algorithm, and for molecular dynamics in the same system. They are found to be equal. This demonstrates the correctness of our implementations.

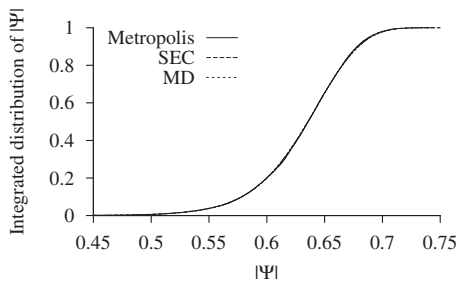


FIG. 3. Comparison of the integrated distribution of an observable [the absolute value of the order parameter  $\Psi$  of Eq. (5)] between the SEC algorithm which breaks detailed balance, molecular dynamics (MD) and the local Metropolis algorithm for 1024 disks at  $\eta=0.71$ .

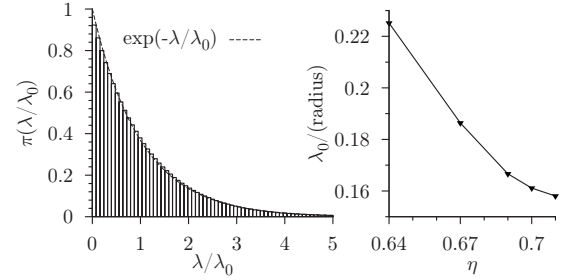


FIG. 4. *Left*: Histogram  $\pi(\lambda/\lambda_0)$  of the free path  $\lambda(\theta)$  for 1024 disks at density  $\eta=0.71$ . The distribution is close to exponential even in the solid phase. *Right*: Mean-free path  $\lambda_0$  in units of the disk radius as a function of  $\eta$ .

As a first step to analyze the performance of the event-chain algorithms, we consider the mean-square displacement  $\langle \Delta_{\mathbf{x}}^2 \rangle$  of individual disks, both in the local Metropolis and in the event-chain algorithms. As mentioned, event-chain algorithms generically outperform the Metropolis algorithm in part because they take larger steps on average. In order to compare the two methods, we measure time in units of attempted one-particle displacements.

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

$$\pi(\lambda/\lambda_0) \simeq \exp(-\lambda/\lambda_0), \quad (2)$$

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, 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  $\langle \Delta_{\mathbf{x}}^2(\ell) \rangle = \ell^2 \exp(-\ell/\lambda_0)$ , which is maximized when  $\ell=2\lambda_0$ ,

$$\max_{\ell} \langle \Delta_{\mathbf{x}}^2(\ell) \rangle = \langle \Delta_{\mathbf{x}}^2(2\lambda_0) \rangle = 4\lambda_0^2/e^2. \quad (3)$$

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 timelike 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,

$$\langle \Delta_{\mathbf{x}}^2(\ell) \rangle = \int_0^{\ell} \int_0^{\ell} ds ds' \pi(s, s').$$

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

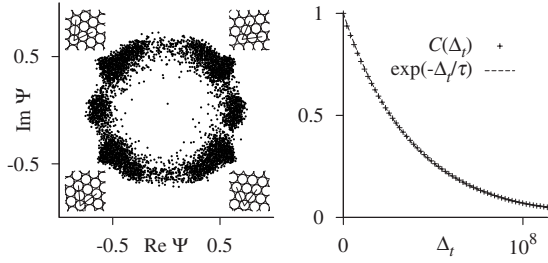


FIG. 5. *Left*: Order-parameter distribution for 256 disks in a periodic square box for  $\eta=0.71$ . *Right*: Correlation function  $C(\Delta_t)$  for this system. The correlation time is obtained from an exponential fit, as shown.

$$D_{\text{loc}}^{\infty}(\ell) = \frac{\langle \Delta_{\mathbf{x}}^2(\ell) \rangle}{\langle M(\ell) \rangle} = 2\lambda_0^2 \underbrace{\frac{\exp(-\ell/\lambda_0) + \ell/\lambda_0 - 1}{\ell/\lambda_0 + 1}}_{\rightarrow 1 \text{ for } \ell/\lambda_0 \rightarrow \infty} \quad (4)$$

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,  $D_{\text{loc}}(\ell)$ , drops to zero for  $\ell/\lambda_0 \rightarrow \infty$  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 autocorrelation time of the orientational order parameter  $\Psi$  [14] 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

$$\psi_j = \frac{1}{n_j} \sum_{\langle k,j \rangle} \exp(i6\varphi_{j,k}). \quad (6)$$

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. 5; the  $\Psi \rightarrow \Psi + \pi$  symmetry in a square box imposes  $\langle \Psi(t) \rangle = 0$ ].

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 autocorrelation 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. 7), and we have to limit

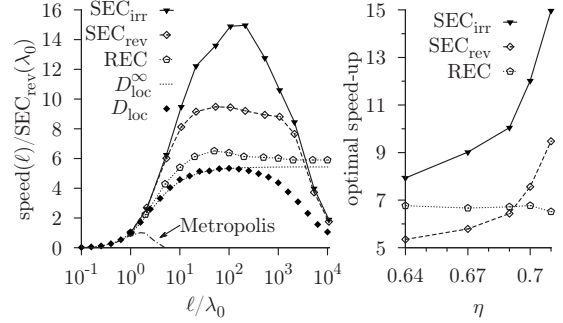


FIG. 6. *Left*: Efficiency of SEC and REC algorithms for 1024 disks at  $\eta=0.71$  (all speeds normalized by the speed of the reversible SEC algorithm at  $\ell/\lambda_0=1$ ). The speed of the local Metropolis algorithm and the mean-square displacement per particle from Eq. (4) are also shown. *Right*: Density dependence of the optimal speed-up factor.

ourselves to small systems. The autocorrelation 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 2.8 GHz 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  $D_{\text{loc}}^{\infty}(\ell)$ , 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 implementa-

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.

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

tions of the molecular-dynamics algorithm [4] 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  $O(\log N)$  [15] per event and even  $O(1)$  [16]. A closer look is thus needed to choose between the two methods.

We used a simple version of the molecular dynamics to compute the decorrelation time of  $\Psi$  in the same way as in Fig. 5. In number of events, molecular dynamics is found to be about three times faster than the irreversible version of SEC for  $\eta \sim 0.7$  and  $N=64-1024$ . It is very interesting to notice that molecular dynamics shows, unlike REC, the same density dependence of its speed as SEC around the transition region. We then determined the CPU time per collision of one of the most rapid current implementations of the hard-disk molecular-dynamics algorithm [15]. For the  $32 \times 32$  system at  $\eta \sim 0.7$ , this implementation reaches about  $1.7 \times 10^9$  collisions per hour on a 2.6 GHz workstation [17]. Our  $xy$  implementation of the SEC algorithm reaches  $3 \times 10^{10}$  collisions per hour on similar hardware. Our implementation is thus about 5 times faster in CPU time to reach thermodynamic equilibrium than the best molecular-dynamics implementation. We should also note that SEC is much easier to implement. A synopsis of these relative and absolute timing issues is presented in Fig. 7. For clarity, we give times in terms of “equivalent Metropolis moves;” this means that one event of the molecular-dynamics algorithm corresponds to  $\sim 3$  SEC events and to  $\sim 60$  Metropolis moves. The horizontal lines indicate 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 diffusion 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

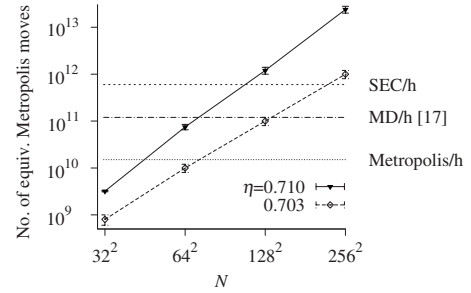


FIG. 7. System-size dependence of the correlation time of the orientational order parameter for two densities. What can be achieved in approximately one hour using the different algorithms discussed in this paper is indicated by horizontal lines.

$\ell/\lambda_0$  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 autocorrelation 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 [18].

This research was initiated at the Kavli Institute for Theoretical Physics at UCSB (URL: <http://www.kitp.ucsb.edu/>). We thank the NSF for partial support under Grant No. PHY05-51164. We are grateful to M. Isobe for helpful correspondence.



- [1] B. J. Alder and T. E. Wainwright, *Phys. Rev.* **127**, 359 (1962).
- [2] B. J. Alder and T. E. Wainwright, *Phys. Rev. A* **1**, 18 (1970).
- [3] N. Metropolis *et al.*, *J. Chem. Phys.* **21**, 1087 (1953).
- [4] B. J. Alder and T. E. Wainwright, *J. Chem. Phys.* **27**, 1208 (1957).
- [5] R. H. Swendsen and J. S. Wang, *Phys. Rev. Lett.* **58**, 86 (1987).
- [6] U. Wolff, *Phys. Rev. Lett.* **62**, 361 (1989).
- [7] C. Mak, *Phys. Rev. E* **73**, 065104 (2006).
- [8] B. I. Halperin and D. R. Nelson, *Phys. Rev. Lett.* **41**, 121 (1978); D. R. Nelson and B. I. Halperin, *Phys. Rev. B* **19**, 2457 (1979); A. P. Young, *ibid.* **19**, 1855 (1979).
- [9] W. Krauth, *Statistical Mechanics: Algorithms and Computations* (Oxford University Press, Oxford, U.K., 2006).
- [10] C. Dress and W. Krauth, *J. Phys. A* **28**, L597 (1995).
- [11] A. Buhot and W. Krauth, *Phys. Rev. Lett.* **80**, 3787 (1998).
- [12] L. Santen and W. Krauth, *Nature (London)* **405**, 550 (2000).
- [13] A. Jaster, *Physica A* **264**, 134 (1999); *Phys. Lett. A* **330**, 120 (2004).
- [14] K. J. Strandburg, *Rev. Mod. Phys.* **60**, 161 (1988).
- [15] M. Isobe, *Int. J. Mod. Phys. C* **10**, 1281 (1999); *Phys. Rev. E* **77**, 021201 (2008).
- [16] G. Paul, *J. Comput. Phys.* **221**, 615 (2007).
- [17] M. Isobe (private communications).
- [18] P. Diaconis, S. Holmes, and R. M. Neal, *Ann. Appl. Probab.* **10**, 726 (2000).