

## ePub<sup>WU</sup> Institutional Repository

Roman Karawatzki and Josef Leydold and Klaus Pötzelberger

Automatic Markov Chain Monte Carlo Procedures for Sampling from Multivariate Distributions

Working Paper

*Original Citation:*

Karawatzki, Roman and Leydold, Josef and Pötzelberger, Klaus (2005) Automatic Markov Chain Monte Carlo Procedures for Sampling from Multivariate Distributions. *Research Report Series / Department of Statistics and Mathematics, 27*. Department of Statistics and Mathematics, WU Vienna University of Economics and Business, Vienna.

This version is available at: <http://epub.wu.ac.at/1400/>

Available in ePub<sup>WU</sup>: December 2005

ePub<sup>WU</sup>, the institutional repository of the WU Vienna University of Economics and Business, is provided by the University Library and the IT-Services. The aim is to enable open access to the scholarly output of the WU.

# Automatic Markov Chain Monte Carlo Procedures for Sampling from Multivariate Distributions



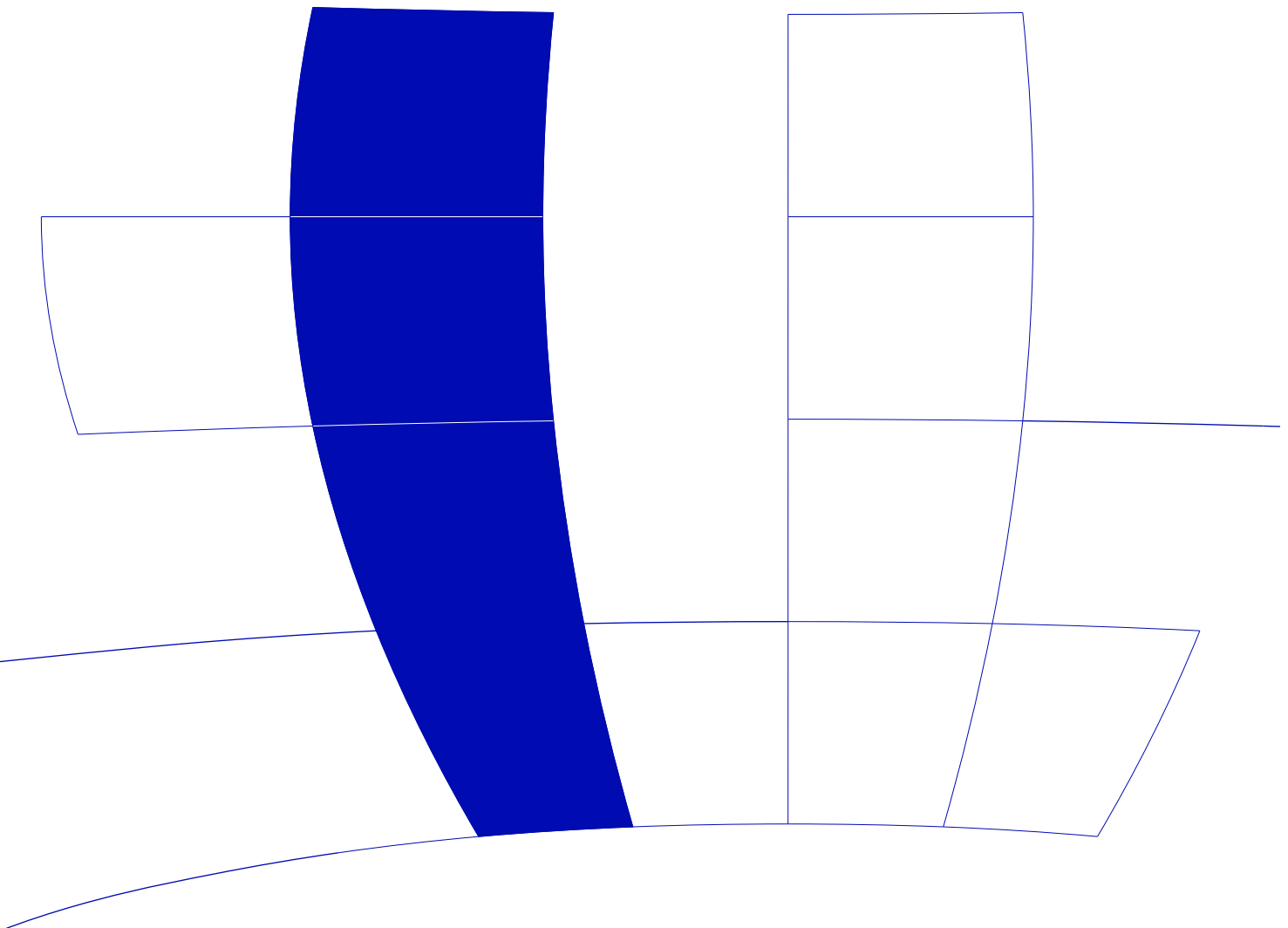
Roman Karawatzki, Josef Leydold, Klaus Pötzelberger

Department of Statistics and Mathematics  
Wirtschaftsuniversität Wien

**Research Report Series**

Report 27  
December 2005

<http://statmath.wu-wien.ac.at/>



# Automatic Markov Chain Monte Carlo Procedures for Sampling from Multivariate Distributions

ROMAN KARAWATZKI, JOSEF LEYDOLD, AND KLAUS PÖTZELBERGER

Department of Statistics and Mathematics

Vienna University of Economics and Business Administration

---

Generating samples from multivariate distributions efficiently is an important task in Monte Carlo integration and many other stochastic simulation problems. Markov chain Monte Carlo has been shown to be very efficient compared to “conventional methods”, especially when many dimensions are involved. In this article we propose a *Hit-and-Run* sampler in combination with the *Ratio-of-Uniforms* method. We show that it is well suited for an algorithm to generate points from quite arbitrary distributions, which include all log-concave distributions. The algorithm works automatically in the sense that only the mode (or an approximation of it) and an oracle is required, i.e., a subroutine that returns the value of the density function at any point  $x$ . We show that the number of evaluations of the density increases slowly with dimension.

Categories and Subject Descriptors: G.3 [**Probability and Statistics**]: Random number generation

General Terms: Algorithms

Additional Key Words and Phrases: hit-and-run sampling, Markov chain Monte Carlo, multivariate random variate generation, ratio-of-uniforms method, log-concave distributions

---

## 1. INTRODUCTION

Sampling random vectors is an important part of many stochastic simulations and randomized algorithms. When many dimensions are involved this becomes a very challenging task. Conventional methods that have been developed for the case of univariate random numbers like rejection or composition do not work efficiently for distributions with moderately many (8 or more) dimensions (see e.g. Hörmann et al. [2004]). Markov chain Monte Carlo methods have proven to be more efficient for higher dimensions. The algorithms generate correlated sequences of random points that converge to the target distribution.

In this article we propose the *Hit-and-Run* sampler in combination with the *Ratio-of-Uniforms* method. It is well suited for an algorithm to generate points from quite arbitrary distributions, which include all log-concave distributions. This algorithm works automatically in the sense that only the mode (or an approximation of it) and an oracle is required, i.e., a subroutine that returns the value of the density function at any point  $\mathbf{x}$ . We use the number of calls to the oracle as a measure of the performance of the algorithm. Some theoretical results and our computational experiences show that this method is fast and its performance decreases only slowly with increasing dimension such that samples from distributions

---

Author’s address: Vienna University of Economics and Business Administration, Department of Statistics and Mathematics, Augasse 2-6, A-1090 Vienna, Austria

email: leydold@statistik.wu-wien.ac.at

This work was supported by the Austrian Science Foundation (FWF), project no. P16767-N12.

with 100 or more variables can be generated. One should notice, however, that the costs for evaluating the density itself depends on the dimension, e.g., it increases quadratically for a multinormal distribution.

The article is organized as follows: First we shortly describe the Hit-and-Run sampler (Sect. 2) and state the main facts about the Ratio-of-Uniforms method (Sect. 3). In Section 4 we propose new algorithms based on these two principles. Our computational experiences are summarized in Sect. 5.

## 2. THE HIT-AND-RUN SAMPLER

For the problem of sampling random points uniformly distributed in some fixed but arbitrary bounded open set  $S \in \mathbb{R}^n$  Smith [1984] introduced the so called *Hit-and-Run* samplers that are based on the following principle.

0. Choose a starting point  $\mathbf{X}_0 \in S$  and set  $k = 0$ .
1. Generate a random direction  $\mathbf{d}_k$  with distribution  $\nu$ .
2. Generate  $\lambda_k$  uniformly distributed in  $\Lambda_k = \{\lambda: \mathbf{x}_k + \lambda \mathbf{d}_k \in S\}$ .
3. Set  $\mathbf{X}_{k+1} = \mathbf{X}_k + \lambda_k \mathbf{d}_k$  and  $k = k + 1$ .
4. Goto 1.

Boneh and Golan [1979] and independently Smith [1980] first considered such an algorithm with  $\nu$  being the uniform distribution over a hypersphere (*hypersphere direction sampling*). It has been shown that this Markov chain is mixing fast [Smith 1984; Chen and Schmeiser 1993], that is, the distribution of the generated point set converges to the uniform distribution with increasing sample size; in particular when the set  $S$  is convex [Lovász 1999; Lovász and Vempala 2004]. Kaufman and Smith [1998] improved the convergence of the Hit-and-Run sampler by non-uniform direction choice and give a (worst case) optimal distribution  $\nu$ . We restate the convergence result by Smith [1984].

**THEOREM 2.1** [SMITH 1984]. *Let  $\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2, \dots$  be the Markov chain generated by the Hit-and-Run Algorithm over an open bounded region  $S \in \mathbb{R}^n$ . Then for any measurable set  $A \subseteq S$ ,*

$$|\mathbb{P}(\mathbf{X}_m \in A | \mathbf{X}_0 = \mathbf{x}) - \mu(A)| < (1 - (\gamma/n2^{n-1}))^{m-1}$$

where  $\mu(\cdot)$  denotes the  $n$ -dimensional content of  $A$  and  $\gamma$  is the ratio of the  $n$ -dimensional content of  $S$  to the  $n$ -dimensional content of the smallest sphere containing  $S$ .

Another variant suggested by Telgen (see Smith [1980]) is to use random coordinate directions (*coordinate direction sampling*) which can be seen as Gibbs sampling with randomized directions. This algorithm has considerable computational advantages over random direction sampling on a per iteration basis. However, it is harder to show convergence results [Berbee et al. 1987; Smith 1980] and it does not always converge at a geometric rate [Bélisle et al. 1998].

The Hit-and-Run sampler can easily be generalized to sample from non-uniform distributions with density  $f$  [Belisle et al. 1993; Chen and Schmeiser 1996; Smith 1996; Bélisle et al. 1998]: Replace the uniform distribution on  $\Lambda_k$  in Step 2 of the above algorithm by the conditional distribution of  $f$ :

2'. Draw  $\lambda_k \in \Lambda_k = \{\lambda: \mathbf{x}_k + \lambda \mathbf{d}_k \in S\}$  from the distribution with density

$$f_k(\lambda) = \frac{f(\mathbf{x}_k + \lambda \mathbf{d}_k)}{\int_{\Lambda_k} f(\mathbf{x}_k + \theta \mathbf{d}_k) d\theta}, \quad \lambda \in \Lambda_k. \quad (1)$$

Notice that this generalized Hit-and-Run sampler can also be seen as a generalized Gibbs sampler: In each step replace sampling from the full conditional distribution along a coordinate direction by sampling from the full conditional distribution along a randomly chosen direction.

Although it has been shown that this algorithm has fast mixing time as well [Bélisle et al. 1998; Lovász and Vempala 2003], there are some drawbacks. Besides the existence of distributions where it converges arbitrarily slow [Bélisle 2000], the necessity of sampling from the conditional density (1) makes it difficult to apply for user-defined distributions. Notice that only one random variate has to be drawn from a particular conditional density. In the last decade automatic algorithms for sampling from large classes of univariate distributions have been developed [Gilks and Wild 1992; Hörmann et al. 2004]. However, these require some setup which can be quite expensive both in time and memory compared to the marginal generation time and thus they are often extremely slow when only one random variate should be generated. Moreover, many of them require the knowledge of some parameters of the (univariate) distributions (e.g. its mode).

### 3. THE RATIO-OF-UNIFORMS METHOD

The Ratio-of-Uniforms method has been introduced by Kinderman and Monahan [1977] and generalized to the multivariate case by Vaduva [1984], Stefănescu and Vaduva [1987], and Wakefield et al. [1991]. It is based on the following theorem.

**THEOREM 3.1** [WAKEFIELD ET AL. 1991]. *Let  $f(\mathbf{x})$  be a positive integrable function on  $\mathbb{R}^n$ . Let  $r > 0$  and  $\mathbf{m} \in \mathbb{R}^n$  be constants. Suppose the point  $(\mathbf{U}, V) \in \mathbb{R}^{n+1}$  with  $\mathbf{U} = (U_1, \dots, U_n)$  is uniformly distributed over the region*

$$\mathcal{A}(f) = \mathcal{A}_{r,\mathbf{m}}(f) = \left\{ (\mathbf{u}, v): 0 < v < \sqrt[rn+1]{f(\mathbf{u}/v^r + \mathbf{m})} \right\}, \quad (2)$$

then  $\mathbf{X} = \mathbf{U}/V^r + \mathbf{m}$  has probability density function  $f(\mathbf{x}) / \int_{\mathbb{R}^n} f(\mathbf{z}) d\mathbf{z}$ .

The proof of this theorem is based on the fact that the map

$$(\mathbf{u}, v) \mapsto (\mathbf{x}, y) = \left( \frac{\mathbf{u}}{v^r} + \mathbf{m}, v^{rn+1} \right) \quad (3)$$

has constant Jacobian (equal to  $rn + 1$ ). We can apply this theorem and get the following algorithm. Notice that  $f$  need not be normalized, i.e., it can be any positive multiple of a density function.

1. Generate a point  $(\mathbf{U}, V)$  uniformly in  $\mathcal{A}_{r,\mathbf{m}}(f)$ .
2. Return  $\mathbf{X} = \mathbf{U}/V^r + \mathbf{m}$ .

An important observation is that the region  $\mathcal{A}(f)$  is bounded for many distributions (at least for sufficiently large values of  $r$ ). The originally proposed algorithm

uses rejection from the minimal bounding rectangle  $\mathcal{R}_{r,\mathbf{m}}(f)$  which is given by

$$\begin{aligned} v^+ &= \sup_{\mathbf{x}} (f(\mathbf{x}))^{1/(rn+1)}, \\ u_i^- &= \inf_{x_i} (x_i - m_i) (f(\mathbf{x}))^{r/(rn+1)}, \\ u_i^+ &= \sup_{x_i} (x_i - m_i) (f(\mathbf{x}))^{r/(rn+1)}. \end{aligned} \quad (4)$$

Thus  $\mathcal{A}(f)$  is bounded if and only if  $f(\mathbf{x}) = \mathcal{O}(\|\mathbf{x}\|^{-n-1/r})$  for all  $\mathbf{x}$ , or equivalently, if and only if  $\|\mathbf{x}\|^{n+1/r} f(\mathbf{x})$  is bounded.

Usually  $\mathbf{m}$  is set to the mode of the density  $f$  as this results in an (almost) optimal rejection constant. However, the acceptance rate decreases exponentially with the dimension in this simple rejection algorithm (see e.g. Hörmann et al. [2004]) and hence is impractical for dimensions larger than 10. For example, when  $\mathcal{A}(f)$  is a ball then the expected number of points that must be generated within  $\mathcal{R}(f)$  to obtain one within  $\mathcal{A}(f)$  grows from 1.27 for dimension  $n = 1$  to 400 for  $n = 10$ ,  $4 \times 10^7$  for  $n = 20$ , and  $6 \times 10^{27}$  for  $n = 50$ . However, we can use the Hit-and-Run sampler to generate a sequence of uniformly distributed points  $(\mathbf{u}, v)$  in  $\mathcal{A}(f)$  in Step 1 even in high dimensions. As  $\mathcal{A}(f)$  is bounded we can expect that the uniform Hit-and-Run sampler is mixing faster than the corresponding Hit-and-Run sampler on the region

$$\mathcal{G}(f) = \{(\mathbf{x}, y) : 0 < y < f(\mathbf{x})\}. \quad (5)$$

Moreover, the set  $\mathcal{G}(f)$  need not be bounded and thus convergence is not assured.

The parameter  $r$  can be used to control the shape of  $\mathcal{A}(f)$ . For increasing values of  $r$  the family of distributions for which this set is bounded is growing. For the special case  $r = 1$  the region  $\mathcal{A}(f)$  is convex in many cases and thus the Hit-and-Run sampler converges fast. The following theorem generalizes a result for the univariate case [Leydold 2000].

**THEOREM 3.2.** *For a density  $f$  and  $r = 1$  the region  $\mathcal{A}(f) \subset \mathbb{R}^{n+1}$  is convex if and only if the transformed density  $T(f(\mathbf{x})) = -(f(\mathbf{x}))^{-1/(n+1)}$  is concave.*

Following Hörmann [1995] we call such a density  $T_c$ -concave with  $c = -1/(n+1)$ .

**PROOF.** Since  $T(y) = -1/\sqrt[n+1]{y}$  is strictly monotonically increasing, the transformation  $(\mathbf{x}, y) \mapsto (\mathbf{x}, T(y))$  maps  $\mathcal{G}(f)$  one-to-one onto  $\mathcal{T}(f) = \{(\mathbf{x}, y) : y < T(f(\mathbf{x}))\}$ , i.e. the region below the transformed density. Hence by  $T(v^{n+1}) = -1/v$  and transformation (3)

$$\mathbb{R}^n \times (0, \infty) \rightarrow \mathbb{R}^n \times (-\infty, 0), \quad (\mathbf{u}, v) \mapsto (\mathbf{x}, y) = (\mathbf{u}/v + \mathbf{m}, -1/v) \quad (6)$$

maps  $\mathcal{A}(f)$  one-to-one onto  $\mathcal{T}(f)$ . Notice that  $f$  is  $T$ -concave if and only if  $\mathcal{T}(f)$  is convex. Thus it remains to show that  $\mathcal{A}(f)$  is convex if and only if  $\mathcal{T}(f)$  is convex, and consequently if and only if hyperplanes remain hyperplanes under transformation (6). Now let  $\mathbf{a}'\mathbf{x} + by = d$  be a hyperplane in  $\mathcal{T}(f)$ . Then  $\mathbf{a}'(\mathbf{u}/v + \mathbf{m}) - b/v = d$  or, equivalently,  $\mathbf{a}'\mathbf{u} - dv = b - \mathbf{a}'\mathbf{m}$ , is a hyperplane in  $\mathcal{A}(f)$ . Analogously we find for a hyperplane  $\mathbf{a}'\mathbf{u} + bv = d$  in  $\mathcal{A}(f)$  the hyperplane  $\mathbf{a}'\mathbf{x} + dy = -b + \mathbf{a}'\mathbf{m}$  in  $\mathcal{T}(f)$ .  $\square$

The following result by Hörmann [1995] immediately holds for multivariate distributions.

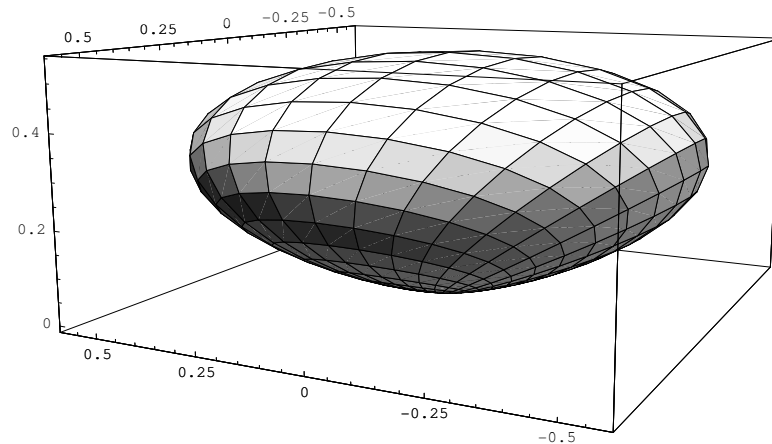


Fig. 1. Region  $\mathcal{A}(f)$  for standard bivariate normal distribution ( $r = 1$ ).

**THEOREM 3.3** [HÖRMANN 1995]. *If a density  $f$  is  $T_c$ -concave for some  $c \in \mathbb{R}$  then  $f$  is  $T_{c_1}$ -concave for all  $c_1 \leq c$ .*

The family of transformations  $T_c$  contains the special case  $T_0(y) = \log(y)$ . Thus we have the following important corollary; see Fig. 1 for an example.

**COROLLARY 3.4.** *For every log-concave density  $f$  and  $r = 1$  the region  $\mathcal{A}(f)$  is convex.*

For densities with higher tails values  $r > 1$  might be necessary. However, we have to note that an analogous (simple) condition for the convexity of  $\mathcal{A}_r(f)$  for  $r \neq 1$  is not known.

#### 4. THE ALGORITHMS

Although we have presented all ingredients of the Hit-and-Run algorithms for sampling from non-uniform multivariate distributions we have to make some considerations about computational details.

##### 4.1 Uniform Sampling

The demanding part of the Hit-and-Run sampler is to sample  $\lambda_k$  uniformly in the interval (or union of intervals)  $\Lambda_k = \{\lambda: \mathbf{x}_k + \lambda \mathbf{d}_k \in \mathcal{A}(f)\}$  for a chosen direction  $\mathbf{d}_k$ . It can be accomplished by the following procedure [Smith 1996]: First the covering interval  $L_k = \{\lambda: \mathbf{x}_k + \lambda \mathbf{d}_k \in \mathcal{R}(f)\} \supseteq \Lambda_k$  is determined. Then a rejection method is employed by generating points uniformly on the line segment  $L_k$  until one falls within  $\Lambda_k$ . We denote the endpoints of the interval  $L_k$  by  $\lambda_{k0}$  and  $\lambda_{k1}$ , i.e.,  $L_k = (\lambda_{k0}, \lambda_{k1})$ . Since this subproblem is one-dimensional the rejection method is typically very efficient for this task.

The acceptance probability in each step of the Hit-and-Run sampler is given by the ratio  $\rho$  of the length of the line segment  $\Lambda_k$  (or the sum of all segments constituting  $\Lambda_k$ ),  $\mu(\Lambda_k)$ , and the length of the line segment  $L_k$ ,  $\mu(L_k)$ , see Hörmann

et al. [2004, §2.2]:

$$\rho = \frac{\mu(\Lambda_k)}{\mu(L_k)}. \quad (7)$$

The expected number of calls to the oracle, i.e., the expected number of iterations is given by its reciprocal  $1/\rho$ .

#### 4.2 Adaptive Uniform Sampling

Although the acceptance rate in a univariate rejection step is much better than those for rejection from the bounding rectangle it can still be improved by the following procedure when  $\Lambda_k$  is an open interval (this is always the case when  $\mathcal{A}(f)$  is convex):

Start with some covering interval  $L_k = (\lambda_{k0}, \lambda_{k1}) \supset \Lambda_k$ . In each iteration generate a point  $\lambda_k$  uniformly in  $L_k$ . If  $\lambda_k \in \Lambda_k$  accept this point and compute  $\mathbf{X}_{k+1}$ . Otherwise, we shrink  $(\lambda_{k0}, \lambda_{k1})$  and try again. This is done by replacing  $\lambda_{k0}$  or  $\lambda_{k1}$  by the generated  $\lambda_k$  depending on the sign of  $\lambda_k$ . (Notice that  $(\lambda_{k0}, \lambda_{k1})$  always contains 0.)

This procedure can be equivalently described as follows: Let  $\lambda_0 \leq 0 < \rho \leq \lambda_1$  with  $\lambda_1 - \lambda_0 = 1$ . We define  $I_k = [\lambda_0^k, \lambda_1^k]$  and random variables  $U_k$ , distributed uniformly on  $I_k$ , recursively. Start with  $\lambda_0^1 = \lambda_0$ ,  $\lambda_1^1 = \lambda_1$  and  $U_1 \sim \mathbb{U}(I_1)$ . For given  $I_k$ , let  $X_k \sim \mathbb{U}(I_k)$  and define  $I_{k+1} = I_k$  if  $U_k \in [0, \rho]$ ,  $I_{k+1} = [U_k, \lambda_1^k]$  if  $U_k < 0$  and  $I_{k+1} = [\lambda_0^k, U_k]$  if  $U_k > \rho$ . Define the stopping time  $T$  as  $T = \min\{k \mid U_k \in [0, \rho]\}$ .

For the special case where  $\lambda_0 = 0$  (and thus  $\lambda_1 = 1$ ) this is a Poisson process. It is then equivalent to generating i.i.d. uniform random numbers  $U_1, U_2, \dots \in (0, 1)$  until the first time  $U_1 \cdot U_2 \cdot \dots \cdot U_T < \rho$  and return  $U = U_1 \cdot U_2 \cdot \dots \cdot U_T$ . The stopping time  $T-1$  of this sampling procedure is Poisson distributed with parameter  $-\log(\rho)$ . Thus we find for the expectation of  $T$ ,  $\mathbf{E}[T] = 1 - \log(\rho)$ , see e.g. [Devroye 1986, §X.3.3].

For the general case with  $\lambda_0 < 0 < \rho < \lambda_1$  the convergence is a little bit slower.

**THEOREM 4.1.** *The stopping time  $T$  of the above sampling procedure has expectation*

$$\mathbf{E}[T] \leq 1 - \log(\rho) \frac{e}{1 - \log 2}. \quad (8)$$

**PROOF.** In each step the interval, on which  $U_k$  is uniformly distributed, is shrunk by a factor  $S_k$ . The distribution of  $S_k$  depends on  $\rho$ ,  $m_1 = |\lambda_0^k|$  and  $m_2 = \lambda_1^k - \rho$ . For fixed  $\rho$ ,  $m_1 = m_2$  is the worst case in the sense that the shrinkage  $S_k$  is stochastically greater than for  $m_1 \neq m_2$ . Thus we consider the following modification which gives an even worse shrink factor. In each step the length of the interval is shrunk by a factor  $S_k^*$  which is uniform on  $[0, \rho]$  with probability  $\rho$ , and uniform on  $[(1+\rho)/2, 1]$  with probability  $1 - \rho$ . Denote the corresponding stopping time, the first instance when  $U_k \in [0, \rho]$  is reached, by  $T^*$ . Notice that  $\mathbf{E}[T] \leq \mathbf{E}[T^*]$ .

We have to show that (8) holds for the expectation of  $T^*$  which is given by

$$\mathbf{E}[T^*] = 1 + \sum_{t=1}^{\infty} \pi_t$$



with

$$\pi_t = \mathbf{P}(U_t S_1^* \cdots S_{t-1}^* > \rho).$$

Let  $s = -\log(\rho) > 0$ .  $U_t S_1^* \cdots S_{t-1}^* > \rho$  implies  $S_i^* > \rho$  for  $i = 1, \dots, t-1$ . Since  $S_t^* | \{S_i^* > \rho\} \sim \mathbb{U}[(\rho+1)/2, 1]$  Markov's inequality, see Billingsley [1986], implies

$$\begin{aligned} \pi_t &= (1-\rho)^{t-1} \mathbf{P}(U_t S_1^* \cdots S_{t-1}^* > \rho | S_1^* > \rho, \dots, S_{t-1}^* > \rho) \\ &\leq \frac{(1-\rho)^{t-1}}{\rho^s} \mathbf{E}[(U_t S_1^* \cdots S_{t-1}^*)^s | S_1^* > \rho, \dots, S_{t-1}^* > \rho] \end{aligned}$$

Now  $\mathbf{E}[U_t^s] = 1/(1+s)$  and

$$\mathbf{E}[S_i^{*s} | S_i^* > \rho] = \frac{2}{1-\rho} \int_{(1+\rho)/2}^1 x^s dx = \frac{2}{1-\rho} \frac{1}{1+s} \left(1 - \left(\frac{1+\rho}{2}\right)^{s+1}\right)$$

together with the independence of  $U_t$  and the shrink factors  $S_i^*$  gives

$$\pi_t \leq \frac{1}{\rho^s} \frac{2^{t-1}}{(1+s)^t} \left(1 - \left(\frac{1+\rho}{2}\right)^{s+1}\right)^{t-1}.$$

Hence by the summation formula of geometric series we have

$$\begin{aligned} \mathbf{E}[T^*] &\leq 1 + \frac{1}{\rho^s} \cdot \frac{1}{1+s} \cdot \frac{1}{1 - 2(1 - ((1+\rho)/2)^{s+1})/(1+s)} \\ &= 1 + \frac{1}{\rho^s} \cdot \frac{1}{s - 1 + 2((1+\rho)/2)^{s+1}}. \end{aligned}$$

Using a convexity argument we get

$$2 \left(\frac{1+\rho}{2}\right)^{s+1} \geq \left(\frac{1+\rho}{2}\right)^s \geq 1 + s \log((1+\rho)/2)$$

and thus we find

$$\mathbf{E}[T^*] \leq 1 + \frac{1}{\rho^s s} \cdot \frac{1}{1 + \log((1+\rho)/2)} = 1 + \log(1/\rho) \frac{e}{1 + \log((1+\rho)/2)},$$

which is bounded by (8).  $\square$

### 4.3 Direction Sampling

Hypersphere sampling, i.e. choosing directions  $\mathbf{d}_k$  uniformly distributed on the sphere, seems to be a good choice as it is simple and easy to implement (see e.g. Hörmann et al. [2004, §11.2.1]). The improved method by Kaufman and Smith [1998] is another possible algorithm.

### 4.4 Bounded "Plate"

Computational experience shows that computing the bounding rectangle  $\mathcal{R}(f)$  numerically is the most time consuming step in higher dimensions. A possible solution is to replace the bounding rectangle  $\mathcal{R}(f)$  by the unbounded "plate" given by  $\{(\mathbf{u}, v): 0 < v < \sqrt[n+1]{f(\mathbf{m})}\}$ . Notice that the line  $L_k$  becomes infinitely long when and only when the  $v$ -coordinate of the random direction is 0. Thus for a direction

drawn uniformly from the hypersphere (or any other absolutely continuous distribution)  $L_k$  has finite length almost surely. However, it can become very long and thus we have to use the adaptive sampling discussed above. Otherwise the number of iterations becomes prohibitively large.

To get a first impression about the expected number of iterations  $\mathbf{E}(I)$  when we use the unbounded “plate” we assume that  $r = 1$ ,  $f(\mathbf{m}) = 1$ , and  $\mathcal{A}(f)$  is a ball of radius 1. This is the case when we look at the multivariate Cauchy distribution with density proportional to  $f(\mathbf{x}) = (2 + \|\mathbf{x}\|^2)^{-(n+1)}$ . The length of a line segment  $L_k$  is then given by  $2\sqrt{1 + \tan(\theta)^2}$  where  $\theta$  is the angle between the direction  $\mathbf{d}$  and the  $v$ -axis. For a point  $\mathbf{X}_k$  and a direction  $\mathbf{d}$  with angle  $\theta$  the expected number of iterations is given by  $1 + \log(2\sqrt{1 + \tan(\theta)^2}/\ell) e/(1 + \log 2)$  by Thm. 4.1 where  $\ell$  denotes the length of  $\Lambda_k$ . When the direction  $\mathbf{d}$  is uniformly distributed on the hypersphere,  $\theta$  follows a distribution with density proportional to the area of the  $(n-1)$ -sphere with radius  $\sin(\theta)$ , i.e. with density  $\sin(\theta)^{n-1} S_{n-1}/S_n$  for  $0 \leq \theta \leq \pi$  where  $S_{n-1} = 2\pi^{n/2}/\Gamma(n/2)$  denotes the area of the surface of the unit  $(n-1)$ -sphere (i.e. in  $\mathbb{R}^n$ ) and  $\Gamma(\cdot)$  denotes the gamma function. Now assume that  $\mathbf{X}_k$  is uniformly distributed in the ball  $\mathcal{A}(f)$ . Then for a given  $\theta$  the length  $\ell$  of  $\Lambda_k$  depends on the distance  $\delta$  of  $\mathbf{X}_k$  from the straight line spanned by  $\mathbf{d}$  through the center of the ball, i.e.,  $\ell = 2\sqrt{1 - \delta^2}$ . The density of the distribution of  $\delta$  for uniformly distributed  $\mathbf{X}_k$  is proportional the area of the  $n$ -dimensional cylindrical surface given by all points of distance  $\delta$ , i.e., the density is given by  $2\delta^{n-1}\sqrt{1 - \delta^2} (n+1) S_{n-1}/S_n$  for  $0 \leq \delta \leq 1$ . Consequently we obtain for the expected number of iterations,

$$\begin{aligned} \mathbf{E}(I) &\leq \int_0^\pi \int_0^1 \left( 1 + \log \left( 2\sqrt{1 + \tan(\theta)^2} / 2\sqrt{1 - \delta^2} \right) \right) e/(1 + \log 2) \cdot \\ &\quad 2\delta^{n-1}\sqrt{1 - \delta^2} (n+1) \frac{S_{n-1}}{S_n} \sin(\theta)^{n-1} \frac{S_{n-1}}{S_n} d\delta d\theta \\ &= \frac{e}{1 + \log(2)} \left( 1 + \int_0^\pi \frac{1}{2} \log(1 + \tan(\theta)^2) \sin(\theta)^{n-1} \frac{S_{n-1}}{S_n} d\theta \right. \\ &\quad \left. - \int_0^1 \frac{1}{2} \log(1 - \delta^2) 2\delta^{n-1}\sqrt{1 - \delta^2} (n+1) \frac{S_{n-1}}{S_n} d\delta \right) \end{aligned}$$

Using the inequality  $\sin(\theta)^{n-1} \leq 1$  ( $0 \leq \theta \leq \pi$ ) for the first integrand and  $0 \leq -x \log(x) \leq 1/e$  ( $0 < x \leq 1$ ) for the second integrand we get by a straightforward computation

$$\mathbf{E}(I) \leq \frac{e}{1 + \log(2)} \left( 1 + \log(2)\pi \frac{S_{n-1}}{S_n} + \frac{2}{e} \frac{n+1}{n} \frac{S_{n-1}}{S_n} \right)$$

i.e., the expected number of iterations is finite for every dimension  $n$ . Stirling’s formula gives  $\Gamma(\frac{n+1}{2})/\Gamma(\frac{n}{2}) = n^{1/2} \pi^{1/2} + o(n^{1/2})$  and  $S_{n-1}/S_n = \sqrt{n} + o(\sqrt{n})$ . Hence

$$\mathbf{E}(I) \leq 4.7 \sqrt{n} + o(\sqrt{n}).$$

#### 4.5 Floating Point Arithmetic

When stating the theoretical background of our method we have assumed that we have real numbers,  $\mathbb{R}$ . However, the algorithms are designed to be implemented

in real world computers which work with floating point numbers that have only a limited precision; see Overton [2001] for an introduction to floating point arithmetic. Thus expressions like  ${}^{n+1}\sqrt{f}$  or  $v^{n+1}$  may result in a reduction of significant digits, overflow, or underflow when the number  $n$  of dimensions is large. As a consequence the generated distribution deviates too much from the target distribution or the algorithm even may fail too work. Therefore we need two strategies to reduce these problems:

- (1) The density  $f$  is rescaled such that  $f(\mathbf{m}) = 1$  for the mode  $\mathbf{m}$ .
- (2) The oracle should return  $\log(f(\mathbf{x}))$ .

The latter point is quite convenient as the logarithms of many densities (or of multiples of densities) are often easier to compute.

#### 4.6 HITRO

We have compiled two algorithms<sup>1</sup> (HITRO-box and HITRO-plate). The first one uses a bounding hyper-rectangle for the rejection steps, the second one uses the unbounded “plate”.

There are a few remarks concerning these Algorithms:

—The set  $\mathcal{A}(f)$  should be convex. This is, e.g., the case for  $r = 1$  when the density  $f$  is log-concave. Otherwise the conditions for the convergence theorems in [Smith 1984] are not satisfied, i.e., whenever  $\Lambda_k$  is disconnected and Step 11 in Algorithm HITRO-box is used then

1. points sampled from  $\Lambda_k$  are not uniformly distributed, and
2. the Markov chain is not time reversible, i.e., the transition probability function is not symmetric.

Thus when  $\mathcal{A}(f)$  is not convex, then the adaptive Step 11 in Algorithm HITRO-box should be skipped. (Algorithm HITRO-plate becomes very slow without Step 11.) Nevertheless, in our computational experiments it seemed that the Markov chain still converges to the target distribution with adaptive uniform sampling even if  $\mathcal{A}(f)$  is not convex.

—We can replace the mode  $\mathbf{m}$  by any other point  $\mathbf{c}$ . This is in particular useful when  $f$  is multimodal. Then  $\mathbf{c}$  could represent the “center” of the distribution. Notice, however, that  $f_m$  should be set to a value close to  $\max f(\mathbf{x})$  (for computational reasons) in Step 1. Furthermore, the starting point  $(\mathbf{U}_0, V_0)$  must be set accordingly.

— $\mathcal{R}(f)$  need not necessarily be the minimal bounding rectangle as defined in (4). It can be larger.

#### 4.7 Convergence

Because the ratio-of-uniforms transformation has constant Jacobian we can immediately derive convergence results for a Markov chain  $\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2, \dots$  generated by a HITRO algorithm.

<sup>1</sup>*Hitro* is the Slovenian word for *fast*.

---

**Algorithm 1** HITRO-box

---

**Input:** Density function  $f$  in  $\mathbb{R}^n$ , mode  $\mathbf{m}$ ; parameter  $r$ , sample size  $N$ .**Output:** Sequence  $\mathbf{X}_k$  of random vectors with asymptotic distribution  $f$ .

[ Setup ]

- 1: Compute  $f_m \leftarrow f(\mathbf{m})$  and bounding rectangle  $\mathcal{R}_{r,\mathbf{m}}(f/f_m) = (\mathbf{R}_l, \mathbf{R}_u)$ .
- 2: Set  $(\mathbf{U}_0, V_0) \leftarrow (\mathbf{0}, \frac{1}{2})$  and  $k \leftarrow 0$ .

[ Generate chain ]

- 3: **repeat**
  - 4:   Generate a random direction  $\mathbf{d}_k$  uniformly on  $(n+1)$ -sphere.  
    [ Generate a point uniformly in  $\Lambda_k$  ]
  - 5:   Compute  $L_k = (\lambda_{k0}, \lambda_{k1}) = \{\lambda: \mathbf{x}_k + \lambda \mathbf{d}_k \in (\mathbf{R}_l, \mathbf{R}_u)\}$ .
  - 6:   **loop**
  - 7:     Generate  $\lambda_k$  uniformly distributed in  $(\lambda_{k0}, \lambda_{k1})$ .
  - 8:     Set  $(\mathbf{U}_{k+1}, V_{k+1}) \leftarrow (\mathbf{U}_k, V_k) + \lambda_k \mathbf{d}_k$ .
  - 9:     Set  $\mathbf{X}_{k+1} \leftarrow \mathbf{U}_{k+1}/(V_{k+1})^r + \mathbf{m}$ .
  - 10:    **if**  $(rn+1) \log(V_{k+1}) \geq \log(f(\mathbf{X}_{k+1})/f_m)$  **then**
  - 11:     Set  $\lambda_{k0} \leftarrow \lambda_k$  (if  $\lambda_k < 0$ ) or  $\lambda_{k1} \leftarrow \lambda_k$  (otherwise).   [ Shrink  $L_k$  ]
  - 12:    **else**
  - 13:     Stop loop.
  - [ Append to chain ]
  - 14:   Set  $k \leftarrow k+1$ .
  - 15: **until**  $k = N$ .
- 

---

**Algorithm 2** HITRO-plate

---

**Input:** Density function  $f$  in  $\mathbb{R}^n$ , mode  $\mathbf{m}$ ; parameter  $r$ , sample size  $N$ .**Output:** Sequence  $\mathbf{X}_k$  of random vectors with asymptotic distribution  $f$ .

[ Setup ]

- 1: Compute  $f_m \leftarrow f(\mathbf{m})$ .
- 2: Set  $(\mathbf{U}_0, V_0) \leftarrow (\mathbf{0}, \frac{1}{2})$  and  $k \leftarrow 0$ .

[ Generate chain ]

- 3: **repeat**
  - 4:   Generate a random direction  $\mathbf{d}_k = (\mathbf{d}_u, d_v)$  uniformly on  $(n+1)$ -sphere.
  - 5:   Set  $\lambda_{k0} \leftarrow -|v/d_v|$  and  $\lambda_{k1} \leftarrow |(1-v)/d_v|$ .
  - 6:   **loop**
  - 7:     Generate  $\lambda_k$  uniformly distributed in  $(\lambda_{k0}, \lambda_{k1})$ .
  - 8:     Set  $(\mathbf{U}_{k+1}, V_{k+1}) \leftarrow (\mathbf{U}_k, V_k) + \lambda_k \mathbf{d}_k$ .
  - 9:     Set  $\mathbf{X}_{k+1} \leftarrow \mathbf{U}_{k+1}/(V_{k+1})^r + \mathbf{m}$ .
  - 10:    **if**  $(rn+1) \log(V_{k+1}) \geq \log(f(\mathbf{X}_{k+1})/f_m)$  **then**
  - 11:     Set  $\lambda_{k0} \leftarrow \lambda_k$  (if  $\lambda_k < 0$ ) or  $\lambda_{k1} \leftarrow \lambda_k$  (otherwise).   [ Shrink  $L_k$  ]
  - 12:    **else**
  - 13:     Stop loop.
  - 14:   Set  $k \leftarrow k+1$ .
  - 15: **until**  $k = N$ .
-

If  $f$  is log-concave (and thus  $\mathcal{A}(f)$  is convex) then for any measurable set  $A \subseteq \mathbb{R}^n$  there exist constants  $M < \infty$  and  $\kappa < 1$  such that for any  $m$

$$|\mathbb{P}(\mathbf{X}_m \in A | \mathbf{X}_0 = \mathbf{x}) - \mu(A)| < M\kappa^m$$

where  $\mu(\cdot)$  denotes the  $n$ -dimensional content of  $A$ . That is, the HITRO algorithms are uniformly ergodic [Bélisle et al. 1998].

This result also holds for the HITRO-box algorithm without adaptive uniform sampling (Step 11) whenever  $\|\mathbf{x}\|^{n+1/r} f(\mathbf{x})$  (and thus  $\mathcal{A}(f)$ ) is bounded [Bélisle et al. 1998].

If  $f$  is log-concave we also can give bounds on the mixing times. However, we need more information on the shape of  $\mathcal{A}(f)$ . Let  $\gamma$  denote the ratio of the diameter of  $\mathcal{A}(f)$  to the radius of the largest ball contained in  $\mathcal{A}(f)$ . Let  $0 < \varepsilon < 1$  and

$$N = \left\lceil 4 \cdot 10^8 \frac{n^2 \gamma^2}{\varepsilon^2} \log(2/\varepsilon) \right\rceil.$$

Then by a bound of Lovász [1999] for every  $A \in \mathbb{R}^n$ ,

$$|\mathbb{P}(\mathbf{X}_N \in A) - \mu(A)| \leq \varepsilon.$$

#### 4.8 Other Variants

There are several obvious variants of the two presented HITRO algorithms.

- *Gibbs Sampling (Step 4)*: The random directions  $\mathbf{d}_k$  can be replaced by coordinate directions that are chosen in a given order. This can be seen as “traditional” Gibbs sampling from the density proportional to the indicator function of the set  $\mathcal{A}(f)$  in  $\mathbb{R}^{n+1}$ . Our computational experiments show that this variant often results in a better performance compared to random directions (when the correlations between the components of the random vector are not very high). Notice, however, that this requires the computation of a bounding rectangle  $\mathcal{R}(f)$ .
- *Slice Sampler (Steps 1 and 5)*: The respective computation of the bounding plate and the bounding rectangle can be replaced by a procedure similar to the first part of the slice sampler proposed by Neal [2003]. For a chosen (random or coordinate) direction  $\mathbf{d}_k$  start from some interval  $(\lambda_{k0}, \lambda_{k1})$  and test whether the corresponding end points of line segment are inside  $\mathcal{A}(f)$ . If not then enlarge the interval by some appropriate factor and test again. Otherwise start (adaptive) uniform sampling from the line segment. Notice, that this procedure only works reliably if the region  $\mathcal{A}(f)$  is convex (hypersphere sampling) or unimodal (coordinate directions), or at least some information about the shape of  $\mathcal{A}(f)$  is available (see also the discussion on multimodal densities in [Neal 2003]). It is useful to use some rectangle or plate to store the information about  $\mathcal{A}(f)$ . Each time when the starting interval  $(\lambda_{k0}, \lambda_{k1})$  has to be enlarged, the rectangle is enlarged analogously. Notice that the price for this procedure (compared to a precomputed bounded rectangle/plate) is that we need two additional evaluations of the density to test whether the starting interval covers  $\Lambda_k$ .
- *Stochastic Optimization (Step 1)*: The computation of the mode (and of the entire bounding rectangle) can be seen as equivalent to the burn-in phase of other Markov chain samplers. Indeed, one could use the algorithm to find the

mode of the density  $f$  by starting with some guess for the upper bound of  $f$ , generate a chain and adapt this guess whenever a larger value for  $f$  was found (see Zabinsky [1998] for a survey on stochastic optimization).

Of course it is also possible to combine the Ratio-of-Uniforms method with other Markov chain samplers, in particular we have implemented the Metropolis Random Walk sampler (with a multinormal proposal density) and the ball sampler (i.e., with a uniform distribution on a ball), see e.g. [Robert and Casella 2004].

*Remark.* When we finished this paper we became aware of a recent talk by Tierney [2005] who also pointed out the usefulness of the Ratio-of-Uniforms method for Markov chain Monte Carlo. In his talk he also discusses some other aspects of using the Ratio-of-Uniforms transformation in the framework of MCMC.

## 5. COMPUTATIONAL EXPERIENCES

We have implemented the proposed algorithms in our library UNU.RAN (Universal Non-Uniform RANdom number generators, [Leydold et al. 2005]) and ran many experiments with multinormal, multi- $t$ , and multi-Cauchy distributions. We used many dimensions and different variance-covariance matrices to test the performance of the proposed algorithms. Figure 2 shows a typical example for the number of calls to the (logarithm of the) density function (oracle). It shows that the performance gain by adaptive uniform sampling from line segments is very large in higher dimensions. Moreover, the ratio between the average number of calls needed when using an unbounded plate instead of a bounding box is less than 2 and decreases with increasing dimension. The expected number of oracle-calls stays well below 7 even for dimensions as high as 100. For “conventional” rejection from bounding rectangle the expected number of oracle calls would be  $5.06 \times 10^{70}$ .

We compared our results with those of the Gibbs sampler as its “obvious” competitor. As we are mainly interested in black-box methods we used transformed density rejection [Gilks and Wild 1992] (which can be applied for all log-concave p.d.f.s) for drawing points from the full conditional distributions. Notice, however, that these results contains both calls to the logarithm of the log-density function and calls to its derivative. Figure 2 shows that the proposed algorithm requires fewer calls to the oracle for dimensions up to 100.

In a second experiment we have measured the marginal generation times for our algorithms and compared it to our implementation of the Gibbs sampler. The results are shown in Figure 3. It shows that the proposed algorithm are much faster in smaller dimensions. However, these numbers deserve some explanations. We have implemented all our algorithms such that only respective function pointers to the (log-) density of the joint distribution and its gradient (or all its first partial derivatives) have to be provided by the user. Moreover, we have tried to make the library robust against numerical under-/overflow. The evaluation of the log-density of the multinormal distribution requires the computation of some quadratic form and UNU.RAN only uses a naïve implementation with computational complexity  $O(n^2)$  (which is sufficient for a small number of dimensions). Thus Figure 3 can be interpreted in the following way:

—The marginal generation time consists of the time for the actual algorithm and

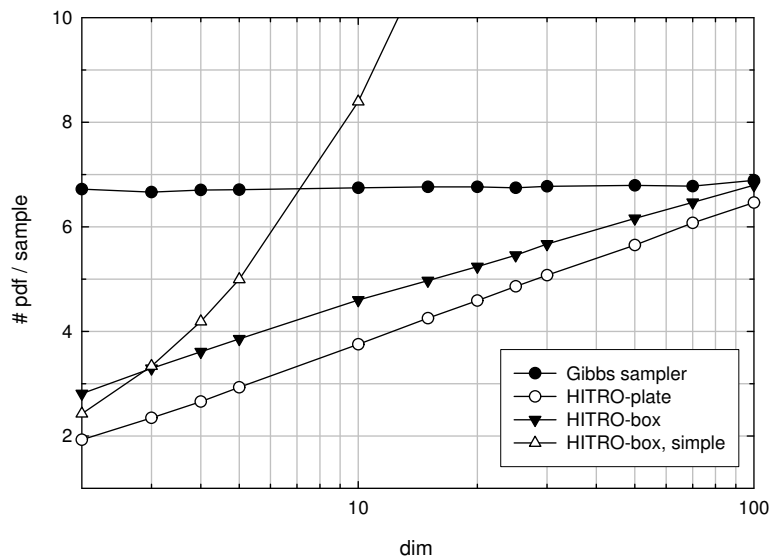


Fig. 2. Average number of p.d.f. calls for the multinormal distribution with variance-covariance matrix  $\Sigma$  where  $\Sigma_{ik} = 0.9^{|i-k|}$ , i.e., the correlation matrix of an AR(1) process with  $\rho = 0.9$ . (“HITRO-box, simple” uses simple rejection for sampling uniformly from the line segment instead of adaptive sampling.)

for the time to evaluate the given log-density.

- The time complexity of the actual algorithm is constant (Gibbs sampler, HITRO-plate) or increases slowly (linear) in dimension. It is much smaller for the HITRO algorithms compared to the Gibbs sampler.
- For higher dimensions computing the log-density is the most expensive part. Thus the number of oracle calls becomes important (Fig. 2). However, about half of the calls for the Gibbs sampler are calls to the partial derivatives of the log-density which are much cheaper than calls to the log-density itself. Hence the Gibbs sampler outperforms the HITRO method in higher dimensions in the case of the multinormal distribution. The situation changes when computing the derivatives are not cheaper than the log-density itself, e.g. for multi- $t$  distributions.

We also made experiments to check the quality of the generated sequence of random points. It shows that the HITRO algorithms have a similar performance as the Gibbs sampler. Running  $\chi^2$  goodness-of-fit tests HITRO showed slightly better statistical properties. We have estimated mean square errors in Monte Carlo computations of moments of marginal distributions (an important task in Bayesian inference). Here the results for the Gibbs sampler are better as long as the correlations  $\rho$  are not too high. However, when we used the Gibbs sampling version of the HITRO-box algorithm (instead of random direction; see §4.8) the performance is similar and depends from the particular distribution, see Fig. 4 for an example.

We also compared our results to Random Walk Metropolis samplers with multi-

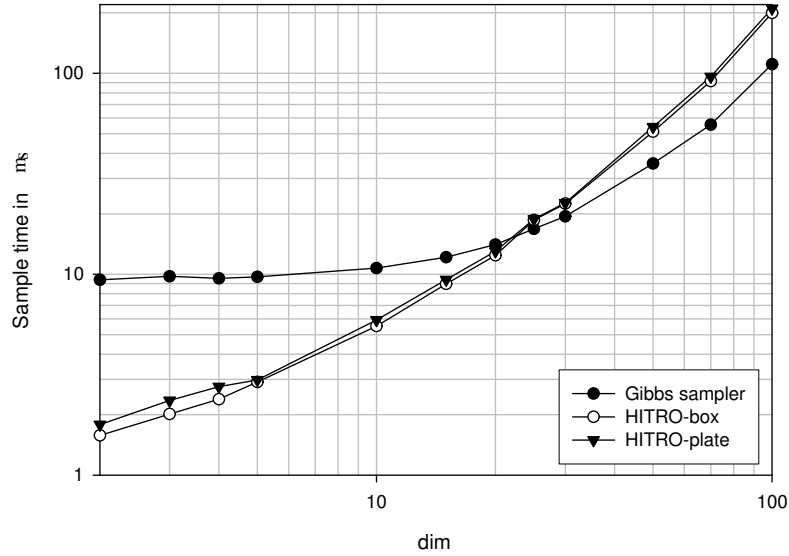


Fig. 3. Marginal generation times for the multinormal distribution with variance-covariance matrix  $\Sigma$  where  $\Sigma_{ik} = 0.9^{|i-k|}$ , i.e., the correlation matrix of an AR(1) process with  $\rho = 0.9$ .

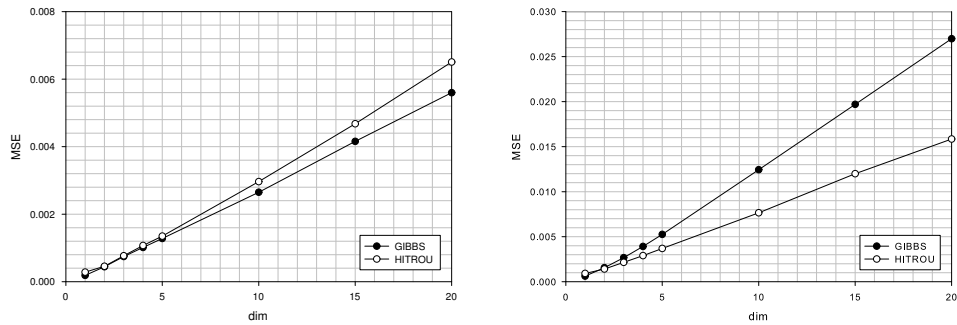


Fig. 4. Mean square error for Monte Carlo estimation of variance of multinormal (l.h.s.) and multi- $t$  distribution with 8 degrees of freedom (r.h.s.). Gibbs sampling using transformed density rejection and HITRO-box using coordinate direction sampling; sample size =  $10^4$ .

normal proposal distributions and those that are uniform on balls (sometimes called *ball sampler*). Here the performance in Monte Carlo computations for our test distributions is much worse.

The region  $\mathcal{A}(f)$  should be convex when adaptive uniform sampling is used. Otherwise, the ergodicity of the proposed Hit-and-Run sampler has not been shown, yet. Nevertheless, we ran experiments on normal mixtures. Figure 5 shows graphs, regions  $\mathcal{A}(f)$ , and chains of length 1000 produced by our algorithm with adaptive



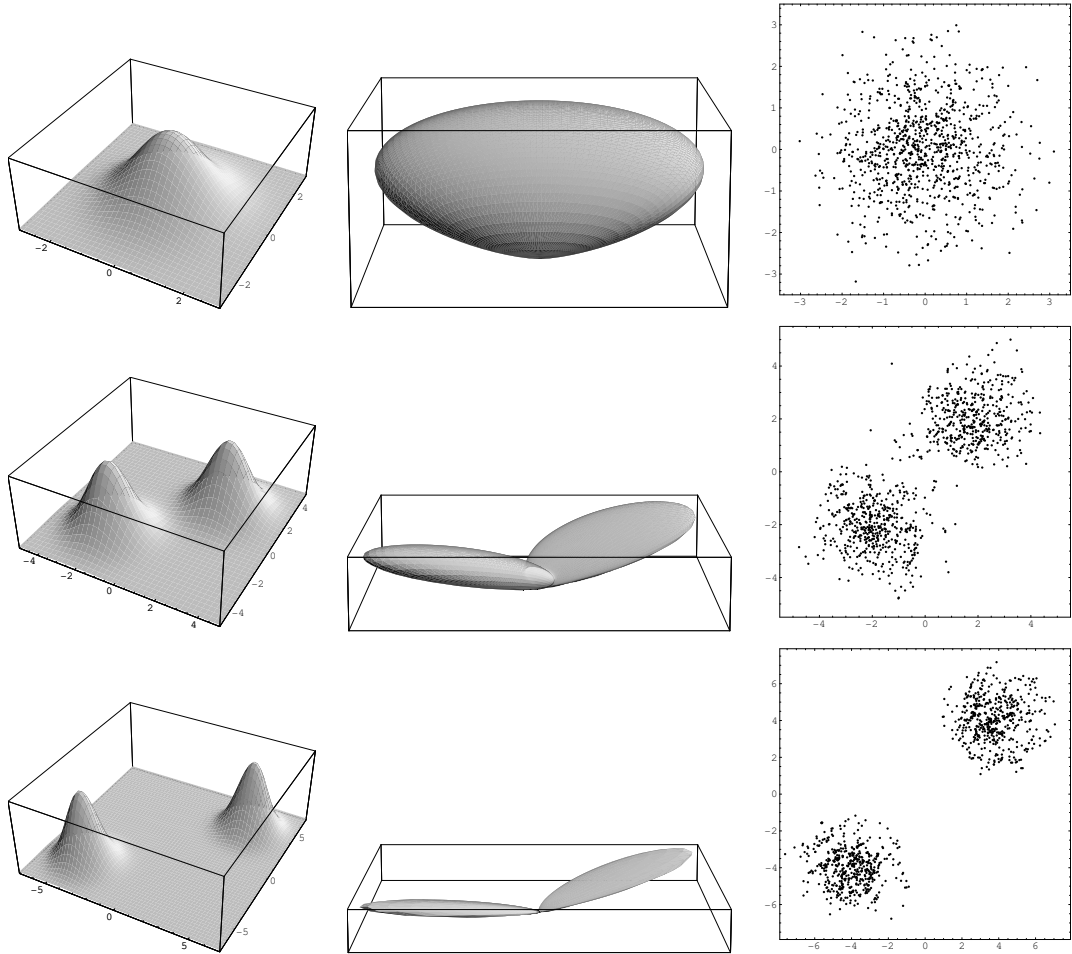


Fig. 5. Graph of mixture density function (9), the surfaces of the RoU-shapes and  $N = 1000$  random sample points for  $\mu = 0$ ,  $\mu = 2$  and  $\mu = 4$  respectively.

uniform sampling for bivariate distributions with densities

$$f(x, y) = \frac{1}{4\pi} (e^{-((x-\mu)^2+(y-\mu)^2)/2} + e^{-((x+\mu)^2+(y+\mu)^2)/2}). \quad (9)$$

Notice that  $\mathcal{A}(f)$  is not convex in all cases. Moreover, for large values of  $\mu$  neither the Gibbs sampler nor a random walk Metropolis sampler would work efficiently. Nevertheless, the Hit-and-Run algorithm seems to mix fast even for this difficult distribution.

## 6. CONCLUSION

We have proposed algorithms based on the Hit-and-Run Sampler and the Ratio-of-Uniforms. These HITRO algorithms are very simple, easy to implement, relatively fast, and work for many distributions out of the box. Compared to the Gibbs

sampler they have similar performance in Monte Carlo integration but there is no necessity for a special generator for sampling from the full conditional distribution. Compared to a black box version of the Gibbs sampler (which uses transformed density rejection) the HITRO algorithms do not need the derivative of the log-density, are much simpler to implement and up to dimension 100 they require fewer evaluations of the density function.

Adaptive uniform sampling with an unbounded plate (Algorithm HITRO-plate) is the best suited practical algorithm among our proposed methods. The computation of the mode can be seen as equivalent to the burn-in phase of other Markov chain samplers.

The slice sampler [Neal 2003] has been proposed to sample points uniformly in the region below the graph of  $f$ ,  $\mathcal{G}(f)$ . There points are sampled uniformly in slices  $\{\mathbf{x}: f(\mathbf{x}) = y\}$ . However, this requires a search algorithm to get a cover for such a slice each time. The sampler proposed by Chen and Schmeiser [1998] also generates a Markov chain with uniform stationary distribution in  $\mathcal{G}(f)$ . However, it requires to fix two parameters which are crucial for the performance of the algorithms.

An important feature of the new algorithm is its simplicity. No proposal distribution has to be adjusted for the target distribution. (The influence of the parameter  $r$  is rather small). Of course rescaling can improve the convergence of the sampler.

#### ACKNOWLEDGMENTS

The authors are grateful for helpful conversations with Wolfgang Hörmann and Günter Tirlir. We thank Sylvia Früwirth-Schnatter for calling our attention to the talk of Luke Tierney. This work was supported by the Austrian *Fonds zur Förderung der Wissenschaftlichen Forschung*, Proj.No. P16767-N12.

#### REFERENCES

- BÉLISLE, C. 2000. Slow hit-and-run sampling. *Stat. Probab. Lett.* 47, 1, 33–43.
- BÉLISLE, C., BONEH, A., AND CARON, R. J. 1998. Convergence properties of hit-and-run samplers. *Commun. Stat., Stochastic Models* 14, 4, 767–800.
- BELISLE, C. J. P., ROMELJN, H. E., AND SMITH, R. L. 1993. Hit-and-run algorithms for generating multivariate distributions. *Mathematics of Operations Research* 18, 255–266.
- BERBEE, H. C. P., BOENDER, C. G. E., RINNOOY KAN, A., SCHEFFER, C. L., SMITH, R. L., AND TELGEN, J. 1987. Hit-and-run algorithms for the identification of nonredundant linear inequalities. *Math. Program.* 37, 184–207.
- BILLINGSLEY, P. 1986. *Probability and Measure*. Wiley & Sons, New York.
- BONEH, A. AND GOLAN, A. 1979. Constraints' redundancy and feasible region boundedness by random feasible point generator (rfpg). In *Third European Congress on Operations Research, EURO III*. Amsterdam.
- CHEN, M.-H. AND SCHMEISER, B. 1993. Performance of the Gibbs, Hit-and-run, and Metropolis samplers. *Journal of Computational and Graphical Statistics* 2, 251–272.
- CHEN, M.-H. AND SCHMEISER, B. 1998. Toward black-box sampling: a random-direction interior-point Markov chain approach. *Journal of Computational and Graphical Statistics* 7, 1–22.
- CHEN, M.-H. AND SCHMEISER, B. W. 1996. General hit-and-run Monte Carlo sampling for evaluating multidimensional integrals. *Oper. Res. Lett.* 19, 4, 161–169.
- DEVROYE, L. 1986. *Non-Uniform Random Variate Generation*. Springer-Verlag, New-York.
- GILKS, W. R. AND WILD, P. 1992. Adaptive rejection sampling for Gibbs sampling. *Applied Statistics* 41, 2, 337–348.

- HÖRMANN, W. 1995. A rejection technique for sampling from T-concave distributions. *ACM Trans. Math. Software* 21, 2, 182–193.
- HÖRMANN, W., LEYDOLD, J., AND DERFLINGER, G. 2004. *Automatic Nonuniform Random Variate Generation*. Springer-Verlag, Berlin Heidelberg.
- KAUFMAN, D. E. AND SMITH, R. L. 1998. Direction choice for accelerated convergence in hit-and-run sampling. *Oper. Res.* 46, 1, 84–95.
- KINDERMAN, A. J. AND MONAHAN, J. F. 1977. Computer generation of random variables using the ratio of uniform deviates. *ACM Trans. Math. Software* 3, 3, 257–260.
- LEYDOLD, J. 2000. Automatic sampling with the ratio-of-uniforms method. *ACM Trans. Math. Software* 26, 1, 78–98.
- LEYDOLD, J., HÖRMANN, W., JANKA, E., KARAWATZKI, R., AND TIRLER, G. 2005. *UNU.RAN – A Library for Non-Uniform Universal Random Variate Generation*. Department of Statistics and Mathematics, WU Wien, A-1090 Wien, Austria. available at <http://statistik.wu-wien.ac.at/unuran/>.
- LOVÁSZ, L. 1999. Hit-and-run mixes fast. *Math. Program.* 86, 3, 443–461.
- LOVÁSZ, L. AND VEMPALA, S. 2003. Hit-and-run is fast and fun. Tech. Rep. MSR-TR-2003-05, Microsoft Research. Available at <http://www-math.mit.edu/~vempala/papers/logcon-hitrun.ps>.
- LOVÁSZ, L. AND VEMPALA, S. 2004. Hit-and-run from a corner. In *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*. ACM Press, Chicago, IL, USA, 310–314.
- NEAL, R. M. 2003. Slice sampling. *Ann. Stat.* 31, 3, 705–767. (With discussions and rejoinder).
- OVERTON, M. L. 2001. *Numerical Computing with IEEE Floating Point Arithmetic*. SIAM, Philadelphia.
- ROBERT, C. AND CASELLA, G. 2004. *Monte Carlo Statistical Methods*, 2nd ed. Springer Texts in Statistics. Springer, New York.
- SMITH, R. L. 1980. Monte Carlo procedures for generating random feasible solutions to mathematical programming. In *ORSA/TIMS Conference*. Whashington, D.C.
- SMITH, R. L. 1984. Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research* 32, 1296–1308.
- SMITH, R. L. 1996. The hit-and-run sampler: a globally reaching Markov chain sampler for generating arbitrary multivariate distributions. In *WSC '96: Proceedings of the 28th conference on Winter simulation*, J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain, Eds. ACM Press, 260–264.
- ȘTEFĂNESCU, S. AND VĂDUVA, I. 1987. On computer generation of random vectors by transformations of uniformly distributed vectors. *Computing* 39, 141–153.
- TIERNEY, L. 2005. Markov chain Monte Carlo using the ratio-of-uniforms transformation. Talk presented at the 2nd IMS-ISBA Joint Meeting MCMCSki, Bormio, January 12–14.
- VADUVA, I. 1984. Computer generation of random vectors based on transformation of uniformly distributed vectors. In *Probability theory, Proc. 7th Conf., Brasov/Rom. 1982*. 589–598.
- WAKEFIELD, J. C., GELFAND, A. E., AND SMITH, A. F. M. 1991. Efficient generation of random variates via the ratio-of-uniforms method. *Statist. Comput.* 1, 2, 129–133.
- ZABINSKY, Z. B. 1998. Stochastic methods for practical global optimization. *Journal of Global Optimization* 13, 433–444.