

PERFECT SIMULATION OF THE HARD DISKS MODEL BY PARTIAL REJECTION SAMPLING

HENG GUO AND MARK JERRUM

ABSTRACT. We present a perfect simulation of the hard disks model via the partial rejection sampling method. Provided the density of disks is not too high, the method produces exact samples in $O(\log n)$ rounds, and total time $O(n)$, where n is the expected number of disks. The method extends easily to the hard spheres model in $d > 2$ dimensions. In order to apply the partial rejection method to this continuous setting, we provide an alternative perspective of its correctness and run-time analysis that is valid for general state spaces.

1. INTRODUCTION

The *hard disks model* is one of the simplest gas models in statistical physics. Its configurations are non-overlapping disks of uniform radius r in a bounded region of \mathbb{R}^2 . For convenience, in this paper, we take this region to be the unit square $[0, 1]^2$. This model was precisely the one studied by Metropolis et al. [14], in their pioneering work on the Markov chain Monte Carlo (MCMC) method. They used Los Alamos' MANIAC computer to simulate a system with 224 disks.

There are two variants of this model. To obtain the *canonical ensemble*, we fix the number (or equivalently, density) of disks and decree that all configurations are “equally likely”, subject only to the disks not overlapping. In the *grand canonical ensemble*, we fix the “average” number of disks. To be more specific, centers of the disks are distributed according to a Poisson point process of intensity $\lambda_r = \lambda/(\pi r^2)$, conditioned on the disks being non-overlapping. The hard disks model, and its higher dimensional generalization (called the *hard spheres model*) are also related to the optimal sphere packing density [6, 20, 2]. See [8, 1] and references therein for more details. See also [13] for the physics perspective.

Our main aim in this work is to describe and analyse a very simple algorithm for exactly sampling from the grand canonical ensemble, based on the partial rejection sampling paradigm introduced by Guo, Jerrum and Liu [5].

More precisely, the challenge is the following: produce a realisation $P \subset [0, 1]^2$ of a Poisson point process of intensity λ_r in the unit square, conditioned on the event that no pair of points in P are closer than $2r$ in Euclidean distance. We refer to this target measure as the *hard disks distribution*. It describes an arrangement of open disks of radius r with centres in $[0, 1]^2$ that are not allowed to overlap, but which otherwise do not interact. It is a special case of the Strauss process [19]. Note that, although the disks do not overlap each other, they may extend beyond the boundary of the unit square. Also, the intensity of the underlying Poisson process is normalised so that the expected number of points of P lying in a disk of radius r is λ . This normalisation gives us sensible asymptotics as the radius of the disks tends to zero (equivalently, the number of disks tends to infinity).

Classical rejection sampling applied to this problem yields the following algorithm: repeatedly sample a realisation P of the Poisson process of intensity λ in the unit square until P satisfies the condition that no two points are closer than $2r$, and return P . Unfortunately, for every $\lambda > 0$, however small, the expected number of unsuccessful trials using this approach increases exponentially in r^{-1} , as $r \rightarrow 0$. Partial rejection sampling [5] requires only a subset of P to be resampled at each iteration. Algorithm 1 below arises from a routine application of the paradigm to the problem at hand.

The work described here was supported by the EPSRC research grant EP/N004221/1 “Algorithms that Count”.

The original partial rejection method [5] and its analysis are tailored for the discrete case. In this paper we provide an alternative view on the correctness of the method, which is also valid in the continuous setting. In other words, as with classical rejection sampling, Algorithm 1 terminates with probability 1, producing a realisation of the exact hard disks distribution.

Theorem 1. *Algorithm 1 is correct: conditional on halting, Algorithm 1 produces a sample from the hard disks distribution with intensity $\lambda_r = \lambda/(\pi r^2)$.*

The proof of this result forms the content of Section 3.

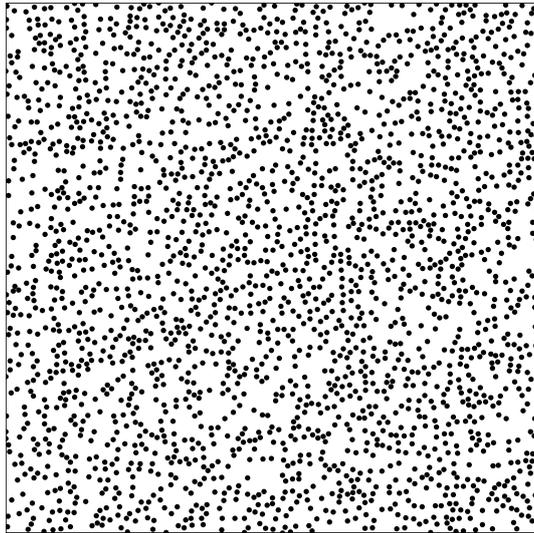


FIGURE 1. A realisation of the hard disks measure with $\lambda = 0.5$ and $r = \frac{1}{200}$. The resulting density is $\alpha = 0.189+$.

In contrast to classical rejection sampling, the expected number of iterations (resampling steps) is now asymptotically $O(\log(r^{-1}))$ as $r \rightarrow 0$, provided λ is not too large. Furthermore, with a suitable implementation, the total runtime is $O(r^{-2})$, i.e., linear in the number of disks. We prove that rapid termination occurs when $\lambda < 0.21027$. This analysis is not tight, and experiments suggests that the actual threshold for rapid termination is around $\lambda \approx 0.5$. Figure 1 is a realisation of $\lambda = 0.5$ with $r = \frac{1}{200}$. The resulting density is $\alpha = 0.189+$.

Theorem 2. *Fix $\lambda < 0.21027$. Then the expected number of iterations of the while-loop in Algorithm 1 is $O(\log r^{-1})$. Moreover, with a suitable implementation, the overall runtime of the algorithm is $O(r^{-2})$.*

The proof of this result forms the content of Section 4.

The method extends naturally to the *hard spheres model* in $d > 2$ dimensions. Here, the desired distribution is a Poisson point process in $[0, 1]^d$ conditioned on no pair of points being closer than $2r$. The natural normalisation for the intensity of the Poisson process in d dimensions is $\lambda_{r,d} = \lambda/(v_d r^d)$, where v_d is the volume of a ball of radius 1 in \mathbb{R}^d . With this convention, we prove that rapid termination occurs in d dimensions provided $\lambda < 2^{-(d+\frac{1}{2})}$.

The *expected packing density* $\alpha(\lambda)$ or simply α for this model is the expected total volume of spheres. (Note that, neglecting boundary effects, α is the proportion of the unit cube occupied by spheres.) The quantity $\alpha(\lambda)$ grows monotonically with λ , but intuitively we expect its rate of growth to slow down dramatically as the spheres pack more tightly. The connection between expected packing density α and intensity λ has recently been thoroughly explored by Jenssen, Joos and Perkins [8]. Using their results, we show that partial rejection sampling can achieve expected packing density $\Omega(2^{-d})$ while retaining rapid termination in $O(\log(r^{-1}))$ iterations. Although sphere packings of density $\Omega(d2^{-d})$ have been proved to exist, there is no polynomial-time sampler that provably achieves packing density beyond $O(2^{-d})$, as far as we are aware.

Other approaches to exact sampling include Coupling From The Past (CFTP), which was adapted to point processes by Kendall [10] and Kendall and Møller [11]. Recently, Moka, Juneja and Mandjes [15] proposed an algorithm based on rejection and importance sampling. Although their algorithm, like ours, is based on rejection sampling, it does not share our asymptotic performance guarantee. Indeed, its runtime appears to grow exponentially as the number of disks goes to infinity, with the density of disks held constant.

The most widely used approach to sampling configurations from the hard disks model is Markov chain Monte Carlo (MCMC). Here the desired distribution is approached in the limit as the runtime t of the sampler tends to infinity. In this sense, MCMC produces an approximate sampler, though the error (in total variation distance) decays exponentially with t . The problem lies in deciding how large t should be in order to ensure that the samples obtained are close enough to the desired distribution. There are two possibilities. The runtime t may be chosen heuristically, in which case the quality of the output from the sampler is not guaranteed. Otherwise, an analytical upper bound on mixing time may be used to determine a suitable t , but then the tendency is for this bound to be very conservative. The experimental advantage of partial rejection sampling is its simple termination rule, combined with the property of generating perfect samples from the desired distribution.

Approximate sampling via Markov chain simulation has been studied by Kannan, Mahoney and Montenegro [9] and Hayes and Moore [7] in the context of the canonical ensemble, where the number of spheres in a configuration is fixed. (So in the MCMC approach, it is the density α that is chosen in advance, while in the approach taken here we choose λ in advance and then α follows.) Kannan et al. [9] show that rapid mixing (convergence to near-stationarity in time polynomial in the number of disks) occurs for densities $\alpha \leq 2^{-(d+1)}$, in dimension d . The best rigorously derived density bound guaranteeing rapid mixing in two dimensions is given by Hayes and Moore [7] and is $\alpha \approx 0.154$. (Note that this improves on the $\alpha = \frac{1}{8}$ obtained by Kannan et al.) It should be noted that these results are not directly comparable with our $\lambda < 0.21027$ owing to the difference in models. To obtain canonical ensembles, we could use Algorithm 1 and further condition on the number of desired disks. However, the only rigorous guarantee for this approach, via [8], is $\alpha(0.21027) > 0.0887$.

It is believed that the hard disks model in two dimensions undergoes a phase transition at a certain critical density α_c : at lower densities configurations are disordered, while at higher densities, long range correlations can be observed. Unfortunately, α_c is well beyond the densities that can be achieved by samplers (either based on MCMC or rejection sampling) with rigorous performance guarantees. However, heuristic approaches using sophisticated MCMC samplers [3] suggest that the critical density is around $\alpha_c \approx 0.7$.

Finally, determining the maximum achievable density α is the classical sphere packing problem. Despite extensive study, the exact solution is known only for dimensions $d = 1, 2, 3, 8$ and 24. See [8, 16] for rigorous bounds on packing densities in general dimension d .

A preliminary version of this paper was presented at ICALP 2018 [4]. Subsequently, the constant in Theorem 2 has been improved by Jake Wellens [21] to 0.2344+.

2. THE SAMPLING ALGORITHM

The following notation will be used throughout. If P is a finite subset of $[0, 1]^2$ then

$$\text{BadPairs}(P) = \{\{x, y\} : x, y \in P \wedge x \neq y \wedge \|x - y\| < 2r\},$$

where $\|\cdot\|$ denotes Euclidean norm, and

$$\text{BadPoints}(P) = \bigcup \text{BadPairs}(P).$$

The open disk of radius r with centre $x \in [0, 1]^2$ is denoted by $D_r(x)$. The finite set $\Pi \subset [0, 1]^2$ always denotes a realisation of the Poisson point process of intensity λ_r on $[0, 1]^2$. For a random variable X and event \mathcal{E} we use $\mathcal{D}(X)$ to denote the distribution (law) of X , and $\mathcal{D}(X | \mathcal{E})$ the distribution of X conditioned on \mathcal{E} occurring. Thus, $\mathcal{D}(\Pi | \text{BadPoints}(\Pi) = \emptyset)$ is the hard disks distribution that we are interested in.

Our goal is to analyse the correctness and runtime of a sampling algorithm for the hard disks model (see Algorithm 1 below), specifically to determine the largest value of λ for which it terminates quickly, i.e., in $O(\log r^{-1})$ iterations. The algorithm is an example application of “Partial Rejection Sampling” [5], adapted to the continuous state space setting.

Algorithm 1 Partial Rejection Sampling for the hard disks model

```

PRS for Hard Disks( $\lambda, r$ ) //  $r$  is the disk radius and  $\lambda_r = \lambda/(\pi r^2)$  the intensity
Let  $P$  be a sample from the Poisson point process of intensity  $\lambda_r$  on the unit square
while  $B \leftarrow \text{BadPoints}(P) \neq \emptyset$  do
   $S \leftarrow B + D_{2r}(\mathbf{0})$  // Resampling set is the Minkowski sum of  $B$  with a disk of radius  $2r$ 
  Let  $P^S$  be a sample from the Poisson point process of intensity  $\lambda_r$  on  $S$ 
   $P \leftarrow (P \setminus B) \cup P^S$ 
end while
return  $P$ 

```

3. CORRECTNESS (PROOF OF THEOREM 1)

Let B be any finite subset of $[0, 1]^2$. We say that B is a *feasible set* of bad points if $\text{BadPoints}(B) = B$; this is equivalent to saying that there is a finite subset $R \subset [0, 1]^2$ with $B = \text{BadPoints}(R)$. The key to establishing correctness of Algorithm 1 is the following loop invariant:

$$\mathcal{D}(P \mid \text{BadPoints}(P) = B) = \mathcal{D}(\Pi \mid \text{BadPoints}(\Pi) = B),$$

for every feasible set B , where P is any intermediate set of points during the execution of the algorithm, and Π is a realisation of the Poisson point process. Let us consider what the right hand side means operationally. Let $S = B + D_{2r}(\mathbf{0})$. (This is the resampling set used by the algorithm.) Let Q be a sample from the distribution $\mathcal{D}(\Pi \mid \text{BadPoints}(\Pi) = B)$. The only points in Q that lie inside S are the points in B . (Any extra points would create more bad pairs than there actually are.) Thus $Q \cap S = B$. Outside of S there are no bad pairs; thus $Q \cap \bar{S}$ is a sample from the hard disks distribution on $\bar{S} = [0, 1]^2 \setminus S$. Note that, setting $B = \emptyset$, we see that $\mathcal{D}(\Pi \mid \text{BadPoints}(\Pi) = \emptyset)$ is just the hard disks distribution on $[0, 1]^2$.

Let T (a random variable) be the number of iterations of the while-loop. On each iteration, the while loop terminates with probability bounded away from 0; thus T is finite with probability 1. (Indeed, T has finite expectation.) Let P_t , for $1 \leq t \leq T$, be the point set P after t iterations of the loop, and P_0 be the initial value of P (which is just a realisation of the Poisson point process on $[0, 1]^2$). We say that $B_0, B_1, \dots, B_t \subset [0, 1]^2$ is a *feasible sequence* of (finite) point sets if there exists a run of Algorithm 1 with $\text{BadPoints}(P_0) = B_0, \dots, \text{BadPoints}(P_t) = B_t$. Theorem 1 will follow easily from the following lemma.

Lemma 3. *Let $B_0, B_1, \dots, B_t \subset [0, 1]^2$ be a feasible sequence. Then*

$$\begin{aligned} \mathcal{D}(P_t \mid \text{BadPoints}(P_0) = B_0 \wedge \dots \wedge \text{BadPoints}(P_t) = B_t) &= \mathcal{D}(P_t \mid \text{BadPoints}(P_t) = B_t) \\ &= \mathcal{D}(\Pi \mid \text{BadPoints}(\Pi) = B_t). \end{aligned}$$

Proof. We prove the result by induction on t . The base case, $t = 0$, holds by construction: P_0 is just a realisation of the Poisson point process on $[0, 1]^2$. Our induction hypothesis is

$$(1) \quad \mathcal{D}(P_t \mid \text{BadPoints}(P_0) = B_0 \wedge \dots \wedge \text{BadPoints}(P_t) = B_t) = \mathcal{D}(\Pi \mid \text{BadPoints}(\Pi) = B_t),$$

for every feasible sequence B_0, \dots, B_t . Extend the feasible sequence to B_{t+1} . For the inductive step, we assume (1) and aim to derive

$$(2) \quad \begin{aligned} \mathcal{D}(P_{t+1} \mid \text{BadPoints}(P_0) = B_0 \wedge \dots \wedge \text{BadPoints}(P_{t+1}) = B_{t+1}) \\ = \mathcal{D}(\Pi \mid \text{BadPoints}(\Pi) = B_{t+1}). \end{aligned}$$

The resampling set on iteration $t + 1$ is $S = B_t + D_{2r}(\mathbf{0})$. As a first step we argue below that

$$(3) \quad \mathcal{D}(P_{t+1} \mid \text{BadPoints}(P_0) = B_0 \wedge \dots \wedge \text{BadPoints}(P_t) = B_t) = \mathcal{D}(\Pi \mid \text{BadPairs}(\Pi) \cap \bar{S}^{(2)} = \emptyset),$$

where $\bar{S}^{(2)}$ denotes the set of unordered pairs of elements from \bar{S} . We have noted that (1) implies that, outside of the resampling set S , the point set P_t is a realisation of the hard disks distribution. Also, the algorithm does not resample points outside of S . Thus $P_{t+1} \cap \bar{S} = P_t \cap \bar{S}$ is Poisson distributed, conditioned on there being no bad pairs. Inside S , resampling has left behind a fresh Poisson point process $P_{t+1} \cap S$. These considerations give (3).

Next, we condition on B_{t+1} :

$$\begin{aligned} \mathcal{D}(P_{t+1} \mid \text{BadPoints}(P_0) = B_0 \wedge \cdots \wedge \text{BadPoints}(P_t) = B_t \wedge \text{BadPoints}(P_{t+1}) = B_{t+1}) \\ = \mathcal{D}(\Pi \mid \text{BadPairs}(\Pi) \cap \bar{S}^{(2)} = \emptyset \wedge \text{BadPoints}(\Pi) = B_{t+1}). \end{aligned}$$

Since B_{t+1} contains no bad pairs with both endpoints in \bar{S} , the event $\text{BadPoints}(\Pi) = B_{t+1}$ entails the event $\text{BadPairs}(\Pi) \cap \bar{S}^{(2)} = \emptyset$. Thus, we have

$$\begin{aligned} \mathcal{D}(P_{t+1} \mid \text{BadPoints}(P_0) = B_0 \wedge \cdots \wedge \text{BadPoints}(P_t) = B_t \wedge \text{BadPoints}(P_{t+1}) = B_{t+1}) \\ = \mathcal{D}(\Pi \mid \text{BadPoints}(\Pi) = B_{t+1}). \end{aligned}$$

The right hand side of this equation does not involve B_0, \dots, B_t , and so

$$\mathcal{D}(P_{t+1} \mid \text{BadPoints}(P_{t+1}) = B_{t+1}) = \mathcal{D}(\Pi \mid \text{BadPoints}(\Pi) = B_{t+1}).$$

This completes the induction step (2) and the proof. \square

We can now complete the proof of Theorem 1. As we observed earlier, T , the number of iterations of the while-loop, is finite with probability 1. By Lemma 3, noting that $B_T = \emptyset$,

$$\mathcal{D}(P_T) = \mathcal{D}(\Pi \mid \text{BadPoints}(\Pi) = \emptyset).$$

In other words, at termination, Algorithm 1 produces a realisation of the hard disks distribution on $[0, 1]^2$.

4. RUN-TIME ANALYSIS (PROOF OF THEOREM 2)

We consider how the number of ‘‘bad events’’, i.e., the cardinality of the set $\text{BadPairs}(P_t)$, evolves with time. As usual Π denotes a realisation of the Poisson point process of intensity λ_r . Also denote by Δ a realisation of the hard disks process of the same intensity. We need the following stochastic domination result.

Lemma 4. *The hard disks distribution is stochastically dominated by the Poisson point process with the same intensity. That is, we can construct a joint sample space for Π and Δ such that $\Delta \subseteq \Pi$.*

Holley’s criterion is a useful test for stochastic domination, but it is not of direct use to us in the proof of Lemma 4, because it applies only to finite state spaces. Fortunately, Preston [17, Theorem 9.1], has derived a version of Holley’s criterion that fits our situation. We will mostly follow Preston’s notation, except that, to save confusion, we will use P and Q , rather than x and y , to denote finite sets of points. In order to state his result, we need some notation. In our application, $\tilde{\omega}_n$ is the distribution on $([0, 1]^2)^{(n)}$ obtained by sampling n points independently and uniformly at random from $[0, 1]^2$, and regarding the points as indistinguishable; furthermore, $\tilde{\omega} = \sum_{n=0}^{\infty} \tilde{\omega}_n/n!$. (For consistency with Preston, we have left $\tilde{\omega}$ unnormalised. If we had made it into a probability distribution by division by e , then $\tilde{\omega}$ could be thought of as follows: sample an integer k from the Poisson distribution with mean 1, and then pick k (unlabelled) points uniformly and independently.) Denote by Ω the set of all finite subsets of $[0, 1]^2$, and by \mathcal{F} the set of non-negative measurable functions $\Omega \rightarrow \mathbb{R}$ satisfying

$$(4) \quad \int f d\tilde{\omega} = 1,$$

and

$$(5) \quad f(P) = 0 \text{ and } P \subseteq Q \text{ implies } f(Q) = 0.$$

(See Preston [17, Section 9] for detailed formal definitions of the concepts here.)

Lemma 5 (Theorem 9.1 of [17]). *Let $f_1, f_2 \in \mathcal{F}$ and suppose that*

$$(6) \quad \frac{f_1(P + \xi)}{f_1(P)} \geq \frac{f_2(Q + \xi)}{f_2(Q)}, \quad \text{for all } Q \subseteq P \in \Omega \text{ and } \xi \in [0, 1]^2 \setminus P$$

(where, by convention, $0/0 = 0$). Then, for all bounded, measurable, non-decreasing functions $g : \Omega \rightarrow \mathbb{R}$,

$$\int g f_1 d\tilde{\omega} \geq \int g f_2 d\tilde{\omega},$$

i.e., if μ_i is the probability measure having density f_i with respect to $\tilde{\omega}$, then μ_1 stochastically dominates μ_2 .

Proof of Lemma 4. We set

$$f_1(P) = C_1 \lambda_r^{|P|}$$

and

$$f_2(P) = \begin{cases} C_2 \lambda_r^{|P|}, & \text{if } \text{BadPairs}(P) = \emptyset; \\ 0, & \text{otherwise.} \end{cases}$$

The normalising constants C_1 and C_2 are chosen so that both $f = f_1$ and $f = f_2$ satisfy (4). (There is an explicit expression for C_1 , namely $C_1 = \exp(-\lambda_r)$, but not for C_2 .) Notice that both f_1 and f_2 also satisfy (5). Notice also that the probability measures μ_1 and μ_2 of the Poisson point process and the hard disks process have densities f_1 and f_2 with respect to $\tilde{\omega}$. The premise (6) of Lemma 5 holds, since the left-hand side is always λ_r and the right-hand side is either λ_r or 0. The conclusion is that μ_1 dominates μ_2 . Strassen's Theorem [12, 18], allows us to conclude the existence of a coupling of Π and Δ as advertised in the statement of the lemma (except, possibly, on a set of measure zero). \square

Lemma 6. *There is a bound $\bar{\lambda} > 0$ such that the expected number of iterations of the while-loop in Algorithm 1 is $O(\log r^{-1})$ when $\lambda < \bar{\lambda}$.*

Proof. First observe that $\text{BadPairs}(P)$ determines $\text{BadPoints}(P)$ and vice versa. So conditioning on the set $\text{BadPairs}(P)$ is equivalent to conditioning on $\text{BadPoints}(P)$.

Introduce random variables $Z_t = |\text{BadPairs}(P_t)|$, for $0 \leq t \leq T$. Our strategy is to show that

$$(7) \quad \mathbb{E}(Z_{t+1} \mid Z_0, \dots, Z_t) \leq \alpha^{-1} Z_t,$$

for some $\alpha > 1$. Then $Z_0, \alpha Z_1, \alpha^2 Z_2, \alpha^3 Z_3, \dots$ is a supermartingale (with the convention that $Z_t = 0$ for all $t > T$). Therefore, $\mathbb{E} Z_t \leq \alpha^{-t} \mathbb{E} Z_0 \leq \frac{1}{2} \lambda_r^2 \alpha^{-t}$. Here, we have used the fact that $|P_0|$ is a Poisson random variable with expectation λ_r , and

$$Z_0 = |\text{BadPairs}(P_0)| \leq \frac{1}{2} |P_0| (|P_0| - 1),$$

and hence

$$\mathbb{E} Z_0 \leq \mathbb{E} [|P_0| (|P_0| - 1)] = \frac{1}{2} \lambda_r^2.$$

Setting $t = O(\log r^{-1} + \log \varepsilon^{-1})$, we obtain $\mathbb{E} Z_t < 1/\varepsilon$, and hence $\Pr(T > t) \leq \varepsilon$. It follows that the expected number of iterations of the while-loop of Algorithm 1 is $O(\log r^{-1})$. Note that the probability of non-termination decreases exponentially with t , so the probability of large deviations above the expected value of T is low.

Crude estimates give $\bar{\lambda} = 1/(4\sqrt{2})$. The calculation goes as follows. Suppose, in (7), we condition on the random variables $\text{BadPoints}(P_0), \dots, \text{BadPoints}(P_t)$, rather than simply on Z_0, \dots, Z_t . This is more stringent conditioning, since the former random variables determine the latter. It is enough to establish (7) under the more stringent conditioning. So fix $\text{BadPoints}(P_0) = B_0, \dots, \text{BadPoints}(P_t) = B_t$, and note that this choice also fixes the resampling sets S_0, \dots, S_t . Suppose $Z_t = |\text{BadPairs}(P_t)| = k$. Inside the resampling set S_t we have a Poisson point process $P_{t+1} \cap S_t$ of intensity λ_r . Outside, by Lemma 3, there is a realisation of the hard disks process. Since we are seeking an upper bound on Z_{t+1} we may, by Lemma 4, replace $P_{t+1} \cap \bar{S}_t$ by a Poisson point process of intensity λ_r .

Let $k' = \mathbb{E}(Z_{t+1} \mid Z_0, \dots, Z_t)$. From the above considerations we have

$$(8) \quad k' \leq \int_{S_t} \lambda_r \int_{[0,1]^2} \lambda_r \mathbf{1}_{\|x-y\| \leq 2r} dy dx.$$

This is an overestimate, as we are double-counting overlapping disks whose centres both lie within S_t . Now, S_t is a union of at most $2k$ disks of radius $2r$. Thus

$$(9) \quad \begin{aligned} k' &\leq 2k\lambda_r^2 \int_{D_{2r}(\mathbf{0})} \int_{\mathbb{R}^2} \mathbf{1}_{\|x-y\| \leq 2r} dy dx \\ &= 2k\lambda_r^2 \int_{D_{2r}(\mathbf{0})} \int_{D_{2r}(x)} dy dx \\ &= 2k\lambda_r^2 \times 4\pi r^2 \times 4\pi r^2 \\ &= 32\lambda^2 k. \end{aligned}$$

There are further sources of slack here: there may be fewer than $2k$ disks, the disks comprising S_t certainly overlap, and, for points x near the boundary, some of disks $D_{2r}(x)$ will lie partly outside the unit square. (The last of these presumably has no effect asymptotically, as $r \rightarrow 0$.) Setting $\bar{\lambda} = 1/(4\sqrt{2}) = 0.17677+$, we see that $\alpha = k/k' > 1$ for any $\lambda < \bar{\lambda}$, and $(Z_t)_{t=0}^\infty$ is a supermartingale, as required. \square

The constant $\bar{\lambda}$ may seem quite small. Note, however, that classical rejection sampling cannot achieve any $\bar{\lambda} > 0$. The argument goes as follows. Divide $[0, 1]^2$ into $r \times r$ squares. If there are two points in the same square then they will certainly be less than distance $2r$ apart. The number of points in each square is Poisson distributed with mean λ/π . Thus for any $\lambda > 0$ the probability that a particular square has at least two points is bounded away from zero. The number of points in each square is independent of all the others. It follows that the runtime of classical rejection sampling is exponential in r^{-2} .

The above derivation for $\bar{\lambda}$ is quite crude and can be improved.

Lemma 7. *The constant $\bar{\lambda}$ in Lemma 6 can be taken to be 0.21027.*

Proof. For each of the $2k$ disks, the right-hand side of inequality (9) counts pairs of points (x, y) with x in the disk, and y anywhere within distance $2r$ of x . Since a bad event is determined by an *unordered* pair of points, this gives rise to significant double counting. In particular, pairs (x, y) with x and y lying in the same ball are double counted. We can subtract off these pairs to obtain a better estimate.

For a single ball, the correction is

$$C = \frac{1}{2} \int_{D_{2r}(\mathbf{0})} \lambda_r \int_{D_{2r}(\mathbf{0})} \lambda_r \mathbf{1}_{\|x-y\| \leq 2r} dy dx.$$

(The initial factor of one half arises because we want to count unordered pairs.) With the change of variables $x = 2rx'$ and $y = 2ry'$ this expression simplifies to

$$\begin{aligned} C &= \frac{1}{2} \times 16r^4 \lambda_r^2 \int_{D_1(\mathbf{0})} \int_{D_1(\mathbf{0})} \mathbf{1}_{\|x'-y'\| \leq 1} dy' dx' \\ &= 8\lambda_r^2 r^4 \int_{D_1(\mathbf{0})} L(\|x'\|) dx', \end{aligned}$$

where $L(\|x'\|)$ is the area of the ‘‘lens’’ $D_1(\mathbf{0}) \cap D_1(x')$. Letting ϱ denote the offset of the centres of the two disks, the area of the lens is given by

$$L(\varrho) = 2 \arccos(\varrho/2) - \frac{1}{2} \varrho \sqrt{4 - \varrho^2}.$$

(This is by elementary geometry: the lens is the intersection of two sectors, one from each of the disks, and its area can be computed by inclusion-exclusion.) An illustration (before shifting y to $\mathbf{0}$) is given in Figure 2. The shaded area is L .

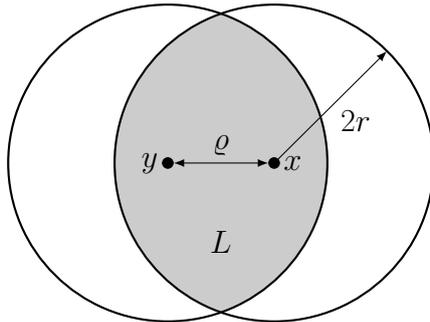


FIGURE 2. An illustration of double counting.

Translating to polar coordinates (ϱ, θ) ,

$$\begin{aligned}
C &= 8\lambda_r^2 r^4 \int_{D_1(\mathbf{0})} L(\|x'\|_2) dx' \\
&= 8\lambda_r^2 r^4 \int_0^{2\pi} \int_0^1 \varrho L(\varrho) d\varrho d\theta \\
&= \frac{8\lambda^2}{\pi^2} \times 2\pi \int_0^1 \varrho L(\varrho) d\varrho \\
&= \frac{16\lambda^2}{\pi} \left[\frac{\pi}{2} + (\varrho^2 - 1) \arccos\left(\frac{\varrho}{2}\right) - \left(\frac{\varrho}{4} + \frac{\varrho^3}{8}\right) \sqrt{4 - \varrho^2} \right]_0^1 \\
&= \left(8 - \frac{6\sqrt{3}}{\pi}\right) \lambda^2.
\end{aligned}$$

(The integral was evaluated using the Maple computer algebra system.) Our revised upper bound on k' is thus

$$(10) \quad k' \leq 2k(16\lambda^2 - C) = 2k\lambda^2 \left(8 + \frac{6\sqrt{3}}{\pi}\right),$$

Solving

$$\bar{\lambda}^2 \left(16 + \frac{12\sqrt{3}}{\pi}\right) = 1$$

yields the improved bound of $\bar{\lambda} = 0.21027+$. \square

There are other factors that could in principle be used to increase $\bar{\lambda}$ further — each disk necessarily overlaps with at least one other disk, some bad events are triple or quadruple counted — but the computational difficulties rapidly increase when attempting to account for these.

We have just seen that the number of iterations of the while-loop in Algorithm 1 is small when λ is below some threshold. It just remains to check that the body of the loop can be implemented efficiently.

Lemma 8. *There is an implementation of Algorithm 1 such that, for any fixed $\lambda < \bar{\lambda}$, the expected runtime of the algorithm is $O(r^{-2})$.*

Proof. No sophisticated data structures are required, but the runtime analysis requires some work.

Divide the unit square into a grid of $r \times r$ squares. Index the grid squares by $\mathcal{I} = [0, n]^2$, where $n \approx r^{-1}$. Let the grid squares be $\{\Gamma_{\mathbf{i}} : \mathbf{i} \in \mathcal{I}\}$. Note that if two points of a point set P lie in the same grid square then they will necessarily form a bad pair and lie in $\mathbf{BadPairs}(P)$. Moreover, if $(x, y) \in \mathbf{BadPairs}(P)$ and $x \in \Gamma_{\mathbf{i}}$ and $y \in \Gamma_{\mathbf{j}}$ then necessarily $\|\mathbf{i} - \mathbf{j}\|_{\infty} \leq 2$. So it seems, intuitively, that we should be able to implement the body of loop to run in time $O(n^2) = O(r^{-2})$, since we only have to examine $O(n^2)$ pairs of grid squares in order to complete the computation. Given Lemma 6, this would give an upper bound on total runtime of $O(r^{-2} \log r^{-1})$.

However, we can do better than this by noting that the amount of work to be done in each iteration decays exponentially. Recall the notation employed in Algorithm 1. Assume that, in addition to P and $B = \text{BadPoints}(P)$, we maintain a list L of grid squares containing points in B . In order to perform the computations within the while-loop, it is only necessary to consider grid squares that are within constant distance of a grid square in L . As the length of the list L decays exponentially, according to Lemma 6, we expect the overall runtime to be $O(r^{-2})$ rather than just $O(r^{-2} \log r^{-1})$. This is indeed the case. However, there is a technical issue; the time taken for the t th iteration of the loop is a random variable, say T_t . Recall the notation of Lemma 6, in particular that $Z_t = |\text{BadPairs}(P_t)|$ denotes the number of bad pairs of points (ones within distance $2r$ of each other) after t iterations of the loop. As may be expected, we can bound the expectation of T_t by a linear function of Z_{t-1} . Unfortunately, we can't just sum these estimates over $t \in \{1, T\}$ to get an upper bound on expected total runtime, as the random variable T_t is presumably correlated with Z_t . Instead, we shall “charge” the operations within the t th iteration of the loop either to Z_{t-1} or Z_t . and hence bound the expected runtime by $O(Z_0 + Z_1 + \dots + Z_T) = O(r^{-2})$.

Consider the work done during the t th iteration of the while-loop. Assume that the point sets $P = P_{t-1}$ and $B = B_{t-1}$ are available either from the initialisation phase or from the previous iteration of the loop. Also assume that we have the list L containing grid squares containing points in B . Note that $|L| \leq 2Z_{t-1}$, since each bad pair contributes at most two bad points. To compute P^S , we could sample a realisation of the Poisson point process in the unit square and reject all those points that are not within distance $2r$ of some point in B . However, this would be inefficient. Instead, we just produce realisations of a Poisson point process within grid squares that are within distance two of a grid square in L . Here, we take the distance between grid squares Γ_i and Γ_j to be $\|\mathbf{j} - \mathbf{i}\|_\infty$. We then reject each new point unless it is within distance $2r$ of a point in B ; the result is the required point set P^S .

Let

$$\begin{aligned} b_i &= |B_{t-1} \cap \Gamma_i|, \\ n_i &= |P_t^S \cap \Gamma_i|, \quad \text{and} \\ p_i &= |P_t \cap \Gamma_i|. \end{aligned}$$

Thus, restricted to grid square Γ_i , we have that b_i is the number of bad points carried forward from the previous iteration, n_i is the number of fresh points added during the current iteration, and p_i is the total number of points at the end of the current iteration. Also, let n'_i be the number of points generated during the current iteration that did not survive because they were not within distance $2r$ of some bad point.

Let $\mathcal{P} = \{(\mathbf{i}, \mathbf{j}) : b_i > 0 \text{ and } \|\mathbf{j} - \mathbf{i}\|_\infty \leq 2\}$. Note that, in order to compute the points that need to be added to P during the current iteration, only pairs of grid squares $\{(\Gamma_i, \Gamma_j) : (\mathbf{i}, \mathbf{j}) \in \mathcal{P}\}$ need to be examined. (Each new point must be within distance $2r$ of an existing bad point.) The work done during iteration t in computing P_t is then proportional to

$$\sum_{(\mathbf{i}, \mathbf{j}) \in \mathcal{P}} 1 + b_i(n_j + n'_j) = A + B + C,$$

where

$$A = \sum_{(\mathbf{i}, \mathbf{j}) \in \mathcal{P}} 1, \quad B = \sum_{(\mathbf{i}, \mathbf{j}) \in \mathcal{P}} b_i n_j \quad \text{and} \quad C = \sum_{(\mathbf{i}, \mathbf{j}) \in \mathcal{P}} b_i n'_j.$$

Here, A represents the fixed cost of cycling through all the relevant pairs of grid squares, and $A + B$ is the additional time required to examine each fresh point and see whether it needs to be retained (i.e., added to P^S).

Now $|\mathcal{P}| \leq 25|L| \leq 50Z_{t-1}$, and so $A = O(Z_{t-1})$. Also, n'_j is stochastically dominated by a Poisson random variable with mean $r^2 \lambda_r = \lambda/\pi = O(1)$; thus

$$\mathbb{E}(C) = \sum_{\mathbf{i}: b_i > 0} 25\lambda/\pi = O(Z_{t-1}).$$

Note that the n_j' new points in grid square Γ_j are freshly generated during the current iteration, and are discarded before the end of the iteration, and so have no effect on the subsequent evolution of the algorithm; the sequence $Z_0, \alpha Z_1, \alpha^2 Z_2, \dots$ remains a supermartingale, and the analysis in Lemma 6 is unaffected. (All other estimates will be deterministic, so this is the only point where we need to consider the potential for conditioning.)

The remaining term B may be bounded as follows:

$$\begin{aligned} B &= \sum_{(i,j) \in \mathcal{P}} b_i n_j \\ &\leq \sum_{(i,j) \in \mathcal{P}} 1 + 2 \binom{b_i}{2} + 2 \binom{n_j}{2} \\ &\leq \sum_{i: b_i > 0} 25 + 50 \sum_i \binom{b_i}{2} + 50 \sum_j \binom{n_j}{2} \\ &= O(Z_{t-1}) + O(Z_{t-1}) + O(Z_t), \end{aligned}$$

Here we use the fact that any pair of distinct points in a single grid square is certainly a bad pair.

The final task is to compute the new set of bad points. Since each of the new bad pairs must involve at least one point in P^S , the time to complete this task is proportional to

$$\sum_{\mathcal{P}'} 1 + n_i p_j \leq A' + B' + C'$$

where

$$\mathcal{P}' = \{(i, j) : n_i > 0 \text{ and } \|\mathbf{j} - \mathbf{i}\|_\infty \leq 2\}$$

and

$$A' = \sum_{\mathcal{P}'} 2, \quad B' = \sum_{\mathcal{P}'} 2 \binom{n_i}{2} \quad \text{and} \quad C' = \sum_{\mathcal{P}'} 2 \binom{p_j}{2}.$$

Now any grid square Γ_i with $n_i > 0$ must be within distance 2 of a grid square in the list L . Therefore, by similar reasoning to that used earlier,

$$\begin{aligned} A' &= \sum_{i: n_i > 0} 25 \times 2 = 25|L| \times 50 \leq 2500Z_{t-1}, \\ B' &\leq \sum_i 25 \times 2 \binom{n_i}{2} \leq 50Z_t \end{aligned}$$

and

$$C' \leq \sum_j 25 \times 2 \binom{p_j}{2} \leq 50Z_t.$$

So again, the time for this phase of the loop is bounded by

$$A' + B' + C' = O(Z_{t-1}) + O(Z_t) + O(Z_t) = O(Z_{t-1}) + O(Z_t).$$

It remains to analyse the initialisation phase. Generating the realisation of a Poisson point process of intensity λ_r in the unit square clearly takes time $O(r^{-2})$. To identify the bad pairs, we cycle through pairs of grid squares (Γ_i, Γ_j) with $\|\mathbf{j} - \mathbf{i}\|_\infty \leq 2$; there are $O(r^{-2})$ of these. An argument identical to those above shows that the time taken to identify the bad pairs is $O(Z_0)$. We are almost done, but we do need a better estimate for Z_0 than the one used in Lemma 6, which was $O(r^{-4})$. (This crude estimate was adequate at the time, as we were only interested in the logarithm of this quantity.) But we can now see that bad pairs can only come from pairs of grid squares separated by distance at most two. There are $O(r^{-2})$ of these, and each of them generates $O(1)$ bad pairs in expectation, so that $Z_0 = O(r^{-2})$. We saw in Lemma 6 that

$Z_0, \alpha Z_1, \alpha^2 Z_2$, is a supermartingale with $\alpha > 1$ (with the convention that $Z_t = 0$ for $t > T$). Thus the overall runtime of Algorithm 1 is

$$O(\mathbb{E}(Z_0 + Z_1 + Z_2 + \dots)) = \frac{\alpha}{\alpha - 1} O(\mathbb{E}(Z_0)) = O(r^{-2})$$

in expectation, assuming $\lambda < \bar{\lambda}$. □

Theorem 2 is obtained by combining Lemmas 6, 7 and 8.

5. THREE OR MORE DIMENSIONS

In higher dimensions, the hard disk model is known as the hard spheres model. Everything in Sections 3 and 4 carries across to $d > 2$ dimensions with little change. For general d , the appropriate scaling for the intensity is $\lambda_{r,d} = \lambda/(v_d r^d)$, where v_d is the volume of a ball of unit radius in d dimensions. Note that in a realisation of a Poisson point process with intensity $\lambda_{r,d}$, the expected number of points in a ball of radius r is λ .

The analogue of equation (8) is

$$k' \leq \int_{S_t} \lambda_{r,d} \int_{[0,1]^d} \lambda_{r,d} \mathbf{1}_{\|x-y\| \leq 2r} dy dx,$$

which leads to

$$k' \leq 2^{2d+1} \lambda^2 k.$$

So setting $\bar{\lambda} = 2^{-(d+\frac{1}{2})}$ we find that $\alpha = k/k' > 1$ for any $\lambda < \bar{\lambda}$. It follows that the runtime of partial rejection sampling is $O(\log r)$ for any $\lambda < \bar{\lambda}$.

By a result of Jenssen, Joos and Perkins [8], we lose just a constant factor when translating from intensity λ to packing density α . (It is partly to connect with their work, we measure intensity in terms of the expected number of points in a ball of radius r .) In the proof of [8, Thm 2], the following inequality is derived:

$$\alpha \geq \inf_z \max \{ \lambda e^{-z}, 2^{-d} \exp[-2 \cdot 3^{d/2} \lambda] \cdot z \}.$$

Assuming $\lambda \leq \bar{\lambda}$, which holds in the range of validity of our algorithm, we have $\sqrt{2}\lambda \leq 2^{-d}$ and hence

$$\begin{aligned} \alpha &\geq \inf_z \max \{ \lambda e^{-z}, \sqrt{2}\lambda \exp[-\sqrt{2}(3/4)^{d/2}] \cdot z \} \\ &= c_d \lambda, \end{aligned}$$

where

$$c_d = \inf_z \max \{ e^{-z}, \sqrt{2} \exp[-\sqrt{2}(3/4)^{d/2}] \cdot z \}.$$

Note that (c_d) is monotonically increasing, with $c_2 = 0.42220+$, and $\lim_{d \rightarrow \infty} c_d = 0.63724+$. It follows that we can reach expected packing density $\Omega(2^{-d})$ with $O(\log r^{-1})$ expected iterations. This is currently the best that can be achieved by any provably correct sampling algorithm with polynomial (in $1/r$) runtime [9]. The asymptotically best packing density currently rigorously known is $d2^{-d}$, but achieving this would require λ to grow exponentially fast in d . This is clearly beyond the capability of partial rejection sampling, but also beyond the capability of any known efficient sampling algorithm.

ACKNOWLEDGEMENTS

We thank Mark Huber and Will Perkins for inspiring conversations and bringing the hard disks model to our attention.

REFERENCES

- [1] Henry Cohn. A conceptual breakthrough in sphere packing. *Notices Amer. Math. Soc.*, 64(2):102–115, 2017. **1**
- [2] Henry Cohn, Abhinav Kumar, Stephen D. Miller, Danylo Radchenko, and Maryna Viazovska. The sphere packing problem in dimension 24. *Ann. of Math. (2)*, 185(3):1017–1033, 2017. **1**
- [3] Michael Engel, Joshua A. Anderson, Sharon C. Glotzer, Masaharu Isobe, Etienne P. Bernard, and Werner Krauth. Hard-disk equation of state: First-order liquid-hexatic transition in two dimensions with three simulation methods. *Phys. Rev. E*, 87:042134, Apr 2013. **3**
- [4] Heng Guo and Mark Jerrum. Perfect simulation of the hard disks model by partial rejection sampling. In *ICALP*, volume 107 of *LIPICs*, pages 69:1–69:10. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. **3**
- [5] Heng Guo, Mark Jerrum, and Jingcheng Liu. Uniform sampling through the Lovasz local lemma. In *STOC*, pages 342–355, 2017. **1, 2, 4**
- [6] Thomas C. Hales. A proof of the Kepler conjecture. *Ann. of Math. (2)*, 162(3):1065–1185, 2005. **1**
- [7] Thomas P. Hayes and Cristopher Moore. Lower bounds on the critical density in the hard disk model via optimized metrics. *CoRR*, abs/1407.1930, 2014. **3**
- [8] Matthew Jenssen, Felix Joos, and Will Perkins. On the hard sphere model and sphere packings in high dimensions. *Forum Math. Sigma*, 7:e1, 19, 2019. **1, 2, 3, 11**
- [9] Ravi Kannan, Michael W. Mahoney, and Ravi Montenegro. Rapid mixing of several Markov chains for a hard-core model. In *ISAAC*, pages 663–675, 2003. **3, 11**
- [10] Wilfrid S. Kendall. Perfect simulation for the area-interaction point process. In *Probability towards 2000 (New York, 1995)*, volume 128 of *Lect. Notes Stat.*, pages 218–234. Springer, New York, 1998. **3**
- [11] Wilfrid S. Kendall and Jesper Møller. Perfect simulation using dominating processes on ordered spaces, with application to locally stable point processes. *Adv. Appl. Probab.*, 32(3):844–865, 2000. **3**
- [12] Torgny Lindvall. On Strassen’s theorem on stochastic domination. *Electron. Comm. Probab.*, 4:51–59, 1999. **6**
- [13] Hartmut Löwen. Fun with hard spheres. In Klaus R. Mecke and Dietrich Stoyan, editors, *Statistical Physics and Spatial Statistics*, pages 295–331, 2000. **1**
- [14] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21(6):1087–1092, 1953. **1**
- [15] S. B. Moka, S. Juneja, and M. R. H. Mandjes. Perfect sampling for Gibbs processes with a focus on hard-sphere models. *ArXiv*, abs/1705.00142, 2017. **3**
- [16] Will Perkins. Birthday inequalities, repulsion, and hard spheres. *Proc. Amer. Math. Soc.*, 144(6):2635–2649, 2016. **3**
- [17] Chris Preston. Spatial birth-and-death processes (with discussion). *Bull. Inst. Internat. Statist.*, 46(2):371–391, 405–408 (1975), 1975. **5, 6**
- [18] V. Strassen. The existence of probability measures with given marginals. *Ann. Math. Statist.*, 36:423–439, 1965. **6**
- [19] David J. Strauss. A model for clustering. *Biometrika*, 62(2):467–475, 1975. **1**
- [20] Maryna S. Viazovska. The sphere packing problem in dimension 8. *Ann. of Math. (2)*, 185(3):991–1015, 2017. **1**
- [21] Jake Wellens. A note on partial rejection sampling for the hard disks model in the plane. *ArXiv*, abs/1808.03367, 2018. **3**

(Heng Guo) SCHOOL OF INFORMATICS, UNIVERSITY OF EDINBURGH, INFORMATICS FORUM, EDINBURGH, EH8 9AB, UNITED KINGDOM.

Email address: `hguo@inf.ed.ac.uk`

(Mark Jerrum) SCHOOL OF MATHEMATICAL SCIENCES, QUEEN MARY, UNIVERSITY OF LONDON, MILE END ROAD, LONDON, E1 4NS, UNITED KINGDOM.

Email address: `m.jerrum@qmul.ac.uk`