# Walks around Monte-Carlo

Boris Polyak

with Elena Gryazina

Institute for Control Science, Moscow

Seminar MFTI

March, 2012

# Monte Carlo: beginning

J. von Neumann, E. Teller, S. Ulam and N. Metropolis (1949)

# First applications

- Simulation (physical processes of particle motion and binary collisions).
- Multidimensional integration.
- Computational mathematics.

# Recent applications

- Convex and global optimization
- Robustness analysis in control and optimization.
- Sampling in complex domains.
- Data mining, Learning, Signal processing, Image processing

# Goal: Sampling in complicated domains

Possible approaches:

- Special regions (box, simplex, $l_p$-ball, positive definite matrices cone)
- Rejection
- Markov chain Monte Carlo (MCMC)– random walk in the domain

# 1D Sampling

Long ago: Tables of random numbers, physical devices (analogs of roulette or coin).

Now there are highly effective pseudo-random generators, fast and with good statistical properties. Example: code *rand* in Matlab generates points uniformly distributed in $[0, 1]$.

How to generate random variables with given cdf $F(x)$? Solution: $x = rand, y = F^{-1}(x)$.

Normal $N(0, 1)$: $randn$ or $x_1 = rand, x_2 = rand$,
$y_1 = \sqrt{-2 \log x_1} \cos(2\pi x_2)$,
$y_2 = \sqrt{-2 \log x_1} \sin(2\pi x_2)$,
$y \sim N(0, I_2)$.

# nD Sampling

1. $rand(n,1)$ - uniform distribution in the box $B = [0,1]^n \in R^n$.

2. $randn(n,1) \sim N(0,I)$.

Exercises: how to generate samples uniformly distributed in 1) sphere 2) ball 3) simplex 4) $l_p$ ball 5) ellipsoid 6) matrix balls

General problem: given a bounded set $Q \in R^n$, how to generate points uniformly distributed in $Q$? Typical example — $Q$ is a polytope.

Rejection: Take simple $G \supseteq Q$ (e.g. a box), generate points uniformly in $G$, reject those which are not in $Q$. Example $Q$ is a unit ball, $G$ is a box $[-1,1]^n$. Then for $n = 2k$ we have $Vol(G)/Vol(Q) = k!2^k/\pi^k = 4{\cdot}10^7$ for $n = 20$, that is we should generate $10^8$ points to get 1–2 points in $Q$. For polytopes this ratio can be much larger.

# Markov Chain Monte Carlo (MCMC)

Idea — apply random walks (vs independent generation of points in MC).

P.Diaconis, The Markov Chain Monte Carlo Revolution, Bull. AMS, Vol. 46, No. 9, 179–205 (2009).

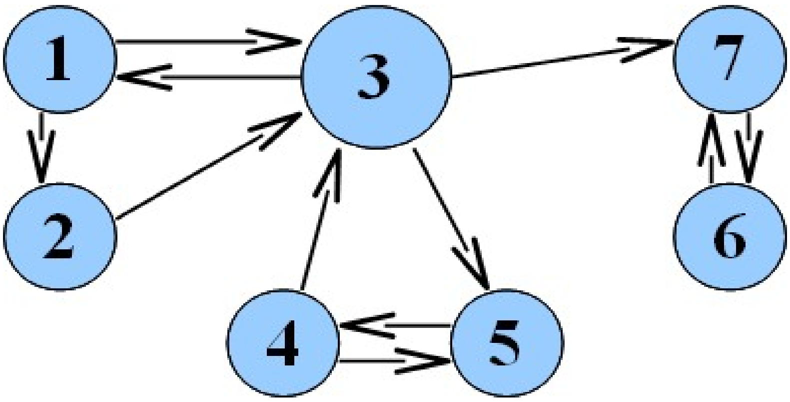
A lot of applications in applied math.

# Random walks in graphs

Given oriented graph $G(V, E), V$ - vertices, $E$ - edges, and $p_{ij}$ - probabilities of transition from vertex $j$ to vertex $i$, $P = ((p_{ij}))$ - transition matrix which is column stochastic ($p_{ij} \geq 0$, $\sum_i p_{ij} = 1$) (nonstandard notation!)

Problem: if there exist limiting probability distribution $x^* = Px^*, x^* \in S, S$ - standard simplex and if power method $x^{k+1} = Px^k$ converges to $x^*$. This is well known in Markov chain theory: $x^*$ always exists, and if $p_{ij} > 0 \quad \forall i, j$ then it is unique and $||x^k - x^*|| \leq c|\lambda_2|^k$, where $\lambda_2$ is the second eigenvalue of $P$. Note that $x_i^*$ can be interpreted as average time of visiting vertex $i$ under random walk.

# Example




$$P = \begin{pmatrix} 0 & 0 & 1/3 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1/3 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1/3 & 0 & 0 & 1 & 0 \end{pmatrix} \qquad x^* = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1/2 \\ 1/2 \end{pmatrix}$$

# Uniform distribution on the graph

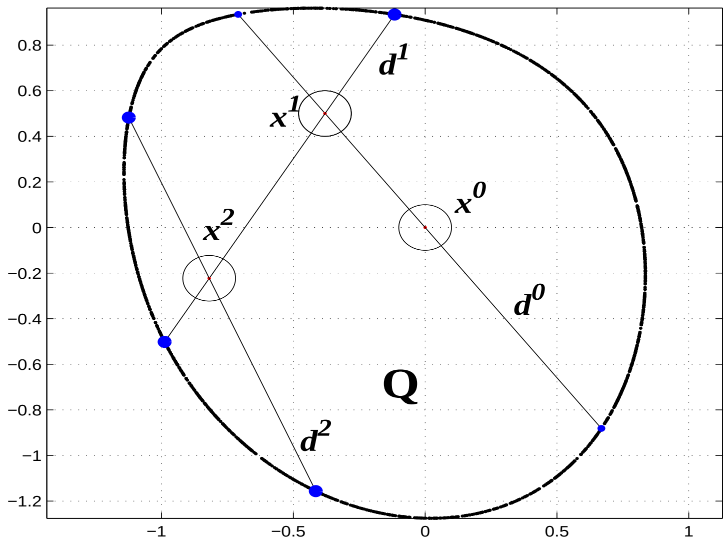
How to make limiting distribution $x^*$ uniform on $G$, i.e. $x^* = e, e_i = 1/n, n$ is the number of vertices?

Note that $P^T e = e$ and for $y^{k+1} = P^T y^k$ we have $y^k \to e$. Thus if $P = P^T$ and $p_{ij} > 0$ then invariant distribution is uniform. How to achieve this if for original transition probabilities $p_{ij} \neq p_{ji}$?

Metropolis random walk. At vertex $j$ choose vertex $i$ randomly with probability $p_{ij}$. Then proceed to $i$ with probability $q = \frac{p_{ji}}{p_{ij} + p_{ji}}$ and stay in $j$ with probability $1 - q$. For such random walk transition probabilities are symmetric and average time of visiting each vertex is the same.

# Hit-and-Run algorithm (sampling in $Q \subset \mathbb{R}^n$)

This is random walk in $Q$. Turchin (1971), Smith (1984)




1. Initial point $x^0 \in Q$.
2. $d = s/||s||, s = randn(n, 1)$ — random direction uniform on the unit sphere
3. Boundary oracle: $L = \{t \in \mathbb{R} : x^0 + td \in Q\}$
4. Next point $x^1 = x^0 + t_1 d$, $t_1$ is uniform random in $L$.
5. $x^0$ is replaced by $x^1$, go to Step 2.

**Theorem** Points $x^k$ are asymptotically uniform in $Q$, i.e., the probability to reach a subset $A \subset Q$ can be estimated as: $|P_k(A) - P(A)| \leq cq^k$, where $P(A) = Vol(A)/Vol(Q)$, $P_k(A) =$ (number of visits of$xA)/k)$ where $q < 1$ does not depend on the initial point $x^0$ (but it depends on the dimension and the geometry of the set).

Idea: for any sets $A \subset Q$, $B \subset Q$ with nonzero volume the probabilities of transition are equal and positive.

# Advantages

- Very simple.
- $Q$ should not be neither convex nor connected (boundary oracle may consist of several intervals), the only assumption is that $Q$ is a closure of an open set.
- Points on the boundary are also generated.
- Boundary oracle is available for numerous sets.

# Boundary oracle is available for numerous sets

- Linear algebraic inequalities set (a polytope)

$$Q = \left\{ x \in \mathbb{R}^n : c_i^T x \le a_i,\ i = 1, \ldots, m \right\}$$

- LMI set

$$Q = \left\{ x \in \mathbb{R}^n : A_0 + \sum_{i=1}^{n} x_i A_i \le 0 \right\}$$

- LMI constrained set of symmetric matrices $P$

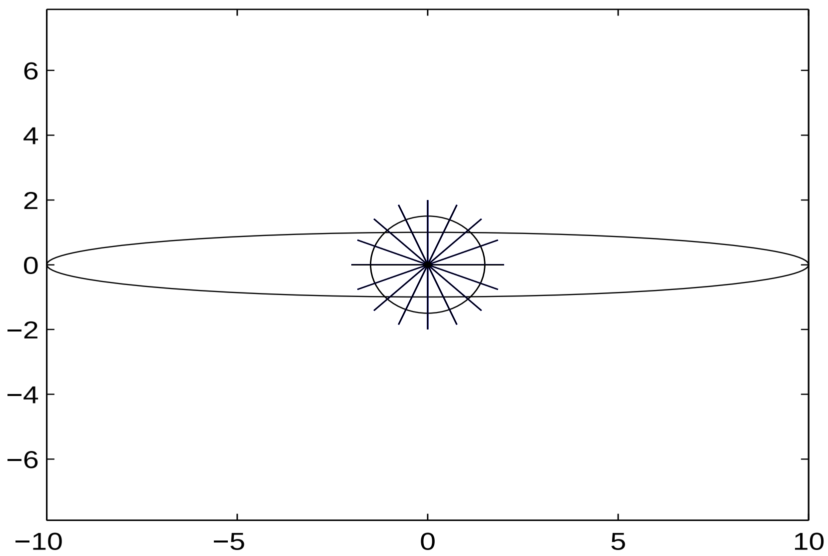
$$Q = \left\{ P : AP + PA^T + C \le 0,\ P \ge 0 \right\}$$

- Quadratic matrix inequalities set

$$Q = \left\{ P : AP + PA^T + PBB^T P + C \le 0,\ P \ge 0 \right\}$$

# Drawbacks

- HR jams in corners.
- HR jams for narrow bodies.
- Strong serial correlation.




(Lovasz, Vempala. Hit-and-Run from a corner, 2007) necessary number of iterations to achieve uniformity with precision $\varepsilon$

$$N > 10^{10} \frac{n^2 R^2}{r^2} \ln \frac{M}{\varepsilon}$$

# How to accelerate convergence?

Kaufman, Smith. Direction choice for accelerated convergence in Hit-and-Run sampling // Operations Research, Vol. 46, No. 1, 1998, p. 84-95.

$d$ – uniform on the sphere $\to$ some other distribution $H$

Kannan and Narayanan. Random walks on polytopes and an affine interior point method
for linear programming // Proc. of ACM Symp. on Theory of Computing, 2009.

# Barrier function machinery

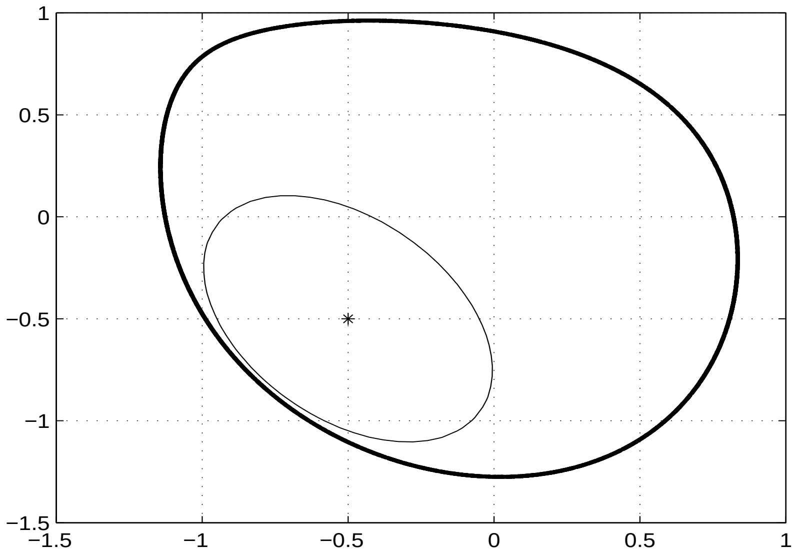
$F(x)$ – self-concordant barrier for $Q$,

$F(x)$ is defined in $\mathrm{int}Q$ and $F(x) \to \infty$ as $x \to \partial Q$.

Dikin ellipsoid

$$E_x = \{y : (\nabla^2 F(x)(y - x), y - x) \le 1\} \subset Q$$

$\nabla^2 F(x) \succ 0$ for all $x \in Q$, suppose $(\nabla^2 F(x))^{-\frac{1}{2}}$ can be calculated.



# New: Barrier Hit-and-Run method

Given $x \in \mathrm{Int}Q$, barrier $F(x)$, $i = 1$.

1. $x = x^i$.

2. Pick random direction $d$ uniformly in $E_x$:

   $$d_x = (\nabla^2 F(x))^{-1/2}\xi, \quad \xi - \text{ is uniform random in the unit ball.}$$

3. Pick $y = x + \lambda d$, where $\lambda \in [\underline{\lambda}, \overline{\lambda}]$ is defined via Boundary Oracle as in the original version of Hit-and-Run.

4. Calculate $r_x = \dfrac{||d||}{\sqrt{(H_x d, d)}}$, $H_y = \nabla^2 F(y)$ and $r_y = \dfrac{||d||}{\sqrt{(H_y d, d)}}$.

5. Accept $y$ with probability $\min\left(1, \left(\dfrac{r_y}{r_x}\right)^n \cdot \dfrac{vol(E_x)}{vol(E_y)}\right)$, $i = i + 1$, $x^i = y$. Go to 1.

# Theoretical validation

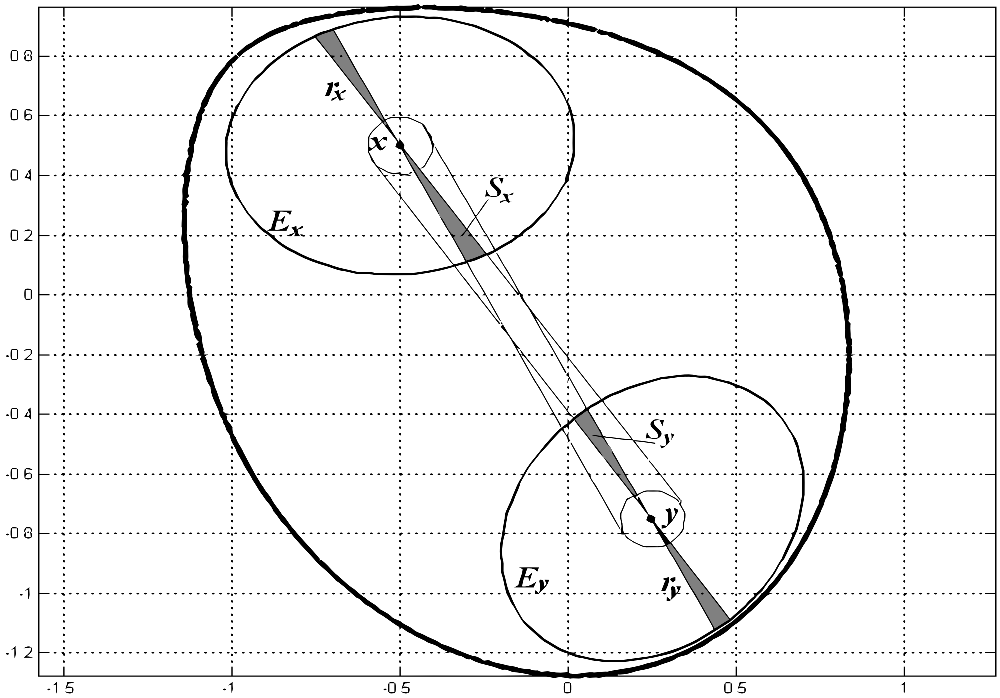
## How can an asymptotical uniformity be guaranteed?

1. There exists a probability density $p(y|x) > 0$ for all $x, y \in Q$.
2. Reversibility

$$p(x|y) = p(y|x)$$

We check both properties for our algorithm.

**Theorem** BHR algorithm provides asymptotically uniformly distributed points for convex set $Q$.




$$P_{x\to\varepsilon(y)} = \frac{vol(S_x)}{vol(E_x)} \cdot \frac{\varepsilon}{L_{xy}} = c \cdot r_x^n \sqrt{\det H_x} \cdot \alpha = \varphi_x \alpha,$$ where $\alpha$ is a probability to accept point $y$ as a new generated point.

$P_{y\to\varepsilon(x)} = \varphi_y \beta$, where $\beta$ is a probability to transfer to point $x$ starting from $y$.

**Metropolis rule:** taking $\alpha = \min\left\{1, \dfrac{\varphi_y}{\varphi_x}\right\}$,

$\beta = \min\left\{1, \dfrac{\varphi_x}{\varphi_y}\right\}$ we obtain $P(x \to B_y) = P(y \to B_x)$.

# How it works for polytopes

$$Q = \{x : Ax \leq b\}, A : m \times n$$

Barrier function and its Hessian

$$F(x) = -\sum \ln(b_i - (a^i, x)), \quad a^i - \text{the rows of matrix } A$$

$$H = \nabla^2 F(x) = A^T D^2 A, \quad D = \operatorname{diag}\left\{ \frac{1}{b_i - (a^i, x)} \right\}$$

Random uniform point in Dikin ellipsoid:

just solve a system of linear equations!

$$Hd = A^T D\xi, \quad \xi - \text{ random vector on the unit sphere}$$

# Comparison with uniform distribution

For the given $Q$ and samples $x^i \in Q$ take arbitraty function $f(x)$ for which cdf-function

$$\Phi(t)$$

can be calculated in a closed form.

Construct an empirical cdf

$$\hat{\Phi}(t) = \left\{ \frac{\text{number of points } x^i : f(x^i) \leq t}{N} \right\}$$

Compare $\Phi(t)$ and $\hat{\Phi}(t)$

# Kannan&Narayanan method (KN)

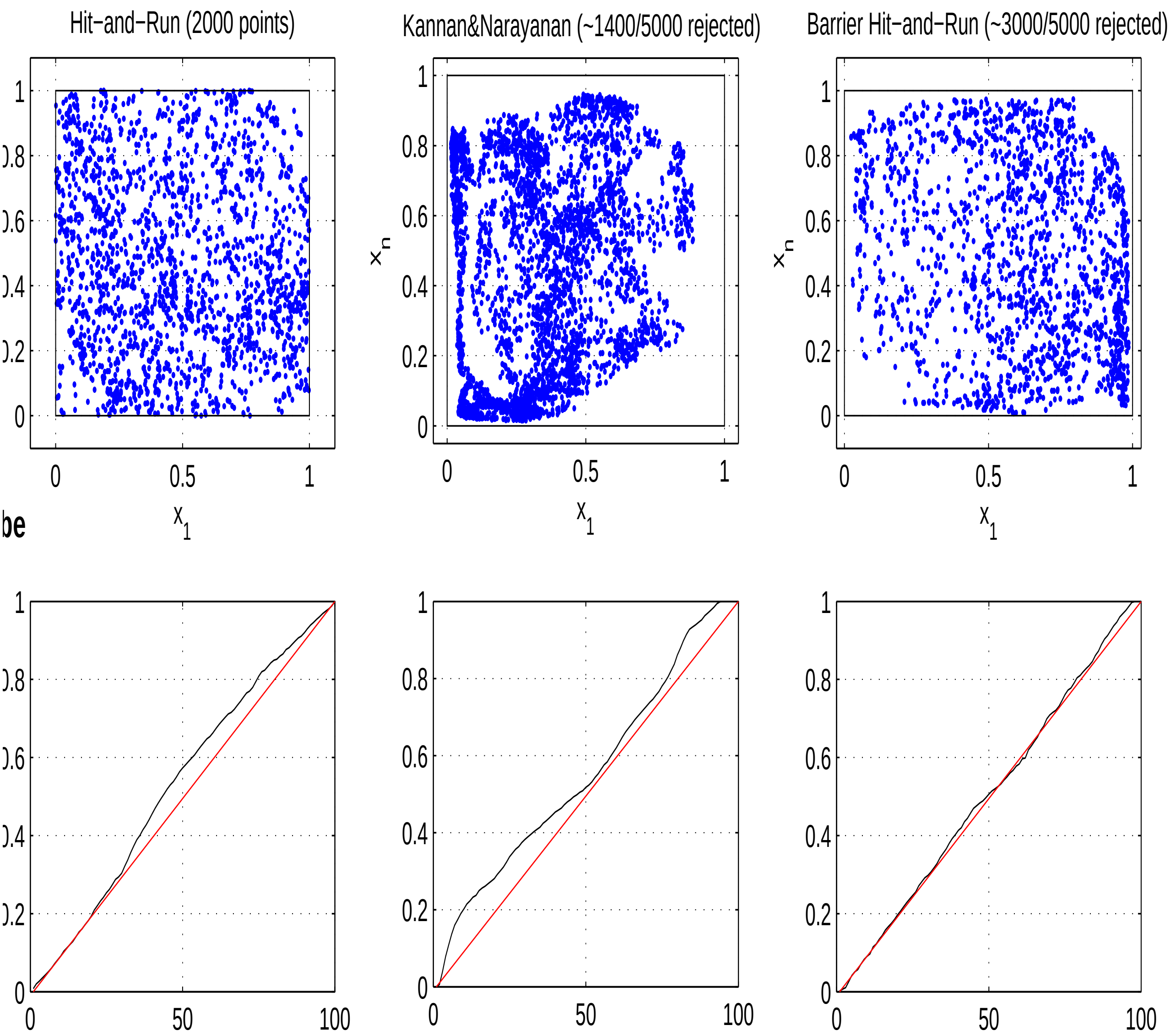
$$E_x = \{y : (\nabla^2 F(x)(y-x), y-x) \leq \frac{3}{40}\}$$

1. Flip a fear coin. If Heads, stay at $x$.
2. Otherwise, pick random $y \in E_x$.
3. If $x \notin E_y$, then stay at $x$;

   if $x \in E_y$ accept $y$ (replace $x$) with probability $\min\left(1, \frac{vol(E_x)}{vol(E_y)}\right)$

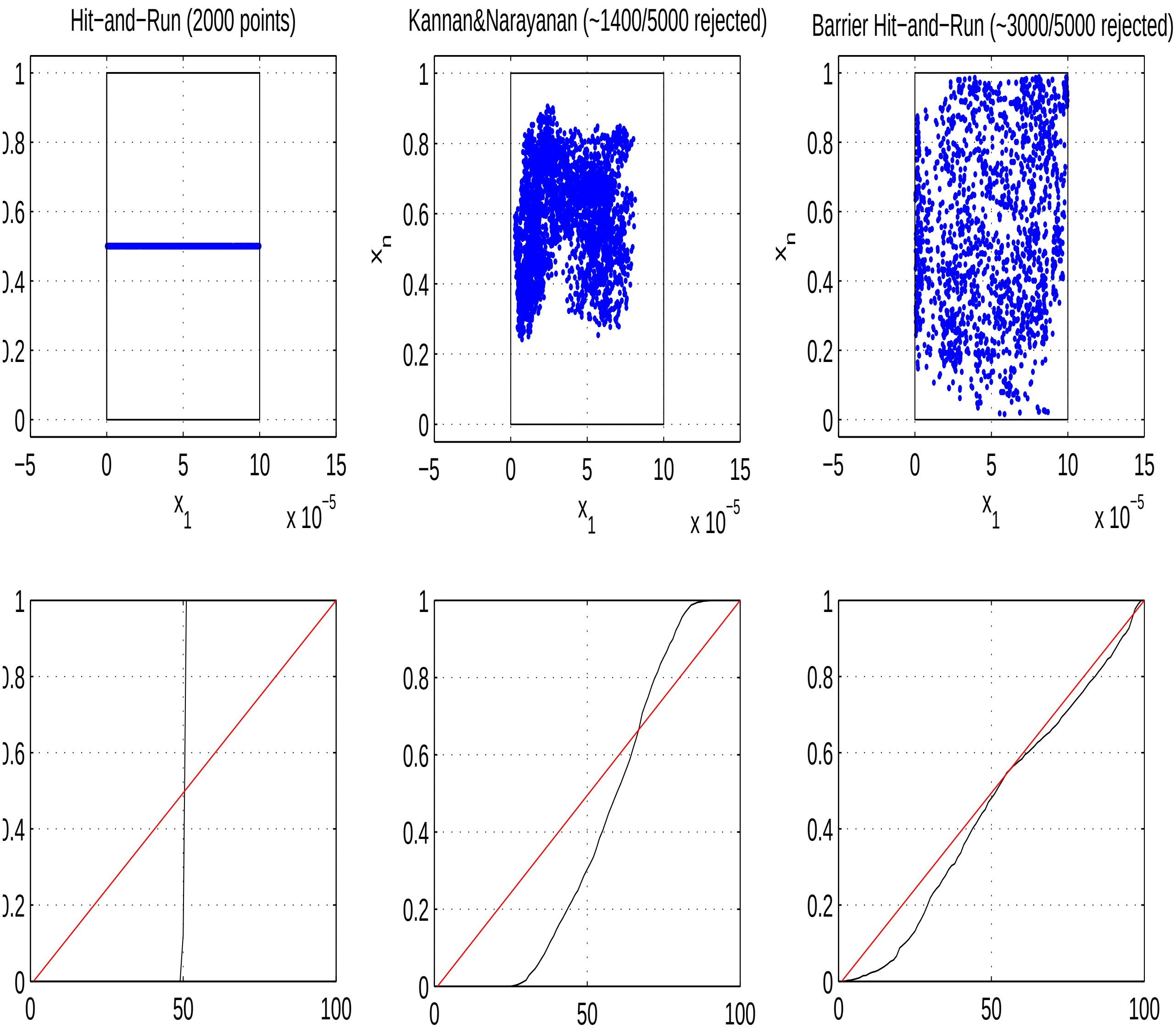
   $\min\left(1, \sqrt{\frac{det(H_y)}{det(H_x)}}\right)$. Go to step 1.

# Simulations: $\mathbb{R}^{10}$-cube $f = x_{10}$

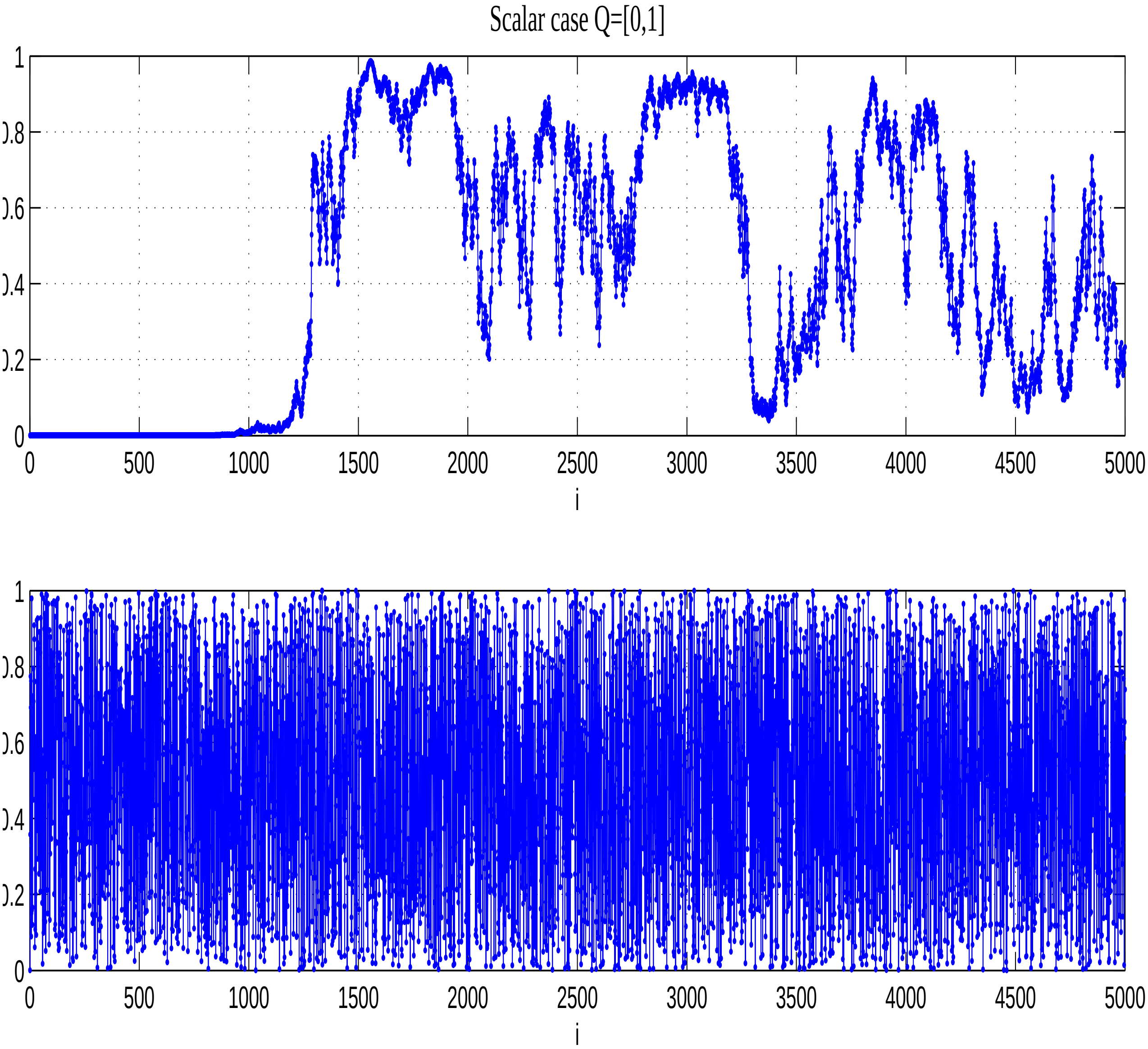


# Simulations: $\mathbb{R}^{10}$-stick

$$Q = \{0 \le x_i \le 10^{-4}, i = 1, \ldots, 9; 0 \le x_{10} \le 1\},\ f = x_{10}$$



# Simulations: scalar case



# Comparison with other methods

- BHR and KN are both affine invariant.

  The behavior of the trajectories is the same for $Q$ for its affine transformation $T(Q)$.

- There exists polynomial estimate of convergence rate for KN method: $N > 7 \times 10^8 \times mn \left( n \ln(20s\sqrt{m}) + \ln\left(\frac{32}{\epsilon}\right)\right)$

  [for $n = 10$, $m = 20$ – $N \sim 10^{13}$ iterations]

- Strong serial correlation for KN method and its absence for BHR.

- BHR: large steps are allowed.

- Original KN method is proposed for polytopes only*, BHR for arbitrary convex set with barrier and Boundary Oracle.

* Hariharan Narayanan. Randomized interior point methods for sampling and optimization (preprint in ArXiv) New version of KN algorithm for arbitrary convex set with a self-concordant barrier.

# Shake-and-Bake: an alternative technique

Idea: billiards with random reflections

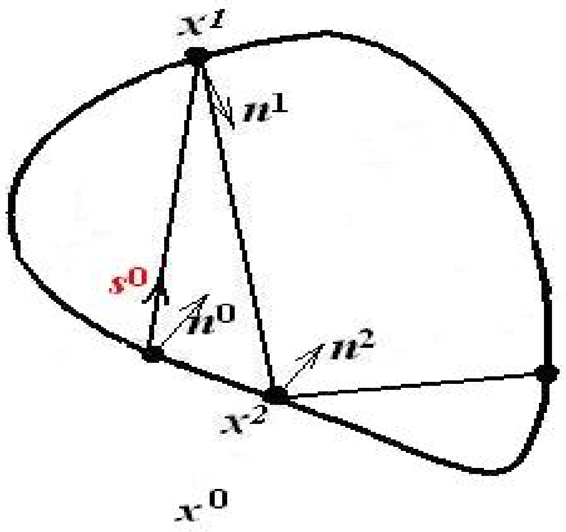
K.A. Borovkov (1991) "On a new variant of the Monte Carlo method Theory of Probability and its Applications, Vol. 36, No. 2, pp. 355-360; (1994) "On simulation of random vectors with given densities in regions and on their boundaries Journal of Applied Probability, Vol. 31, No.1, pp. 205-220.

F. Comets, S. Popov, G.M. Schutz, M. Vachkovskaia "Billiards in a general domain with random reflections"/arXiv:math/0612799v3

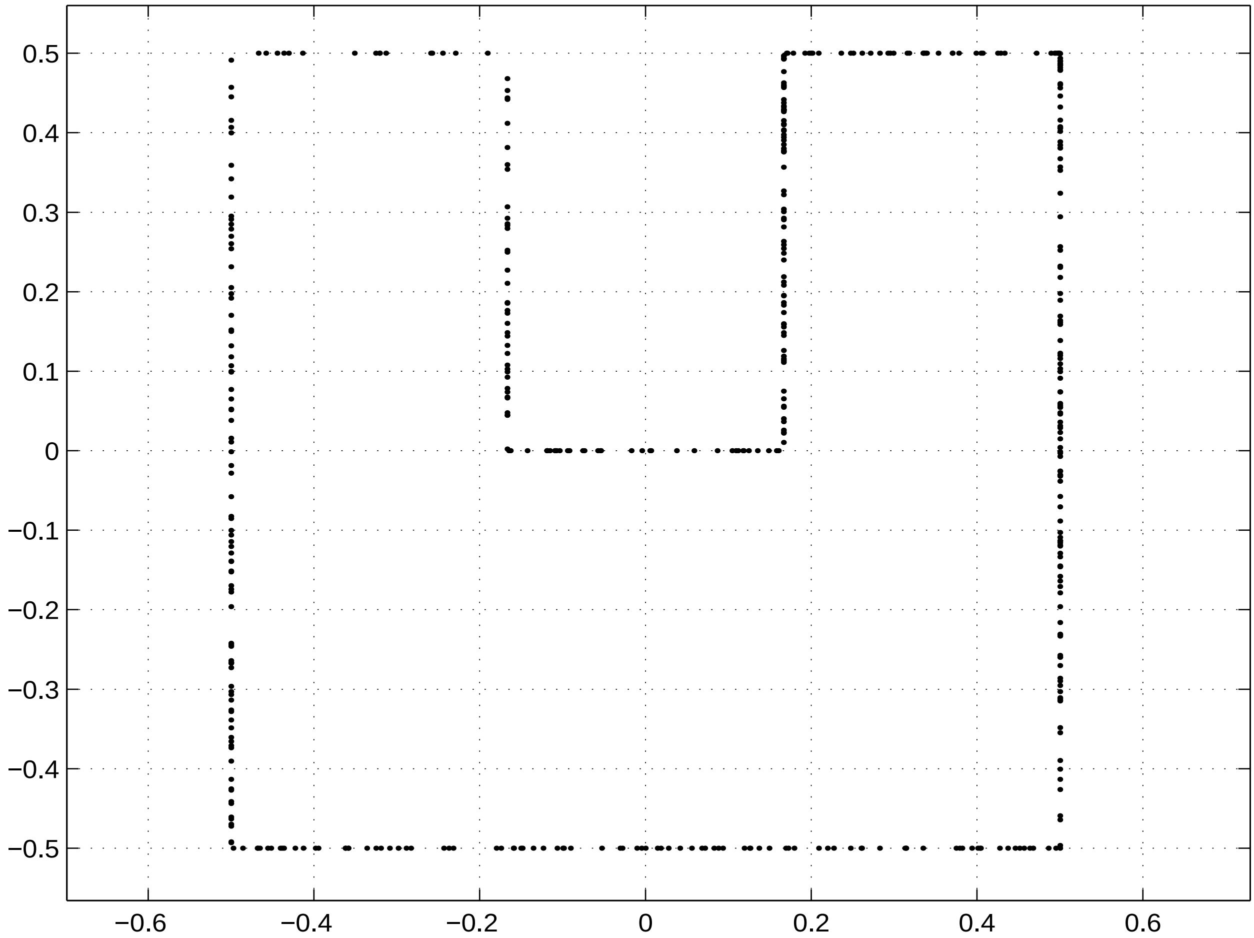
- Points are asymptotically uniformly distributed in the boundary of $Q$.
- Boundary oracle is used (first intersection with the boundary).
- Normals to the boundary are needed.
- The set $Q$ should be connected, but not necessarily convex.

# Shake-and-Bake: the algorithm

1. $i = 0$, $x^0 \in \partial X$, $n^0$ is the normal.
2. Choose random direction $s^i$, $s^i = \sqrt{1 - \xi^{\frac{2}{n-1}}}\, n^0 + r$,
   $\xi$ uniform random in $(0, 1)$,
   $r$ is random unit uniform direction $(n^0, r) = 0$.
3. $x^{i+1} = x^i + \bar{t}s$,
   $\bar{t}$ is given by the boundary oracle for the direction $s$.
4. $x^{i+1} \Rightarrow x^i$, go to Step 2.



# Shake-and-Bake for nonconvex sets



# Open problems

- Parallel versions of Hit-and-Run and Shake-and-Bake.
- How to generate direction $x$, $||x|| = 1$ uniformly in a polyhedral cone $Ax \leq 0$? How to exploit it in HR and SB?
- Multistep HR. From a point $x^0 \in Q$ take several directions $d^i$ (uniformly distributed on the unit sphere) and find corresponding $t_i$ via boundary oracles. Then choose $i$-th direction with probability proportional to $t_i^n$. Probably this method is much faster than the standard HR.
- Random walks on discrete grids.
- Other ideas for points generation — triangulization, approximation by ellipsoids or polytopes etc.