Stochastics and Statistics

Random sampling: Billiard Walk algorithm ☆

Elena Gryazina [a],[*], Boris Polyak [a],[b]

[a] *Institute for Control Sciences RAS, Moscow, Russia*
[b] *Laboratory of Structural Methods of Data Analysis in Predictive Modeling in Moscow Institute of Physics and Technology, Moscow, Russia*

ABSTRACT

Hit-and-Run is known to be one of the best random sampling algorithms, its mixing time is polynomial in dimension. However in practice, the number of steps required to obtain uniformly distributed samples is rather high. We propose a new random walk algorithm based on billiard trajectories. Numerical experiments demonstrate much faster convergence to the uniform distribution.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Generating points uniformly distributed in an arbitrary bounded region $Q \subset \mathbb{R}^n$ finds applications in many computational problems (Tempo, Calafiore, & Dabbene, 2004; Rubinstein & Kroese, 2008).

Straightforward sampling techniques are usually based on one of the three approaches: rejection, transformation, and composition. In the rejection approach, the region of interest Q is embedded into a region with available uniform sampler B (usually a box or a ball). At the next step, samples that do not belong to Q are rejected. Assume Q is the unit ball, and the bounding region B is the box $[-1, 1]^n$. Then for $n = 2k$, the ratio of the volumes of the box and the ball is equal to $q = \frac{\text{Vol}(Q)}{\text{Vol}(B)} = \frac{\pi^k}{k!2^k}$, thus $q \approx 10^{-8}$ for $n = 20$, so that one has to generate $\sim 10^8$ samples to obtain just a few of them in Q. For polytopes this ratio can be much smaller. Another way to exploit pseudo-random number generator for a simple region B is to map B onto Q via a smooth deterministic function with constant Jacobian. For instance, to obtain uniform samples in $Q = \{x : x^T A x < 1\}$, A being a positive definite matrix, it suffices to generate samples y uniformly in the unit ball $||y||_2 < 1$ and transform them as $x = A^{-1/2}y$. Unfortunately, such a transformation exists just for a limited class of regions. In the composition approach, the set Q is partitioned into a finite number of sets that can be efficiently sampled. For instance, a polytope can be partitioned into simplices, but the large number of them makes the procedure computationally hard.

Other sampling procedures use modern versions of the Monte Carlo technique based on the Markov Chain Monte Carlo (MCMC) approach (Gilks, Richardson, & Spiegelhalter, 1996; Diaconis, 2009). For instance, efficient algorithms for computing volumes using random walks can be found in Dyer, Frieze, and Kannan (1991), Lovasz and Somonovits (1993), Lovasz and Deak (2012). One of the most famous and efficient algorithms of the MCMC type is Hit-and-Run (HR), which was originally proposed by Turchin (1971) and independently by Smith (1984). The brief description of the HR algorithm is as follows. At every step HR generates a random direction uniformly over the unit sphere and picks the next point uniformly on the segment of the straight line in the given direction in Q. HR is applicable to various control and optimization problems (Polyak & Gryazina, 2008; Polyak & Gryazina, 2011; Dabbene, Shcherbakov, & Polyak, 2010) as well as to simulation-based multiple criteria decision analysis (Tervonen, van Valkenhoef, Basturk, & Postmus, 2013). Unfortunately, even for simple "bad" sets, such as level sets of ill-posed functions, HR techniques fail or become computationally inefficient.

A variety of applications and drawbacks of the existing techniques provides much room for improving and developing new sampling algorithms. For instance, there were attempts to exploit the approach proposed for interior-point methods of convex optimization (Nesterov & Nemirovsky, 1994) and to combine it with MCMC algorithms. As a result, the Barrier Monte Carlo method (Polyak & Gryazina, 2010) generates random points with better uniformity properties as compared to the standard Hit-and-Run.

On the other hand, the complexity of every iteration is in general high enough (the calculation of $\left(\nabla^2 F(x)\right)^{-1/2}$ is required, where $F(x)$ is a barrier function of the set). Moreover, the Barrier Monte Carlo method does not accelerate convergence for simplex-like sets.

In this paper we propose a new random walk algorithm motivated by physical phenomena of gas diffusing in a vessel. A particle of gas moves with a constant speed until it meets the boundary of the vessel, then it reflects (the angle of incidence equals the angle of reflection) and so on. When a particle hits another one, its direction and speed change. In our simplified model we assume that the direction changes randomly, while the speed remains the same. Thus our model combines the ideas of the Hit-and-Run technique and use of the billiard trajectories. There exists a vast literature on mathematical billiards, and many useful facts can be extracted from there (Tabachnikov, 1995; Galperin & Zemlyakov, 1990; Sinai, 1970; Sinai, 1978; Kozlov & Treshchev, 1991). The traditional theory addresses the behavior of one particular billiard trajectory in different billiard tables, their ergodic properties, and the conditions for the existence of periodic orbits. In stochastic analogs of the classical billiard (Evans, 2001), a direction after reflection is chosen randomly uniformly. Shake-and-Bake algorithms are based on stochastic billiards and generate points on the boundary of a convex set (Boender et al., 1991). The recently proposed version of the Shake-and-Bake algorithm (Dieker & Vempala, xxxx) exhibits polynomial-time convergence to the uniform distribution. Our algorithm is aimed at sampling the interior of a set (actually, later in the text we consider open regions). Besides that, we extend billiard trajectories of random length keeping the standard reflection law. Such an incorporation of randomness also improves the ergodic properties.

The paper is organized as follows. In Section 2 we present a novel sampling algorithm and prove that it produces asymptotically uniformly distributed samples in Q. In Section 3 we pay much attention to some properties of the Billiard Walk (BW), implementation issues are discussed as well. Simulation of BW for particular test domains is presented in Section 4. Much attention is devoted to the capability of BW to get out of the corner, in comparison with HR. Here we consider just the most demonstrative types of geometry. In Section 4.6 we briefly discuss possible applications of the algorithm.

2. Algorithm

Assume there is a bounded, open connected set $Q \subset \mathbb{R}^n, n \geqslant 2$, and a point $x^0 \in Q$. Our aim is to generate asymptotically uniform samples $x^i \in Q, i = 1, \ldots, N$.

The new BW algorithm generates a random direction uniformly over the unit sphere. Then the next sample is chosen as the end-point of the billiard trajectory of length ℓ. This length is chosen randomly; i.e., we assume that the probability of collision with another particle is proportional to δt for small time instances δt, this validates the formula for ℓ in the algorithm below. The scheme of the method is given in Fig. 1, while the precise routine is as follows.

2.1. Algorithm of Billiard Walk (BW)

1. Take $x^0 \in Q$; $i = 0, x = x^0$.
2. Generate the length of the trajectory $\ell = -\tau \log \xi, \xi$ being uniform random on $[0,1]$, τ is a specified constant parameter of the algorithm.
3. Pick a random direction $d \in \mathbb{R}^n$ uniformly distributed over the unit sphere (i.e., $d = \xi/\|\xi\|$, where $\xi \in \mathbb{R}^n$ has the standard Gaussian distribution). Construct a billiard trajectory starting
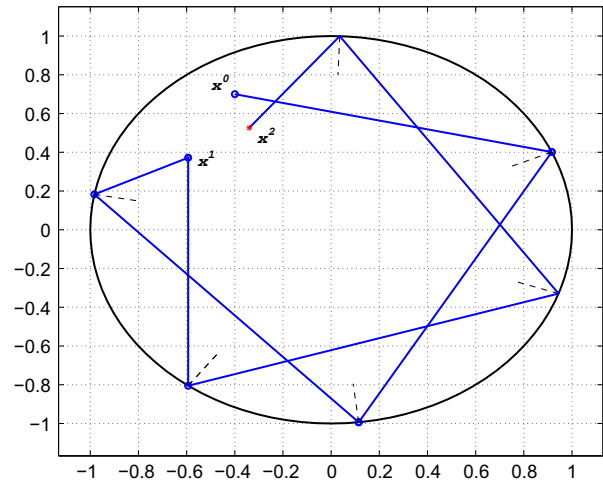

Fig. 1. Billiard walk.

at x^i and having initial direction d. When the trajectory meets the boundary with internal normal $s, \|s\| = 1$, the direction is changed as

$$d \to d - 2(d, s)s,$$

where (d, s) is the scalar product.
4. If a point with nonsmooth boundary is met or the number of reflections exceeds R, go to step 2. Otherwise proceed until the length of the trajectory equals ℓ.
5. $i = i + 1$, take the end-point as x^{i+1} and go to step 2.

The algorithm involves two parameters τ and R and we discuss their choice below.

We prove asymptotical uniformity of the samples produced by BW for convex and nonconvex cases separately. The requirements on Q are different for these two cases, while the sampling algorithm remains the same. Consider the Markov Chain induced by the BW algorithm $x^0, x^1, \ldots$. For an arbitrary measurable set $A \subseteq Q$, denote by $\mathbf{P}(A|x)$ the probability of obtaining $x^{i+1} \in A$ for $x^i = x$ by the BW algorithm. Then $\mathbf{P}_N(A|x)$ is the probability to get $x^{i+N} \in A$ for $x^i = x$. We also denote by $p(y|x)$ the probability density function for $\mathbf{P}(A|x)$, i.e. $\mathbf{P}(A|x) = \int_A p(y|x)dy$.

Theorem 1. *Assume Q is an open bounded convex set in $\mathbb{R}^n$, the boundary of Q is piecewise smooth. Then the distribution of points x^i generated by the BW algorithm tends to the uniform one over Q, i.e.*

$$\lim_{N \to \infty} \mathbf{P}_N(A|x) = \lambda(A)$$

for any measurable $A \subseteq Q$, $\lambda(A) = \text{Vol}(A)/\text{Vol}(Q)$ and any starting point x.

Proof. First, the algorithm is well-defined: at step 4 with zero probability the algorithm sticks at a point with nonsmooth boundary. On the other hand ℓ and d are chosen in such a way that, with positive probability, x^{i+1} is obtained by less than R reflections (see detailed discussion of "bad" situations in SubSections 3.1 and 3.2).

In view of Theorem 2 in Smith (1984) based on the asymptotic properties of Markov Chains, the two assumptions on $p(y|x)$ imply that the uniform distribution over Q is a unique stationary distribution, and it is achieved for any starting point $x \in Q$. The first assumption requires the existence of $p(y|x)$ and its symmetry; the second assumption claims its positivity $p(y|x) > 0$ for all $x, y \in Q$.

Now we show that there exists a probability density function; i.e. for any $x, y \in Q$, the transition probability from x to a small neighborhood δy of y is proportional to the volume of δy. Among the trajectories proceeding from x to δy, there exist a conic bundle of trajectories with no reflections, as well as some trajectories with $1, 2, \ldots, R$ reflections. For a bundle of trajectories with no reflections $\mathbf{P}(\delta y | x) \sim \mathbf{P}(\delta \theta) \mathbf{P}(\delta \ell)$, where $\mathbf{P}(\delta \theta)$ is the probability of choosing the spatial angle and $\mathbf{P}(\delta \ell)$ is the probability of choosing a certain trajectory length $\ell \in \delta \ell$ while $p \sim q$ means "p is proportional to q". $\mathbf{P}(\delta \theta)$ is proportional to the volume of the base of the cone, $\mathbf{P}(\delta \ell) \sim \delta \ell$, thus $\mathbf{P}(\delta y | x) \sim \mathrm{vol}(\delta y)$.

The bundles of trajectories with reflections are also cones with small spatial angle $\delta \theta$. The area of reflection with a smooth boundary can be approximated as plain region. Then reflection does not change the geometry of the bundle, and the proof for this situation remains the same as for the bundle of trajectories with no reflections. Hence, $p(y | x)$ exists for all $x, y \in Q$.

For convex bodies, the positivity of $p(y | x)$ is obvious, all the points are reachable by a trajectory with no reflections.

The symmetry of the probability density function follows from the uniformity of the distribution of the directions and reversibility of a billiard trajectory due to the reflection law: the angle of incidence is equal to the angle of reflection. Therefore, all the assumptions on $p(y | x)$ are satisfied, and the distribution of points x^i generated by the BW algorithm tends to the uniform distribution on Q. $\square$

Theorem 2. *Assume Q is a bounded and open set, the boundary of Q is piecewise smooth and for all $x, y \in Q$ there exists a piecewise-linear path such that it connects x and y, lies inside Q and has no more than B linear parts (B is an arbitrary positive integer). Then the distribution of points x^i generated by the BW algorithm tends to the uniform distribution on Q in the same sense as in* Theorem 1.

Proof. Again, the algorithm is well defined: with probability one a point $x^{i+1} \neq x^i$ is found for arbitrary $x^i \in Q$.

All the constraints on Q are important. The existence of a piecewise-linear path implies connectedness and guarantees that, starting from any point, we can reach a measurable neighborhood of any other point in Q. Boundedness is necessary to define the uniform distribution on Q and to prevent the trajectories to go to infinity. Openness allows us to connect any two points with a tube of nonzero measure. Hence, there exists a piecewise linear trajectory connecting two arbitrary points.

Consider $p_N(y | x)$, the probability density function of $\mathbf{P}_N(A | x)$. The inequality $p_N(y | x) > 0$ holds for all integer $N > B$. The equality $p(x^{i+1} | x^i) = p(x^i | x^{i+1})$ (reversibility) holds for every pair of consecutive points due to the reflection law: the angle of incidence is equal to the angle of reflection. Therefore, $p_N(y | x) = p_N(x | y)$.

Hence, the distribution of the subsequence $x^0, x^N, x^{2N}, \ldots,$ tends to the uniform one for $N > B$. The same is true for every subsequence $x^i, x^{N+i}, x^{2N+i}, \ldots$. Since all the subsequences have asymptotically uniform distribution, the distribution of points x^i generated by the BW algorithm tends to the uniform distribution on Q. $\square$

There exist plenty of nonconvex domains that satisfy the conditions of Theorem 2. For instance, an estimate of B for the toroid is given in Subsection 4.8. Note that the constant B characterizes the geometry of Q.

3. Discussion

We discuss some implementation issues first.

3.1. Nonsmooth boundary points

The measure of the set of points belonging to nonsmooth boundary is zero but the probability of hitting the nonsmooth part of the boundary is nonzero for some starting x^0. For instance, consider two similar convex and nonconvex sets

$$Q_1 = \left\{ x \in \mathbb{R}^2 : x_1^2 / 4 + x_2^2 < 1, x_1 < \sqrt{3} - |x_2| \right\},$$

$$Q_2 = \left\{ x \in \mathbb{R}^2 : x_1^2 / 4 + x_2^2 < 1, x_1 < \sqrt{3} + |x_2| \right\},$$

both being truncated ellipses (Fig. 2). A large portion of directions makes trajectories of length $\ell \geqslant 4$ starting at the focus $x^0 = \left(-\sqrt{3}; 0 \right)$ hit the nonsmooth boundary at the second focus $x^1 = (\sqrt{3}; 0)$. The reason is that we take a particular starting point. The measure of "bad" starting points is zero and this effect never happens when the starting point is taken randomly with some distribution.

3.2. Choice of τ and R

To run the algorithm we need to specify the parameters τ and R. The value of τ strongly affects the behavior of the method. For τ small enough, BW becomes slower than HR; it behaves as a ball walk with radius τ. Empirical observations show that fast
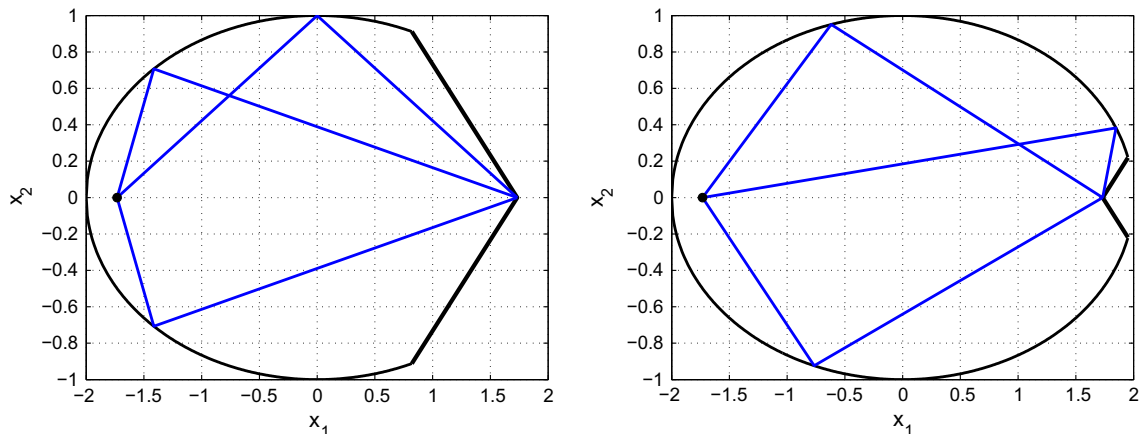

Fig. 2. Convex and nonconvex domains where a nonsmooth boundary point can be achieved.

convergence to the uniform distribution is achieved for $\tau \approx \mathrm{diam}Q$, where $\mathrm{diam}Q$ is the diameter of the set Q.

We restrict the number of reflections by R for every trajectory (step 4 of the Algorithm). The goal is to avoid situations when the trajectory length remains less than ℓ after a large number of reflections (a typical example is addressed in SubSection 4.4). The choice of R is mostly focused on eliminating computationally hard trajectories. The value of R should be large enough to implement most of the trajectories. But R also depends on τ. The longer the trajectory one needs to implement, the more reflections are required. We usually take $R = 10n$ to make it dimension dependent.

3.3. Preliminary transformation of Q

If Q is "ill-shaped," sometimes it can be improved with its linear transformation. For instance, if Q is a box $Q = \{x \in \mathbb{R}^n : |x_i| < a_i, i = 1, \ldots, n\}$, and it is far from being cubic ($\min a_i / \max a_i \ll 1$), a simple scaling transforms Q into a cube. A similar scaling transforms an ellipsoid into a ball. In the general case, the following scaling can be helpful. Assume Q has a barrier $F(x)$ defined on Q such that $F(x) \to +\infty$ as $x \to \partial Q$. In Nesterov and Nemirovsky (1994), a special class of self-concordant barriers is considered. For instance, for the polytope defined by m linear inequalities $Q = \{x \in \mathbb{R}^n : (a_i, x) < b_i, i = 1, \ldots, m\}$, this barrier is $F(x) = -\sum_i \log(b_i - (a_i, x))$. Then it is easy to find an approximate minimum x^* of $F(x)$. *Dikin ellipsoid* $E = \{x : (H(x - x^*), (x - x^*)) \leqslant 1, H = \nabla^2 F(x^*)\}$, lies in Q and it is a good approximation of the polytope Q. Hence we can calculate the linear mapping $T = H^{-1/2}$; by generating directions $d' = Td$, where d is uniformly distributed over the unit sphere, we can strongly accelerate the convergence. However sometimes none of the transformations can improve the shape of the set; the simplex is known to be the worst-case example.

3.4. Boundary oracle and normals

Both the HR and BW algorithms require computation of the intersections of a straight line (defined by the point x^k and the direction d of the trajectory) with the set Q. We call *Boundary Oracle* (BO) the procedure that calculates the boundary of the segment $[\underline{t}, \bar{t}]$, where

$$\underline{t} = \max_{t<0}\{t : x^k + td \in \partial Q\}, \quad \bar{t} = \min_{t>0}\{t : x^k + td \in \partial Q\}$$

(we assume that Q is convex, otherwise the point of the first intersection of the straight line and the boundary of Q is taken). Thus HR needs both $\underline{t}$ and $\bar{t}$ for every iteration, and the computational cost of HR is equal to two BO per sample. BW takes $\bar{t}$ for every reflection and the computation cost of BW is one BO per reflection. In most applications, finding BO is not a problem. For instance, if Q is a polytope defined by m linear inequalities

$$Q = \{x \in \mathbb{R}^n : (a^i, x) < b_i, i = 1, \ldots, m\}$$

then BO $[\underline{t}, \bar{t}]$ can be written explicitly. Calculate $t_i = \frac{b_i - (a^i, x^k)}{(a^i, d)}, i = 1, \ldots, m$, and take

$$\underline{t} = \max_{i:t_i<0} t_i, \quad \bar{t} = \min_{i:t_i>0} t_i.$$

Numerous examples of BO for other sets Q (for instance, defined by *Linear Matrix Inequalities*) can be found in Polyak and Gryazina (2008), Polyak and Gryazina (2011), Dabbene et al. (2010).

Billiard walks also require the calculation of normals s at the boundary points. In most applications it is not hard; for instance, for a polytope we have $s = a_i$, where i is the index for which the maximum or the minimum in the formulas above is achieved.

3.5. A comparison of HR and BW

Our goal in the test examples below is to compare HR and BW. We use several tools for this purpose. Sometimes theoretical considerations can help to compare the number of iterations to quit a corner. It is well known that HR may require too many iterations to get out of the corner, see estimates in Lovasz and Vempala (2004). We will show that estimates for BW are much more optimistic for many particular examples. On the other hand, we use simulation for the comparison as well. We exploit different tools to demonstrate that one sampling set is closer to uniform than another. Sometimes graphical figures in the 2D plane are quite evident. In other cases we demonstrate strong serial correlation in the samples. Finally, we use a parametric partition of Q and compare the number of empirical frequencies with the theoretical number for the uniform distribution via the χ^2 criterion.

To make final conclusions on the comparison of the two methods, we should have the following in mind. Of course, computationally BW is harder than HR. It requires more BO calculations, each reflection at the boundary also requires extra calculations for normals. We characterize the computational complexity by the number of calls to the BO and compare the outcomes of HR and BW obtained from the same number of BO (the number of samples is different in this case). The observed acceleration of convergence to the uniform distribution often makes BW preferable to HR.

4. Test sets and simulation

Some sets below are unbounded; we present them to analyze the behavior at a corner. We say that a trajectory quits the corner if it goes to infinity.

4.1. Plane angle

Let the angle $Q \subset R^2$ be equal to $\alpha < \pi$. Then any billiard trajectory quits Q after no more than $N^* = \lceil \pi/\alpha \rceil$ reflections for all initial points and initial directions; here $\lceil a \rceil$ stands for the smallest integer greater than or equal to a. The proof of this fact is as follows: if we reflect the angle N times around its side, billiard trajectory becomes the straight line. Every intersection of the line and the angle side corresponds to the reflection of the billiard trajectory. A straight line cannot intersect any straight line (not coinciding with itself) twice, and the total number of intersections with the reflected angle sides is no more than N^*. Thus N^* reflections are enough to quit the corner.

For HR we quit Q with probability $1 - (1 - \alpha/\pi)^N$ after no more than N iterations. For $N = N^*$ being large enough, HR quits Q with probability $1 - 1/e = 0.63$, while BW quits with probability one.

It is of interest to estimate the average number of reflections (over random initial directions). Consider the triangle $Q = \{x \in \mathbb{R}^2 : |x_1| \leqslant \operatorname{atan} \frac{\alpha}{2}, x_2 \leqslant 1\}$ with one of the angles equal to α. Let the BW trajectories start at $x^0 = (0; 0.1)$ and calculate the number of reflections until the trajectory reaches the line $x_2 = 1$ (i.e. quits the corner). Fig. 3 depicts 25 trajectories plotted for $\alpha = \pi/4$.

The results of 5000 runs and various α are given in Table 1. The empirical observations show that the average number of reflections for BW is equal to $N^*/2$. For HR we calculate the number of iterations until BO reaches the line $x_2 = 1$.

We conclude that BW is slightly more efficient that HR.

4.2. Multidimensional case: polyhedral cone Q

For a polyhedral cone there exists a number M which does not depend on the initial data, such that any billiard trajectory quits Q after no more than M reflections (see Sinai, 1978, also Tabachnikov,
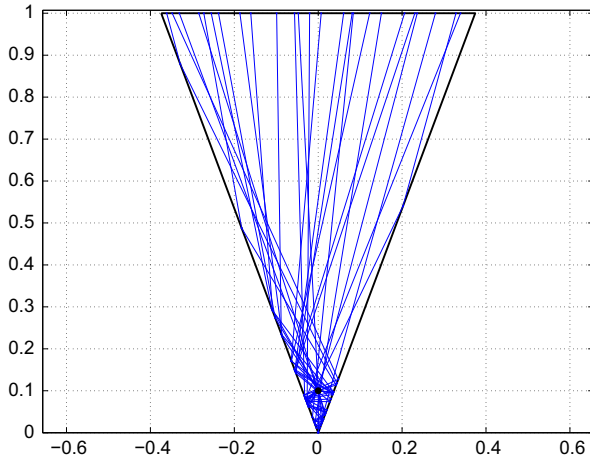
Fig. 3. 25 trajectories reaching the line $x_2 = 1$ starting from $[0; 0.1]$.

Table 1
The mean and standard deviation (in parentheses) for the number of BW reflections and the number of HR iterations required to quit the angle α.

α	BW	HR
$\pi/2$	2.28 (0.87)	2.37 (1.74)
$\pi/4$	3.08 (1.3)	3.75 (2.98)
$\pi/10$	5.94 (2.93)	8.23 (7.1)
$\pi/50$	25.08 (14.46)	39.25 (34.54)

1995, Theorem 7.17). However M depends on the geometry of Q. If M is large ($M > R$) then sometimes the BW algorithm gets stuck at x^i. However it can be proved that BW is well defined with probability one.

4.3. Orthant $Q = \{x \in R^n : x > 0\}$

It is easy to show that a billiard trajectory quits Q after no more than n reflections for an arbitrary initial point and initial direction. Indeed, for a given initial direction d, every reflection makes one of the components positive (if $d > 0$ componentwise, the trajectory quits Q). Let $I = \{i : d_i < 0\}$, then every reflection eliminates at least one negative component of d, and after no more than n reflections we have $I = \emptyset$.

A HR trajectory quits Q with probability $2^{-(n-1)}$ after a single iteration, thus it requires approximately 2^{n-1} iterations to quit Q with probability $1 - 1/e = 0.63$. The probability to quit the orthant after no more that n iterations is $2^{-(n-2)}(1 - 2^{-n})$ for HR and it decreases dramatically as the dimension n grows. Hence BW is much more efficient than HR for this case. Simulations for the cube (SubSection 4.6) confirm this statement.

All these results show that a polyhedral corner is not a problem for BW in contrast to HR, where the distance of the initial point to the corner and the size of the angle plays a significant role. The results can be extended to curvilinear corners with nondegenerate linear approximation, i.e. if a linear approximation of a corner is a polyhedral cone with nonempty interior.

4.4. Concave corner

In concave corners (that is, corners with concave boundaries) pure billiard trajectories may expose a large number of reflections (Sinai, 1970). Consider a typical domain (Fig. 4) with concave angle

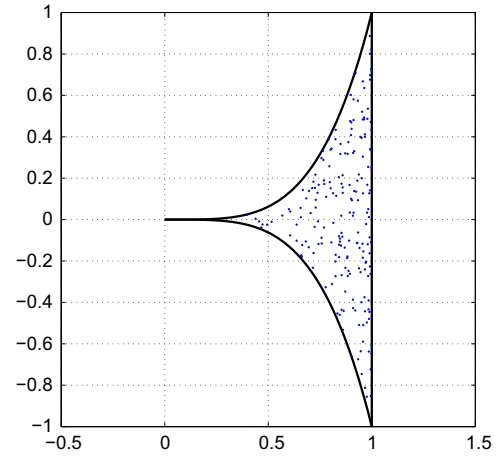$$Q = \left\{ x \in R^2 : -x_1^4 < x_2 < x_1^4, \quad x_1 \geqslant 1 \right\}. \tag{1}$$

Fig. 4. 200 points generated by BW for domain (1).

For a fixed ℓ, the length of a billiard trajectory may remain less than ℓ after a large number of reflections. Indeed, start the trajectory at the point $x^0 = (0.9; \varepsilon)$, ε being small enough, fix $\ell = 1, d = (-1; 0)$ and compute the number of reflections required. The results are shown in Table 2. As one can notice, the number of reflections increases dramatically as the second coordinate of x^0 tends to zero, and even for $x_2^0 = 10^{-4}$, the trajectory can not be implemented. To avoid these situations we restrict the number of reflections by R in the BW algorithm. But, in general, these "bad" directions are rare. Fig. 4 depicts 200 points for domain (1), the average number of reflections per point is six.

4.5. Strip

For domains of the form $Q = \{x \in R^2 : 0 < x_2 < 1, |x_1| < M\}$, M being large enough, HR and BW demonstrate different abilities to walk along x_1. Below we show that if one counts the average number of steps per one BO call, BW is approximately 6 times faster. For a random line intersecting the strip, let $\overline{\Delta x}$ be the horizontal component of the intersection averaged over directions. HR takes 2 BO per step, and the average shift along x_1 for uniformly distributed initial point is

$$\overline{\Delta x}_{HR} = \frac{1}{2}\overline{\Delta x} \int_0^1 \int_0^1 |x_1 - x_2| dx_1 dx_2 = \frac{1}{6}\overline{\Delta x}$$

per one BO. BW gives $\overline{\Delta x}_{HR} = 1/2\overline{\Delta x}$ for the first reflection (1 BO) and then $\overline{\Delta x}$ for every subsequent BO. The average shift along x_1 produced by BW after N BO is $\overline{\Delta x}_{BW} = \left(1 - \frac{1}{2N}\right)\overline{\Delta x}$ per one BO. Thus BW is 6 times more efficient than HR.

4.6. Cube

For the unit cube $Q = \{x \in \mathbb{R}^n : 0 < x < 1\}$ (the inequality is understood component-wise), we can compute the next point of the BW algorithm explicitly.

Table 2
The number of reflections required to implement the trajectory of length 1 for domain (1) starting at $x^0 = (0.9; \varepsilon)$ in the direction $d = (-1; 0)$.

ε	Number of reflections
1e−3	746
5e−4	1851
4e−4	2480
3e−4	3617
2e−4	6158
1.1e−4	13,496
1.01e−4	$> 5 \cdot 10^6$

At the current point x, for given ℓ and d calculate $k_i = \lfloor x_i + \ell d_i \rfloor$ ($\lfloor x \rfloor$ is the maximal integer less than or equal to x) and walk to y:

$$y_i = \begin{cases} x_i + \ell d_i - k_i, & k_i \text{ is even} \\ 1 - (x_i + \ell d_i - k_i), & k_i \text{ is odd} \end{cases}, \quad i = 1, \ldots, n.$$

Of course there is no need to apply MCMC algorithms for random sampling in the cube, one can generate a vector of n independent uniform random variables over $[0,1]$. Moreover, the shape of the cube is so nice that the distribution of the HR points converges to the uniform fast enough. Nevertheless it is of interest to compare BW and HR for this simplest case.

For various dimensions n we sample $N_{BW} = 1000$ points by the BW algorithm and calculate the amount N_{BO} of the BO calls needed. Then we sample $N_{HR} = \lceil N_{BO}/2 \rceil$ points by HR. In implementing BW, we take $\tau = \sqrt{n}, R = 10n$, starting point is uniform random for both sampling algorithms.

First we examine serial correlation for points produced by different samplers. To judge about serial correlation, we partition the unit cube into $q_i, i = 1, \ldots, 2^n$, small cubes of equal volumes. Then we calculate the empirical probability to proceed between different parts $\mathbf{P}(x^{i+1} \notin q_j | x^i \in q_j)$. Table 3 shows the results as compared to the theoretical probability U for independent uniformly distributed points (which is $1 - 2^{-n}$). One can see that serial correlation is much stronger for HR than for the BW samples.

Then we make the χ^2 frequency test for 10,000 HR points in $\mathbb{R}^{10}$. We take 10 equal volume slabs in the ith coordinate direction for $i = 1, \ldots, 10$, and make 10 χ^2 tests all together. The results are shown in Table 4.

We start BW with computational complexity 20,000 BO and obtain 2148 points (~ 215 per slab). The χ^2 frequency test results are shown in Table 5.

Upper and lower χ^2 values for 10% statistical significance for 9 degrees of freedom are $[3.3, 16.9]$ (for two tailed χ^2 test). Thus HR fails all 10 χ^2 tests while BW fails just 2 out of the 10 tests.

4.7. Simplex

The next test set is the standard n-dimensional simplex

$$Q = \left\{ x_i > 0, \sum x_i = 1, i = 0, 1, \ldots, n \right\}.$$

The simplex is a set with many corners and the geometry of simplex cannot be improved by any affine transformation. We know that for

HR walk it takes a lot of iterations to get out of a corner, thus it is interesting to compare HR and BW.

Smooth boundary of Q is specified by the points $\partial Q = \{x \in \mathbb{R}^{n+1} : x_k = 0, x_i \neq 0, i = 0, \ldots, n, i \neq k\}$, and the internal normal of unit length for these points is

$$s = \sqrt{\frac{1}{n(n+1)}} \left[-1, \ldots, \underbrace{n}_{k}, \ldots, -1 \right]^T.$$

The length of any edge of the simplex is $\sqrt{2}$ for every dimension n, the choice of the parameter $\tau = \sqrt{2}$ seems reasonable.

Note that for $n = 2$, the samples visually look uniformly distributed for both algorithms. To decide about uniformity more rigorously in the multidimensional case, consider the sequence of enclosed simplices $S_\alpha = \{x \in \mathbb{R}^{n+1} : x_i \geq \alpha, \sum x_i = 1\}, 0 \leq \alpha \leq \frac{1}{n+1}$. For $\alpha = 0, S_0$ is the initial simplex, and for $\alpha = \frac{1}{n+1}$, the simplex S_α contains one point. Let $\hat{f}(\alpha)$ be the portion of points contained in S_α, and denote $f(\alpha) = \text{vol} S_\alpha / \text{vol} S_0 = (1 - (n+1)\alpha)^n$. Fig. 5 shows $\hat{f}(\alpha)$ for $n = 50, N = 300, x^0 = \{1/(n+1), \ldots 1/(n+1)\}$. The red line corresponds to the uniformly distributed points, the black line describes the distribution for the HR points, and the blue line for BW points. We conclude that for BW samples, the empirical values of $\hat{f}(\alpha)$ are much closer to the mean value $f(\alpha)$ than for the HR samples.

We also perform two χ^2 tests for $n = 10$. For the first one we partition Q into 10 simplices $Q = S_0 \supset S_{\alpha_1} \supset \ldots \supset S_{\alpha_{10}} = \emptyset$ such that $\text{Vol}(S_{\alpha_i} \setminus S_{\alpha_{i+1}}) = \text{Vol} S_{\alpha_9} = \frac{1}{10} \text{Vol} S_0$. These differences $S_i \setminus S_{\alpha_{i+1}}$ are of various geometry but their volumes are equal. For the second test we take $n + 1$ subsets of the same volume and geometry, these subsets Q_i contain points mostly close to the selected vertex v^i:

$$Q_i = \{x \in Q : \|x - v^i\|_2 < \|x - v^j\|_2, \quad j \neq i\}.$$

Restricting ourselves to 20,000 BO, we obtain 10,000 HR points and about 2000 BW points. Tables 6 and 7 present the experimental results.
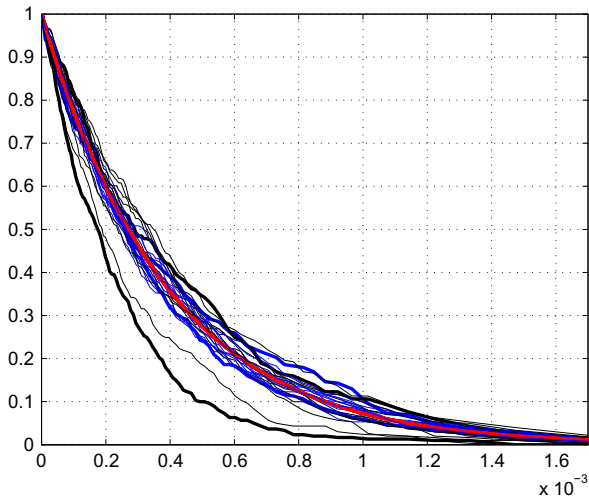
Table 3
Empirical probability to proceed between different parts of the cube for BW and HR, and the uniform distribution.

n	BW	HR	U
10	0.902	0.391	0.99902344
25	0.957	0.388	0.99999997
50	0.976	0.383	0.99999999

Table 5
The observed frequency of the BW points in the slab j in the ith coordinate direction and the χ^2 statistics.

	1	2	3	4	5	6	7	8	9	10	χ^2 st.
1	198	204	188	218	216	224	218	242	191	249	17.22
2	234	207	196	226	204	196	225	230	220	210	8.21
3	210	230	218	201	202	214	232	200	222	219	5.6
4	242	243	203	198	202	232	221	208	220	179	17.7
5	211	231	184	236	229	206	210	235	192	214	13.52
6	209	193	242	205	216	208	212	223	222	218	7.12
7	190	223	226	233	197	217	226	195	200	241	13.42
8	200	231	199	191	207	211	212	220	247	230	12.46
9	231	213	212	224	189	234	209	225	197	214	8.6
10	204	237	227	198	201	230	208	215	211	217	7.11

Table 4
The observed frequency of the HR points in the slab j in the ith coordinate direction and the χ^2 statistics.

	1	2	3	4	5	6	7	8	9	10	χ^2 st.
1	927	1087	1096	985	987	992	963	979	1000	984	24.64
2	1129	969	884	1026	1049	963	935	959	983	1103	52.31
3	1135	1134	970	951	983	976	980	961	822	1088	81.7
4	1008	1029	977	1046	961	870	932	971	1117	1089	49.05
5	820	1004	1092	1043	956	960	1107	1174	916	928	100.23
6	1001	1068	992	1014	1051	1004	944	958	935	1033	17.71
7	1015	890	905	916	1010	953	1028	982	1077	1224	87.93
8	1130	1098	1074	1078	1032	1021	824	956	886	901	95.24
9	913	983	980	1059	1023	902	1017	1050	1011	1062	28.63
10	1056	1013	966	972	950	1002	951	979	1016	1095	19.87

Fig. 5. The portion of points contained in S_α for uniformly distributed points (red), HR (black) and BW (blue). $n = 50$, 300 points. The horizontal axis corresponds to the parameter α. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 6. The (x_1, x_2)-projection of HR points (black) and BW points (blue) for toroid (2). $n = 10$, $N_{BW} = 500$, $N_{HR} = 1764$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Recalling the upper and lower χ^2 test values [3.3, 16.9], we conclude that the BW points fit uniform distribution while the HR points do not.

4.8. Toroid

Both the HR and BW algorithms are applicable to nonconvex sets. Consider the toroid formed by an n-dimensional ball of radius r with its center rotating over a circle in the (x_1, x_2)-plane:

$$Q = \{x \in \mathbb{R}^n : ||x - c_x|| < r\}, \tag{2}$$

where $c_{xi} = \frac{x_i}{\sqrt{x_1^2 + x_2^2}}$, $i = 1, 2$, $c_{xi} = 0, i > 2$.

The conditions of Theorem 2 are satisfied with $B = \left\lceil \frac{\pi}{2 \arccos\frac{1-r}{1+r}} \right\rceil + 1$, i.e. for all $x, y \in Q$ there exists a piecewise-linear path such that it connects x and y, lies inside Q, and has no more than B linear parts.

Fig. 6 depicts 500 BW samples and 1764 HR points (projected onto the (x_1, x_2)-plane) for the set (2) of dimension 10 and $r = 1/3$. The number of samples is different because implementation of 500 BW samples requires 1764 BO calculations. HR points are plotted with black dots, BW points with blue ones.
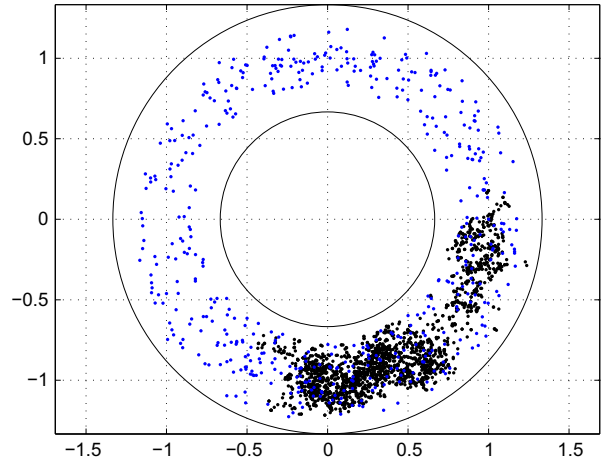
It can be easily seen that the angle distribution of the BW points is much more uniform than those for the HR points, the latter remain in the neighborhood of the initial point. Note that the visual lack of uniformity in the radial direction is an "optical effect" because we provide a 2D projection of the 10D picture.

5. Applications

In this paper we do not address numerous applications of the new version of random sampling. We can mention just few of them: global optimization (in particular, concave programming), control problems, robustness issues, numerical integration, calculation of the volume and the center of gravity and so on; see, for instance, our previous papers (Polyak & Gryazina, 2008; Polyak & Gryazina, 2011; Dabbene et al., 2010; Polyak & Gryazina, 2010). We plan to consider these applications in future works.

Table 6
The observed frequencies for the HR and BW points in the subsets $S_{\alpha_i} \setminus S_{\alpha_{i+1}}$ and the χ^2 statistics.

	1	2	3	4	5	6	7	8	9	10	χ^2
HR	855	917	925	897	997	978	1024	1025	1080	1303	144.04
BW	152	175	163	177	189	192	206	182	214	241	6.06

Table 7
The observed frequencies for the HR and BW points in the subsets $Q_i, i = 0, \ldots, 10$ and the χ^2 statistics.

	0	1	2	3	4	5	6	7	8	9	10	χ^2
HR	976	751	1050	1018	826	676	1084	521	1028	1424	947	697.54
BW	172	188	167	177	158	179	176	154	147	178	195	12.1

References

Boender, C., Caron, R., McDonald, J., Kan, A. R., Romeijn, H., Smith, R., et al. (1991). Shake-and-bake algorithms for generating uniform points on the boundary of bounded polyhedra. *Operations Research, 39*(6), 945–954.

Dabbene, F., Shcherbakov, P., & Polyak, B. (2010). A randomized cutting plane method with probabilistic geometric convergence. *SIAM Journal of Optimization, 20*(6), 3185–3207.

Diaconis, P. (2009). The markov chain monte carlo revolution. *Bulletin of the American Mathematical Society, 46*(2), 175–205.

Dieker, A., & Vempala, S. S. (xxxx). Stochastic billiards for sampling from the boundary of a convex set. <http://www2.isye.gatech.edu/~adieker3/publications/stochasticbilliard-submit.pdf>.

Dyer, M., Frieze, A., & Kannan, R. (1991). A random polynomial-time algorithm for approximating the volume of convex bodies. *Journal of the ACM, 38*(1), 1–17.

Evans, S. N. (2001). Stochastic billiards on general tables. *Annals of Applied Probability*, 419–437.

Galperin, G., & Zemlyakov, A. (1990). *Mathematical billiards*. Moscow (in Russian): Nauka.

Gilks, W., Richardson, S., & Spiegelhalter, D. (1996). *Markov chain Monte Carlo*. Chapmen and Hall.

Kozlov, V. V., & Treshchev, D. V. (1991). Billiards: A genetic introduction to the dynamics of systems with impacts. *Translations of mathematical monographs* (Vol. 89). Providence, RI: American Mathematical Society.

Lovasz, L., & Vempala, S. (2004). Hit-and-run from a corner. In *Proceedings of the 36th annual ACM symposium on theory of computing, Chicago, IL, USA* (pp. 310–314).

Lovasz, L., & Deak, I. (2012). Computational results of an $O^*(n^4)$ volume algorithm. *European Journal of Operational Research, 216*, 152–161.

Lovasz, L., & Somonovits, M. (1993). Random walks in a convex body and an improved volume algorithm. *Random Structures & Algorithms, 4*(4), 359–412.

Nesterov, Y., & Nemirovsky, A. (1994). *Interior point polynomial methods in convex programming*. Philadelphia: SIAM.

Polyak, B., & Gryazina, E. (2008). Hit-and-Run: New design technique for stabilization, robustness and optimization of linear systems. In *Proceedings of the IFAC World Congress, Seoul, South Korea* (pp. 376–380).

Polyak, B., & Gryazina, E. (2010). Markov Chain Monte Carlo method exploiting barrier functions with applications to control and optimization. In *IEEE multi-conference on systems and control* (pp. 1553–1557).

Polyak, B., & Gryazina, E. (2011). Randomized methods based on new Monte Carlo schemes for control and optimization. *Annals of Operational Research, 189*(1), 343–356.

Rubinstein, R., & Kroese, D. (2008). *Simulation and the Monte Carlo method*. NJ: Wiley.

Sinai, Y. G. (1970). Dynamical systems with elastic reflections. *Russian Mathematical Surveys, 25*(2), 137–189.

Sinai, Y. G. (1978). Billiard trajectories in a polyhedral angle. *Russian Mathematical Surveys, 33*(1), 219–220.

Smith, R. (1984). Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research, 32*(6), 1296–1308.

Tabachnikov, S. (1995). *Geometry and billiards*. RI: Amer. Math. Soc.

Tempo, R., Calafiore, G., & Dabbene, F. (2004). *Randomized algorithms for analysis and control of uncertain systems, communications and control engineering series*. London: Springer-Verlag.

Tervonen, T., van Valkenhoef, G., Basturk, N., & Postmus, D. (2013). Hit-and-run enables efficient weight generation for simulation-based multiple criteria decision analysis. *European Journal of Operational Research, 224*, 552–559.

Turchin, V. (1971). On the computation of multidimensional integrals by the Monte Carlo method. *Theory of Probability and its Applications, 16*(4), 720–724.