

# Exact Hamiltonian Monte Carlo for Truncated Multivariate Gaussians

Ari Pakman\* and Liam Paninski†

Department of Statistics,  
Center for Theoretical Neuroscience  
and Grossman Center for the Statistics of Mind

Columbia University

## Abstract

We present a Hamiltonian Monte Carlo algorithm to sample from multivariate Gaussian distributions in which the target space is constrained by linear and quadratic inequalities or products thereof. The Hamiltonian equations of motion can be integrated exactly and there are no parameters to tune. The algorithm mixes faster and is more efficient than Gibbs sampling. The runtime depends on the number and shape of the constraints but the algorithm is highly parallelizable. In many cases, we can exploit special structure in the covariance matrices of the untruncated Gaussian to further speed up the runtime. A simple extension of the algorithm permits sampling from distributions whose log-density is piecewise quadratic, as in the “Bayesian Lasso” model.

**Keywords:** Markov Chain Monte Carlo, Hamiltonian Monte Carlo, Truncated Multivariate Gaussians, Bayesian Modeling.

---

\*ari@stat.columbia.edu

†liam@stat.columbia.edu

# 1 Introduction

The advent of Markov Chain Monte Carlo methods has made it possible to sample from complex multivariate probability distributions [Robert and Casella, 2004], leading to a remarkable progress in Bayesian modeling, with applications to many areas of applied statistics and machine learning [Gelman et al., 2004].

In many cases, the data or the parameter space are constrained [Gelfand et al., 1992] and the need arises for efficient sampling techniques for truncated distributions. In this paper we will focus on the Truncated Multivariate Gaussian (TMG), a  $d$ -dimensional multivariate Gaussian distribution of the form

$$\log p(\mathbf{X}) = -\frac{1}{2}\mathbf{X}^T\mathbf{M}\mathbf{X} + \mathbf{r}^T\mathbf{X} + \text{const.} \quad (1.1)$$

with  $\mathbf{X}, \mathbf{r} \in \mathbb{R}^d$  and  $\mathbf{M}$  positive definite, subject to  $m$  inequalities

$$Q_j(\mathbf{X}) \geq 0 \quad j = 1, \dots, m, \quad (1.2)$$

where  $Q_j(\mathbf{X})$  is a product of linear and quadratic polynomials. These distributions play a central role in models as diverse as the Probit and Tobit models [Albert and Chib, 1993, Tobin, 1958], the dichotomized Gaussian model [Emrich and Piedmonte, 1991, Cox and Wermuth, 2002], stochastic integrate-and-fire neural models [Paninski et al., 2004], Bayesian isotonic regression [Neelon and Dunson, 2004], the Bayesian bridge model expressed as a mixture of Bartlett-Fejer kernels [Polson and Scott, 2011], and many others.

The standard approach to sample from TMGs is the Gibbs sampler [Geweke, 1991, Kotecha and Djuric, 1999]. The latter reduces the problem to one-dimensional truncated Gaussians, for which simple and efficient sampling methods exist [Robert, 1995, Damien and Walker, 2001]. While it enjoys the benefit of having no parameters to tune, the Gibbs sampler can suffer from two problems, which make it inefficient in some situations. Firstly, its runtime scales linearly with the number of dimensions. Secondly, even though a change of variables that maps  $\mathbf{M}$  in (1.1) to the identity often improves the mixing speed [Rodriguez-Yam et al., 2004], the exploration of the target space can still be very slow when the constraints (1.2) impose high correlations among the coordinates. Figure 1 illustrates this effect in a simple example. Improvement over the Gibbs runtime can be obtained with a hit-and-run algorithm [Chen and Deely, 1992], but the latter suffers from the same slow convergence problem when the constraints impose strong correlations.

In this paper we present an alternative algorithm to sample from TMG distributions for constraints  $Q_j(\mathbf{X})$  in (1.2) given by linear or quadratic functions or products thereof, based on the Hamiltonian Monte Carlo (HMC) approach. The HMC method, introduced in Duane et al. [1987], considers the log of the probability distribution as minus the potential energy of a particle, and introduces a Gaussian distribution for momentum variables in order to define a Hamiltonian function. The method generally avoids random walks and mixes faster than Gibbs or Metropolis-Hastings techniques. The HMC sampling procedure alternates between sampling the Gaussian momenta and letting the position of the particle evolve by integrating its Hamiltonian equations of motion. In most models, the latter cannot be integrated exactly, so the resulting position is used as a Metropolis proposal, with an acceptance probability that depends exponentially on the energy gained due to the numerical error. The downside is that two parameters must be fine-tuned for the algorithm to work properly: the integration time-step size and the number of time-steps. In general the

values selected correspond to a compromise between a high acceptance rate and a good rate of exploration of the space [Hoffman and Gelman, 2011]. More details of HMC can be found in the reviews by Kennedy [1990] and Neal [2010].

The case we consider in this work is special because the Hamiltonian equations of motion can be integrated exactly, thus leading to the best of both worlds: HMC mixes fast and, as in Gibbs, there are no parameters to tune and the Metropolis step always accepts (because the energy is conserved exactly). The truncations (1.2) are incorporated via hard walls, against which the particle bounces off elastically. The runtime depends highly on the shape and location of the truncation, as most of the computing time goes into finding the time of the next wall bounce and the direction of the reflected particle. But unlike the Gibbs sampler, these computations are parallelizable, potentially allowing fast implementations.

The idea behind the HMC sampler for a Gaussian in a truncated space turns out to be applicable also when the log-density is piecewise quadratic. We show that a simple extension of the algorithm allows us to sample from such distributions, focusing on the example of the ‘‘Bayesian Lasso’’ model [Park and Casella, 2008].

Previous HMC applications that made use of exactly solvable Hamiltonian equations include sampling from non-trivial integrable Hamiltonians [Kennedy and Bitar, 1994], and importance sampling, with the target distribution approximated by a distribution with an integrable Hamiltonian [Rasmussen, 2003, Izaguirre and Hampton, 2004].

In the next Section we present the new sampling algorithm for linear and quadratic constraints. In Section 3 we present four example applications. In all our examples, the matrix  $\mathbf{M}$  or its inverse have a special structure that allows us to accelerate the runtime of the sampler. In Section 4 we discuss the extension to the Bayesian Lasso model. We have implemented the sampling algorithm in the R package ‘‘tmg,’’ available in the CRAN repository.

## 2 The Sampling Algorithm

### 2.1 Linear Inequalities

Consider first sampling from

$$\log p(\mathbf{X}) = -\frac{1}{2}\mathbf{X} \cdot \mathbf{X} + \text{const.} \quad (2.1)$$

subject to

$$\mathbf{F}_j \cdot \mathbf{X} + g_j \geq 0 \quad j = 1, \dots, m. \quad (2.2)$$

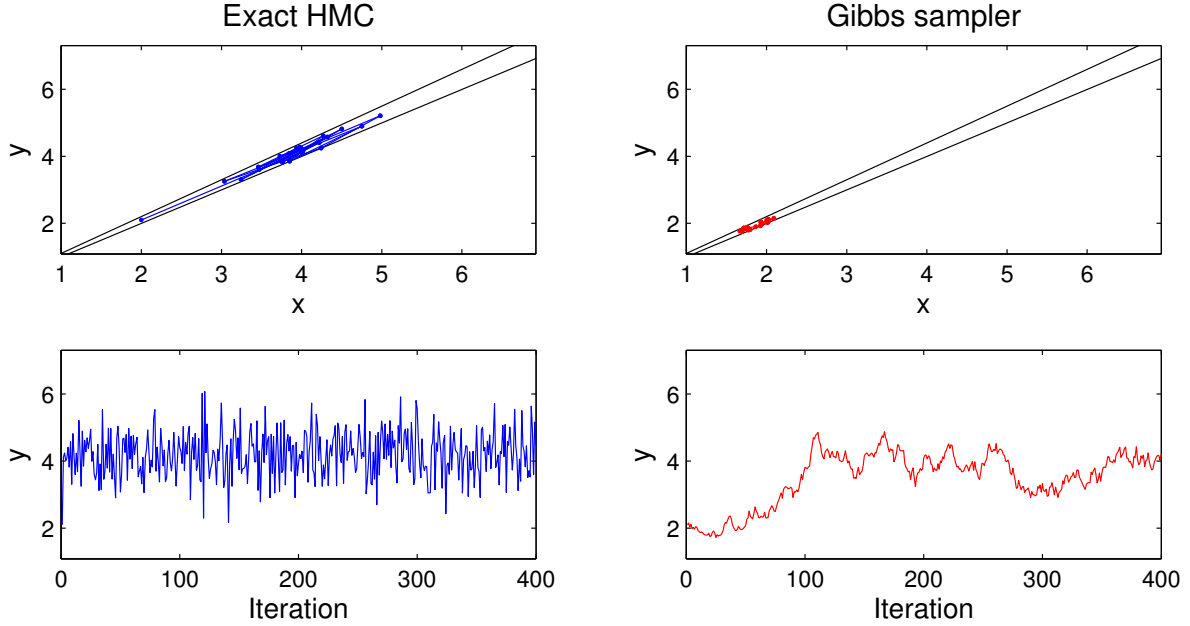
Any quadratic form for  $\log p(\mathbf{X})$ , as in (1.1), can be brought to the above canonical form by a linear change of variables. Let us denote the components of  $\mathbf{X}$  and  $\mathbf{F}_j$  as

$$\mathbf{X} = (x_1, \dots, x_d), \quad (2.3)$$

$$\mathbf{F}_j = (f_j^1, \dots, f_j^d). \quad (2.4)$$

In order to apply the HMC method, we introduce momentum variables  $\mathbf{S}$ ,

$$\mathbf{S} = (s^1, \dots, s^d), \quad (2.5)$$



**Figure 1: HMC vs Gibbs sampler.** Comparison for a two-dimensional distribution with  $\log p(x, y) \propto -\frac{1}{2}(x-4)^2 - \frac{1}{2}(y-4)^2$ , constrained to the wedge  $x \leq y \leq 1.1x$  and  $x, y \geq 0$ . The initial point is  $(x, y) = (2, 2.1)$ . Upper panels: first 20 iterations. Lower panels: Second coordinate of the first 400 iterations. In the HMC case we used  $T = \pi/2$ . The HMC sampler moves rapidly to oscillate around  $y = 4$ , as desired, while the Gibbs sampler mixes relatively slowly.

and consider the Hamiltonian

$$H = \frac{1}{2} \mathbf{X} \cdot \mathbf{X} + \frac{1}{2} \mathbf{S} \cdot \mathbf{S}, \quad (2.6)$$

such that the joint distribution is  $p(\mathbf{X}, \mathbf{S}) = \exp(-H)$ . The equations of motion following from (2.6) are

$$\dot{x}_i = \frac{\partial H}{\partial s^i} = s^i \quad (2.7)$$

$$\dot{s}^i = -\frac{\partial H}{\partial x_i} = -x_i \quad i = 1, \dots, d \quad (2.8)$$

which can be combined to

$$\ddot{x}_i = -x_i, \quad (2.9)$$

and have a solution

$$x_i(t) = a_i \sin(t) + b_i \cos(t). \quad (2.10)$$

The constants  $a_i, b_i$  can be expressed in terms of the initial conditions as

$$a_i = \dot{x}_i(0) = s^i(0) \quad (2.11)$$

$$b_i = x_i(0) \quad (2.12)$$

The HMC algorithm proceeds by alternating between two steps. In the first step we sample  $\mathbf{S}$  from  $p(\mathbf{S}) = \mathcal{N}(0, \mathbb{I}_d)$ . In the second step we use this  $\mathbf{S}$  and the last value of  $\mathbf{X}$  as initial conditions, and let the particle move during a time  $T$ , after which the position and momentum have values  $\mathbf{X}^*$  and  $\mathbf{S}^*$ . The value  $\mathbf{X}^*$  belongs to a Markov chain with equilibrium distribution  $p(\mathbf{X})$ . To see this, note that for a given  $T$ , the particle trajectory is deterministic once the momentum  $\mathbf{S}$  is sampled, so the transition probability is

$$p(\mathbf{X}^*|\mathbf{X}, T) = p(\mathbf{S}) |\partial\mathbf{S}/\partial\mathbf{X}^*|. \quad (2.13)$$

Two important properties of Hamiltonian dynamics are the conservation of energy

$$H(\mathbf{X}, \mathbf{S}) = H(\mathbf{X}^*, \mathbf{S}^*) \quad (2.14)$$

and the conservation of volume in phase space

$$d\mathbf{S}^* d\mathbf{X}^* = d\mathbf{S} d\mathbf{X} \quad (2.15)$$

which implies the equation

$$|\partial\mathbf{S}/\partial\mathbf{X}^*| = |\partial\mathbf{S}^*/\partial\mathbf{X}|. \quad (2.16)$$

From the above results the detailed balanced condition follows as

$$p(\mathbf{X})p(\mathbf{X}^*|\mathbf{X}, T) = p(\mathbf{X})p(\mathbf{S})|\partial\mathbf{S}/\partial\mathbf{X}^*| \quad (2.17)$$

$$= e^{-H(\mathbf{X}, \mathbf{S})} |\partial\mathbf{S}/\partial\mathbf{X}^*| \quad (2.18)$$

$$= e^{-H(\mathbf{X}^*, -\mathbf{S}^*)} |\partial\mathbf{S}^*/\partial\mathbf{X}| \quad (2.19)$$

$$= p(\mathbf{X}^*)p(\mathbf{X}|\mathbf{X}^*, T) \quad (2.20)$$

where we used the invariance of  $H$  under  $\mathbf{S} \rightarrow -\mathbf{S}$ . We will discuss the appropriate choice for  $T$  in Section 2.4.

The trajectory of the particle is given by (2.10) until it hits a wall, and this occurs whenever any of the inequalities (2.2) is saturated. To find the time at which this occurs, it is convenient to define

$$K_j(t) = \sum_{i=1}^d f_j^i x_i(t) + g_j \quad j = 1, \dots, m. \quad (2.21)$$

$$= \sum_{i=1}^d f_j^i a_i \sin(t) + \sum_{i=1}^d f_j^i b_i \cos(t) + g_j \quad (2.22)$$

$$= u_j \cos(t + \varphi_j) + g_j \quad (2.23)$$

where

$$u_j = \sqrt{\left(\sum_{i=1}^d f_j^i a_i\right)^2 + \left(\sum_{i=1}^d f_j^i b_i\right)^2}, \quad (2.24)$$

$$\tan \varphi_j = -\frac{\sum_{i=1}^d f_j^i a_i}{\sum_{i=1}^d f_j^i b_i}. \quad (2.25)$$

Along the trajectory we have  $K_j(t) \geq 0$  for all  $j$  and a wall hit corresponds to  $K_j(t) = 0$ , so from (2.23) it follows that the particle can only reach those walls satisfying  $u_j > |g_j|$ . Each one of those reachable walls has associated two times  $t_j > 0$  such that

$$K_j(t_j) = 0, \quad (2.26)$$

and the actual wall hit corresponds to the smallest of all these times. Suppose that the latter occurs for  $j = h$ . At the hitting point, the particle bounces off the wall and the trajectory continues with a reflected velocity. The latter can be obtained by noting that the vector  $\mathbf{F}_h$  is perpendicular to the reflecting plane. Let us decompose the velocity as

$$\dot{\mathbf{X}}(t_h) = \dot{\mathbf{X}}_{\perp}(t_h) + \alpha_h \mathbf{F}_h, \quad (2.27)$$

where  $\mathbf{F}_h \cdot \dot{\mathbf{X}}_{\perp}(t_h) = 0$  and

$$\alpha_h = \frac{\mathbf{F}_h \cdot \dot{\mathbf{X}}(t_h)}{\|\mathbf{F}_h\|^2}. \quad (2.28)$$

The reflected velocity,  $\dot{\mathbf{X}}_R(t_h)$ , is obtained by inverting the component perpendicular to the reflecting plane

$$\dot{\mathbf{X}}_R(t_h) = \dot{\mathbf{X}}_{\perp}(t_h) - \alpha_h \mathbf{F}_h, \quad (2.29)$$

$$= \dot{\mathbf{X}}(t_h) - 2\alpha_h \mathbf{F}_h. \quad (2.30)$$

It is easy to verify that this transformation leaves the Hamiltonian (2.6) invariant. Once the reflected velocity is computed, we use it as an initial condition in (2.11) to continue the particle trajectory.

If we prefer to keep the original distribution in the form (1.1), we should consider the Hamiltonian

$$H = \frac{1}{2} \mathbf{X}^T \mathbf{M} \mathbf{X} - \mathbf{r}^T \mathbf{X} + \frac{1}{2} \mathbf{S}^T \mathbf{M}^{-1} \mathbf{S}. \quad (2.31)$$

This election for the mass matrix leads to the simple solution

$$x_i(t) = \mu_i + a_i \sin(t) + b_i \cos(t), \quad (2.32)$$

where

$$\mu_i = \sum_{j=1}^d M_{ij}^{-1} r_j, \quad (2.33)$$

$$a_i = \dot{x}_i(0) = \sum_{j=1}^d M_{ij}^{-1} s_j(0), \quad (2.34)$$

$$b_i = x_i(0) - \mu_i. \quad (2.35)$$

Since the particle trajectory depends on  $\dot{x}_i(0)$ , we can start each iteration by sampling  $\dot{x}_i(0) \sim \mathcal{N}(0, \mathbf{M}^{-1})$ , instead of sampling  $s_i(0) \sim \mathcal{N}(0, \mathbf{M})$  itself.

## 2.2 The runtime of the sampler

The runtime of each iteration has a contribution that scales linearly with  $m$ , the number constraints, since we have to compute the  $m$  values  $u_j$  at (2.24). But the dominant computational time goes to compute  $\varphi_j$  and  $t_j$ , defined in (2.25) and (2.26), which only needs to be done when  $u_j > |g_j|$ . The average number of coordinates for which this condition occurs, as well as the number of times the particle hits the walls per iteration, varies according to the value of  $T$  and the shape and location of the walls.

The sums in expressions (2.24) and (2.25) can be interpreted as matrix-vector multiplications, with cost  $O(md)$  for general constraint matrices  $\mathbf{F} = (\mathbf{F}_1^T \mathbf{F}_2^T \dots \mathbf{F}_m^T)^T$ . Note that these matrix-vector multiplications are highly parallelizable. In addition, in many cases there may be some special structure that can be exploited to speed computation further; for example, if  $\mathbf{F}$  can be expressed as a sparse matrix in a convenient basis, this cost can be reduced to  $O(d)$ .

Also, in both frames (2.6) and (2.31) we must act, for each sample, with a matrix  $R^{-1}$ , where  $\mathbf{M} = R^T R$ . Equivalently, we must multiply by  $Z^T$ , where  $\mathbf{M}^{-1} = \Sigma = Z^T Z$ . In the frame (2.6) this is needed because the samples must be mapped back to the original frame (1.1) as

$$\mathbf{X} \rightarrow R^{-1} \mathbf{X} + \mathbf{M}^{-1} \mathbf{r}, \quad (2.36)$$

and in the frame (2.31) the action of  $R^{-1}$  is needed in order to sample the initial velocity from  $\mathcal{N}(0, \mathbf{M}^{-1})$  at each iteration. The action of  $R^{-1}$  takes  $O(d^2)$  generally, but in some cases the matrix  $\mathbf{M}$  or its inverse have a special structure that allows us to accelerate this operation. This is the case in the four example applications we present in Section 3.

Regarding which of the two frames (2.6) and (2.31) is preferred, this depends on the nature of the constraints, since the latter change when we transform the quadratic form for  $\log p(\mathbf{X})$  in (1.1) to the form (2.1). For example, a sparse set of constraints  $\mathbf{F}$  in the original frame leads to a fast evaluation of (2.24) and (2.25), but the wall geometry in the transformed frame may lead to a smaller number of wall hits and therefore shorter runtime.

## 2.3 Quadratic and Higher Order Inequalities

The sampling algorithm can be extended in principle to polynomial constraints of the form

$$Q_j(\mathbf{X}) \geq 0 \quad j = 1, \dots, m. \quad (2.37)$$

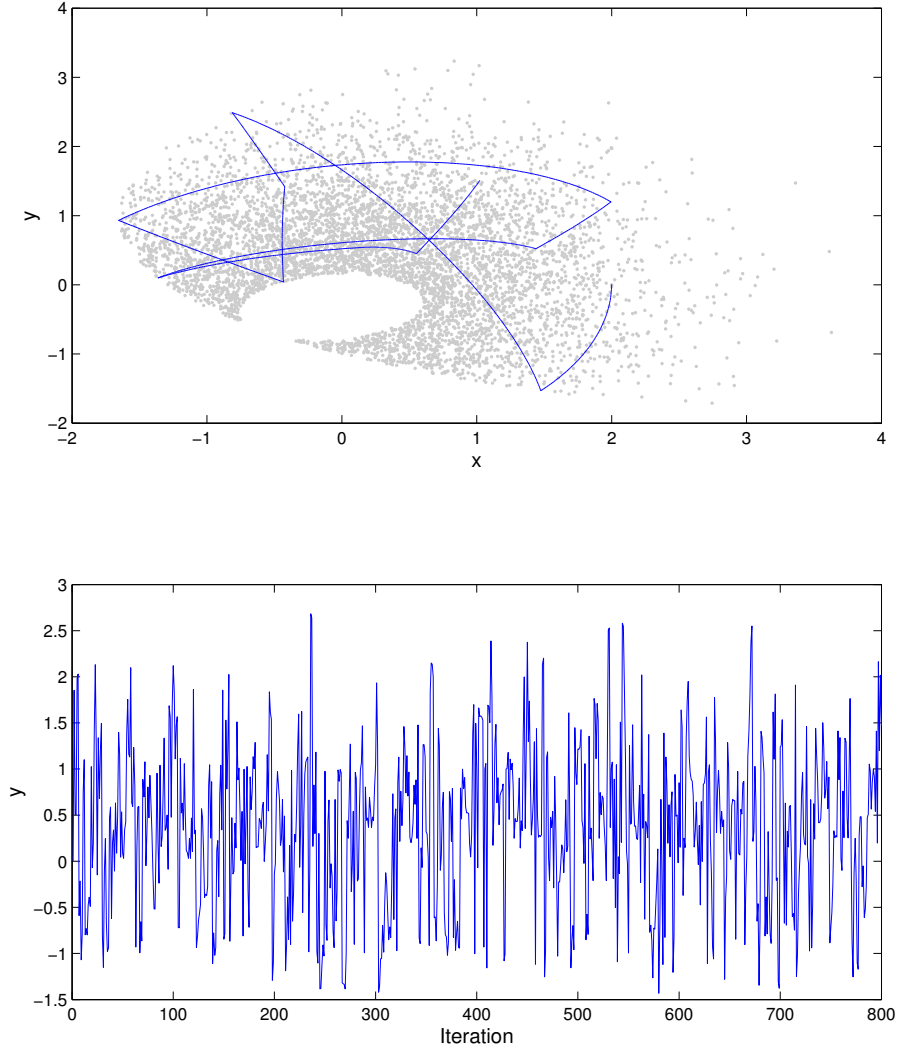
Evaluating  $Q_j(\mathbf{X})$  at the solution (2.10) leads to a polynomial in  $\sin(t)$  and  $\cos(t)$ , whose zeros must be found in order to find the hitting times. When a wall is hit, we reflect the velocity by inverting the sign of the component perpendicular to the wall, given by the gradient  $\nabla Q_j(\mathbf{X})$ . This vector plays a role similar to  $\mathbf{F}_j$  in (2.27)-(2.30). Of course, for general polynomials  $Q_j(\mathbf{X})$  computing the hitting times might be numerically challenging.

One family of solvable constraints involves quadratic inequalities of the form

$$Q_j(\mathbf{X}) = \mathbf{X}^T \mathbf{A}_j \mathbf{X} + \mathbf{X} \cdot \mathbf{B}_j + C_j \geq 0 \quad j = 1, \dots, m, \quad (2.38)$$

where  $\mathbf{A}_j \in \mathbb{R}^{d,d}$ ,  $\mathbf{B}_j \in \mathbb{R}^d$ ,  $C_j \in \mathbb{R}$ . For statistics applications where these constraints are important, see e.g. Ellis and Maitra [2007]. Inserting (2.10) in the equality for (2.38) leads, for each  $j$ , to the following equation for the hitting time:

$$q_1 \cos^2(t) + q_2 \cos(t) + q_3 = -\sin(t)(q_4 \cos(t) + q_5), \quad (2.39)$$



**Figure 2: Truncation by quadratic inequalities.** Above: 6000 samples of a two-dimensional canonical normal distribution, constrained by the quadratic inequalities (2.54) -(2.55). The piecewise elliptic curve shows the trajectory of the particle in the first iterations, with starting point  $(x, y) = (2, 0)$ . Below: first 800 iterations of the vertical coordinate. For the algebraic solution of (2.48), we used the C++ code from the DynamO package [Bannerman et al., 2011].

with

$$q_1 = \sum_{i,k} A^{ik} b_i b_k - \sum_{ik} A^{ik} a_i a_k, \quad (2.40)$$

$$q_2 = \sum_i B^i b_i, \quad (2.41)$$

$$q_3 = C + \sum_{ik} A^{ik} a_i a_k, \quad (2.42)$$

$$q_4 = 2 \sum_{i,k} A^{ik} a_i b_k, \quad (2.43)$$

$$q_5 = \sum_i B^i a_i \quad (2.44)$$



and we omitted the  $j$  dependence to simplify the notation. If the ellipse in (2.38) is centered at the origin, we have  $\mathbf{B}_j = q_2 = q_5 = 0$ , and equation (2.39) simplifies to

$$q_1 + 2q_3 + u \sin(2t + \varphi) = 0 \quad (2.45)$$

where

$$u^2 = q_1^2 + q_4^2, \quad (2.46)$$

$$\tan \varphi = \frac{q_1}{q_4}, \quad (2.47)$$

and the hit time can be found from (2.45) as in the linear case. In the general  $\mathbf{B}_j \neq 0$  case, the square of (2.39) gives the quartic equation

$$r_4 \cos^4(t) + r_3 \cos^3(t) + r_2 \cos^2(t) + r_1 \cos(t) + r_0 = 0, \quad (2.48)$$

where

$$r_4 = q_1^2 + q_4^2, \quad (2.49)$$

$$r_3 = 2q_1q_2 + 2q_4q_5, \quad (2.50)$$

$$r_2 = q_2^2 + 2q_1q_3 + q_5^2 - q_4^2, \quad (2.51)$$

$$r_1 = 2q_2q_3 - 2q_4q_5, \quad (2.52)$$

$$r_0 = q_3^2 - q_5^2. \quad (2.53)$$

Equation (2.48) can be solved exactly for  $\cos(t)$  using standard algebraic methods [Herbison-Evans, 1994]. A wall hit corresponds, among all the constraints  $j$ , to the solution for  $\cos(t)$  with smallest  $t > 0$  and  $|\cos(t)| \leq 1$ , which also solves (2.39). As an example, Figure 2 shows samples from a two-dimensional canonical normal distribution, constrained by

$$\frac{(x-4)^2}{32} + \frac{(y-1)^2}{8} \leq 1, \quad (2.54)$$

$$4x^2 + 8y^2 - 2xy + 5y \geq 1. \quad (2.55)$$

Equipped with the results for linear and quadratic constraints, we can also find the hitting times for constraints of the form

$$Q(\mathbf{X}) = \prod_j Q_j(\mathbf{X}) \geq 0 \quad (2.56)$$

where each  $Q_j(\mathbf{X})$  is a linear or a quadratic function. Each factor defines an equation as (2.26) or (2.48), and the hitting time is the smallest at which any factor becomes zero. For other polynomials, one has to resort to numerical methods to find the hitting times.

## 2.4 Travel time and efficiency

The value of  $T$  should be chosen to make the sampling as efficient as possible. This is not easy because the path of the particle will be determined by the wall bounces, so it is difficult to compute in advance. A safe strategy is to choose a value of  $T$  that makes the sampling efficient at least for

those trajectories with no wall hits. The efficiency can be quantified via the Effective Sample Factor (ESF) and Effective Sample Size (ESS) [Liu, 2008]. Let us call the samples  $\mathbf{X}^{(p)}$ . The variance in the estimation of the expected value of a function  $h(\mathbf{X})$  using  $m$  samples is

$$\text{var}\left(\frac{h(\mathbf{X}^{(1)}) + \dots + h(\mathbf{X}^{(m)})}{m}\right) = \frac{\text{var}(h(\mathbf{X}))}{m} \left[1 + \sum_{j=1}^{m-1} 2\left(1 - \frac{j}{m}\right)\rho_j\right] \quad (2.57)$$

with  $\rho_j$  the autocorrelation function (ACF),

$$\rho_j = \text{corr}[h(\mathbf{X}^{(1)}), h(\mathbf{X}^{(1+j)})]. \quad (2.58)$$

The ESF and ESS are defined as

$$\text{ESF} = \left[1 + \sum_{j=1}^{m-1} 2\left(1 - \frac{j}{m}\right)\rho_j\right]^{-1}, \quad (2.59)$$

$$\text{ESS} = m \times \text{ESF}, \quad (2.60)$$

and a sampling scheme is more efficient when its ESF is higher, because that leads to a lower variance in (2.57).

For concreteness, let us consider the estimation of the mean of a coordinate  $x_i$  in the frame in which the Hamiltonian is given by (2.6). In the absence of wall hits, from (2.10)-(2.11) it follows that successive samples are given by

$$x_i^{(p+1)} = x_i^{(p)} \cos(T) + s_i^{(p)} \sin(T). \quad (2.61)$$

Assuming that the momenta samples are i.i.d., i.e.,

$$\langle s_i^{(p)} s_i^{(q)} \rangle = \delta_{pq}, \quad (2.62)$$

it is easy to see that the ACF is given by

$$\rho_j = \langle x_i^{(1)} x_i^{(j+1)} \rangle = \cos^j(T). \quad (2.63)$$

In principle we could plug this expression into (2.59) and find the value of  $T$  that maximizes the ESF. In particular, when  $\cos(T) < 0$ , there are values of  $T$  which lead to a super-efficient sampler with  $\text{ESF} > 1$ , so that the variance (2.57) is *smaller* than using i.i.d. samples (i.e., with  $\rho_j = 0$ ). This is similar to the antithetic variates method to reduce the variance of Monte Carlo estimates [Hammersley and Morton, 1956].

In practice, we have found that the above strategy is not very useful. The assumption (2.62) holds only approximately for standard generators of random variables, and this leads to very unstable values for the ESF for any value of  $T$ . A safe choice is to use  $T = \pi/2$ , which leads to

$$x_i^{(p+1)} = s_i^{(p)}, \quad (2.64)$$

so our sampling of the space  $\mathbf{X}$ , for trajectories with no wall hits, will be as efficient as our method to sample the momenta  $\mathbf{S}$ . Since the i.i.d. assumption holds approximately, this leads to values of ESF that bounce around 1. The ESF in a general case will also depend on the shape and location

	Avg. CPU Time		$y$
HMC $_{T=\pi/2}$	3.78 secs	ESF	<b>2.7 (1.18/5.11)</b>
		ESS/CPU	<b>5,802 (2,507/10,096)</b>
HMC $_{T=\pi/10}$	1.38 secs	ESF	0.052 (0.033/0.094)
		ESS/CPU	301 (191/547)
Gibbs	0.57 secs	ESF	0.017 (0.011/0.032)
		ESS/CPU	237 (154/456)

**Table 1: HMC vs Gibbs sampler.** Comparison of efficiency criteria for 30 runs of three samplers in the example of Figure 1. For the definition of ESF and ESS, see Section 2.4. ESS/CPU is the Effective Sample Size in units of the CPU runtime. A sampler is more efficient for higher ESF and ESS/CPU. In each run, we used 8,000 samples, after discarding 2,000 samples as burn-in. The CPU time for each run is random, but its variability across runs is negligible due to the large number of samples. For ESF and ESS/CPU, we show median (first/third quartiles). For the Gibbs sampler, we used the algorithm of [Damien and Walker, 2001]. The runtime for HMC is relatively high due to the many reflections of the particle against the walls. Note that the efficiency of the HMC sampler depends strongly on the value of the travel time  $T$ .

of the constraint walls. But we have found that the super-efficient case  $\text{ESF} > 1$  is not uncommon, although as mentioned, the actual value of the ESF is quite unstable. Table 1 illustrates the big difference in the efficiency of the HMC sampler for  $T = \pi/2$  and  $T = \pi/10$  in the two-dimensional example of Figure 1. A similar reasoning suggests the use of  $T = \pi/2$  also when working in the frame in which the Hamiltonian is given by (2.31). In the next two Sections we adopted  $T = \pi/2$ , and we rotated the coordinates to a canonical frame with Hamiltonian (2.6).

In Tables 1 and 2 we compare the efficiency of our HMC method to the Gibbs sampler. To implement the latter efficiently we made two choices. Firstly, we rotated the coordinates to a canonical frame in which the unconstrained Gaussian has unit covariance. This transformation often makes the Gibbs sampler mix faster [Rodriguez-Yam et al., 2004]. For the Gibbs sampler itself, we used the slice sampling version of [Damien and Walker, 2001]. This algorithm augments by one the number of variables, but turns the conditional distributions of the coordinates of interest into uniform distributions. The latter are much faster to sample than the truncated one-dimensional Gaussians one gets otherwise. We checked that the efficiency of the algorithm of [Damien and Walker, 2001], measured in units of ESS/CPU time, is much higher than that of the Gibbs sampler based on direct sampling from the one-dimensional truncated normal using inverse cdf or rejection sampling.

### 3 Examples

In this Section we present four example applications of our algorithm. In the first example, we present a detailed efficiency comparison between the HMC and the Gibbs samplers. As mentioned in Section 2.2, in both frames (2.6) and (2.31), for each sample of the HMC we must act with a matrix  $R^{-1}$ , where  $\mathbf{M} = R^T R$ , or multiply by  $Z^T$ , where  $\mathbf{M}^{-1} = \Sigma = Z^T Z$ . In all our examples, we show how some special structure of  $\mathbf{M}$  or  $\Sigma$  allows us to accelerate these operations.

### 3.1 Probit and Tobit Models

The Probit model is a popular discriminative probabilistic model for binary classification with continuous inputs [Albert and Chib, 1993]. The conditional probabilities for the binary labels  $y = \pm 1$  are given by

$$p(y = -1|\mathbf{z}, \boldsymbol{\beta}) = \Phi(\mathbf{z} \cdot \boldsymbol{\beta}) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\mathbf{z} \cdot \boldsymbol{\beta}} dw e^{-\frac{w^2}{2}} \quad (3.1)$$

$$= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^0 dw e^{-\frac{(w+\mathbf{z} \cdot \boldsymbol{\beta})^2}{2}} \quad (3.2)$$

$$p(y = +1|\mathbf{z}, \boldsymbol{\beta}) = 1 - \Phi(\mathbf{z} \cdot \boldsymbol{\beta}) \quad (3.3)$$

$$= \frac{1}{\sqrt{2\pi}} \int_0^{+\infty} dw e^{-\frac{(w+\mathbf{z} \cdot \boldsymbol{\beta})^2}{2}} \quad (3.4)$$

where  $\mathbf{z} \in \mathbb{R}^p$  is a vector of regressors and  $\boldsymbol{\beta} \in \mathbb{R}^p$  are the parameters of the model. Given  $N$  pairs of labels and regressors

$$\mathbf{Y} = (y_1, \dots, y_N), \quad (3.5)$$

$$\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_N), \quad (3.6)$$

the posterior distribution of the parameters  $\boldsymbol{\beta}$  is

$$p(\boldsymbol{\beta}|\mathbf{Y}, \mathbf{Z}) \propto p(\boldsymbol{\beta}) \prod_{i=1}^N p(y_i|\mathbf{z}_i, \boldsymbol{\beta}) \quad (3.7)$$

$$\propto p(\boldsymbol{\beta}) \int_{y_i w_i \geq 0} dw_1 \dots dw_N e^{-\frac{1}{2} \sum_{i=1}^N (w_i + \mathbf{z}_i \cdot \boldsymbol{\beta})^2} \quad i = 1, \dots, N, \quad (3.8)$$

where  $p(\boldsymbol{\beta})$  is the prior distribution. The likelihood  $p(y_i|\mathbf{z}_i, \boldsymbol{\beta})$  corresponds to a model

$$y_i = \text{sign}(w_i) \quad (3.9)$$

$$w_i = -\mathbf{z}_i \cdot \boldsymbol{\beta} + \varepsilon_i \quad (3.10)$$

$$\varepsilon_i \sim \mathcal{N}(0, 1) \quad (3.11)$$

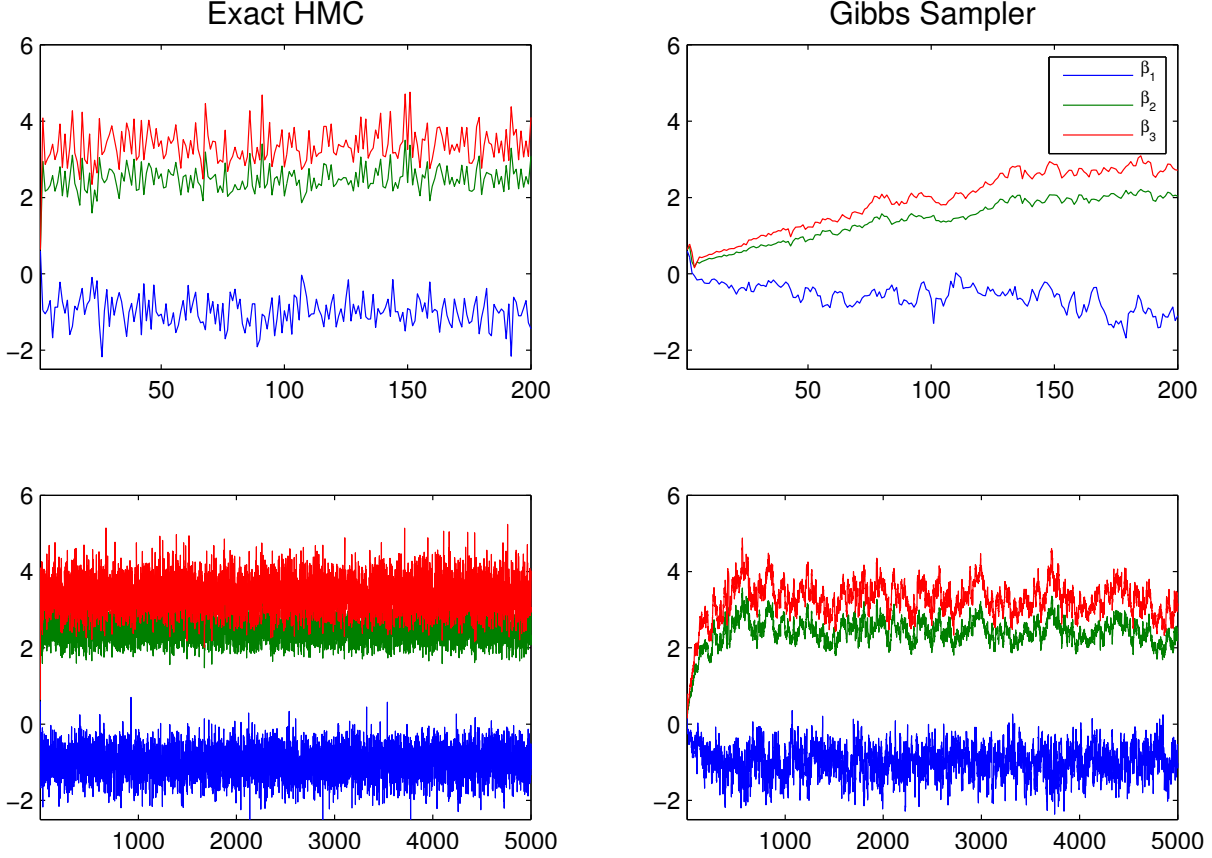
in which only the sign of  $w_i$  is observed, but not its value. Assuming a Gaussian prior with zero mean and covariance  $\sigma^2 \mathbb{I}_p$ , expression (3.8) is the marginal distribution of a multivariate Gaussian on  $(\boldsymbol{\beta}, w_1, \dots, w_N)$ , truncated to  $y_i w_i \geq 0$  for  $i = 1, \dots, N$ . The untruncated Gaussian has zero mean and precision matrix

$$\mathbf{M} = \begin{pmatrix} M_{\boldsymbol{\beta}\boldsymbol{\beta}} & M_{\boldsymbol{\beta}w} \\ M_{w\boldsymbol{\beta}} & M_{ww} \end{pmatrix} \in \mathbb{R}^{p+N, p+N} \quad (3.12)$$

$$= \begin{pmatrix} \sigma^{-2} \mathbb{I}_p + BB^T & B \\ B^T & \mathbb{I}_N \end{pmatrix} \quad (3.13)$$

where

$$B^T = \begin{pmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_N \end{pmatrix} \in \mathbb{R}^{N, p} \quad (3.14)$$



**Figure 3: Bayesian Probit model.** First 200 and 5000 samples from the posterior (3.8) of a model with  $p = 3$ .  $N = 800$  pairs  $(y_i, z_i)$  were generated with  $\beta_1 = -9, \beta_2 = 20, \beta_3 = 27$  and we assumed a Gaussian prior with zero mean and  $\sigma^2 = 1$ . Note that the means of the sampled values are different from the values used to generate the data, due to the zero-mean prior. Left: Exact HMC sampler. Right: Gibbs sampler, with whitened covariance to improve mixing [Rodriguez-Yam et al., 2004].

We can sample from the posterior in (3.8) by sampling from the truncated Gaussian for  $(\beta, w_1, \dots, w_N)$  and keeping only the  $\beta$  values. It is easy to show that without the term  $\sigma^{-2}\mathbb{I}_p$  in (3.13), coming from the prior  $p(\beta)$ , the precision matrix would have  $p$  null directions and our method would not be applicable, since we assume the precision matrix to be positive definite. Note that the dimension of the TMG grows linearly with the number  $N$  of data points.

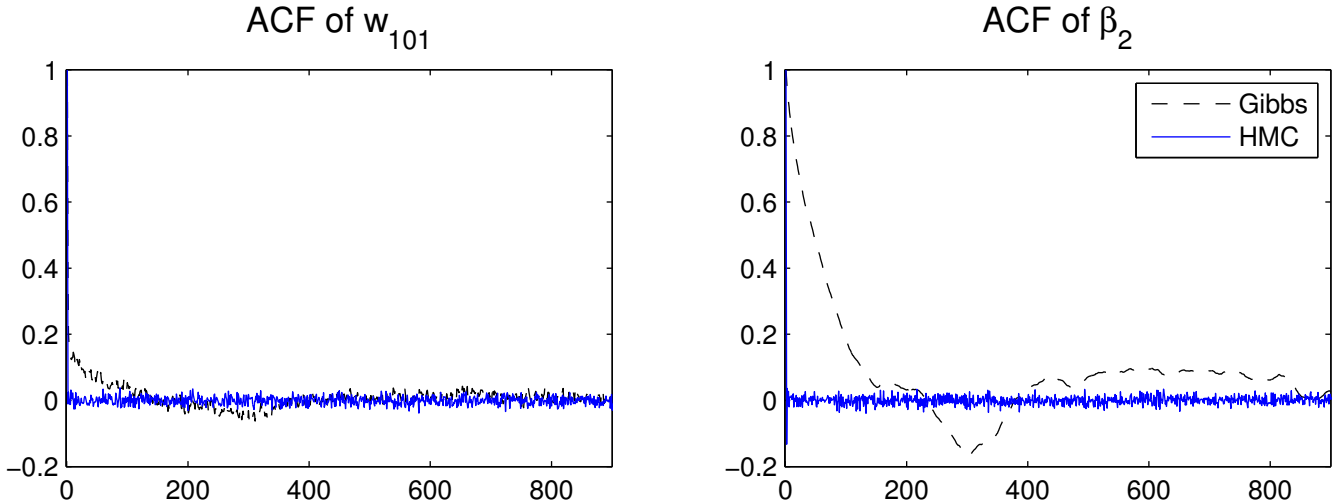
The structure of (3.13) leads to a simple form for the untruncated covariance  $\Sigma = \mathbf{M}^{-1}$ ,

$$\mathbf{M}^{-1} = \begin{pmatrix} \mathbb{I}_p & 0 \\ -B^T & \mathbb{I}_N \end{pmatrix} \begin{pmatrix} \sigma^2 \mathbb{I}_p & 0 \\ 0 & \mathbb{I}_N \end{pmatrix} \begin{pmatrix} \mathbb{I}_p & -B \\ 0 & \mathbb{I}_N \end{pmatrix} \quad (3.15)$$

$$= Z^T Z \quad (3.16)$$

where

$$Z^T = \begin{pmatrix} \mathbb{I}_p & 0 \\ -B^T & \mathbb{I}_N \end{pmatrix} \begin{pmatrix} \sigma \mathbb{I}_p & 0 \\ 0 & \mathbb{I}_N \end{pmatrix}, \quad (3.17)$$



**Figure 4: Autocorrelation functions.** Autocorrelation functions (ACFs) for the first 900 lags of two of the 803 variables sampled in the Probit example. See Table 2 for a comparison of efficiency measures of these two variables between the two samplers.

and this form is such that acting with  $Z^T$  takes  $O(N)$  time, instead of  $O(N^2)$ .

As an illustration, Figure 3 shows the values of  $\beta$ , sampled using Gibbs and exact HMC, from the posterior of a model with  $p = 3$  where  $N = 800$  data points were generated. We used  $z_i^1 = 1$ ,  $z_i^2 \sim \text{Unif}[-5, +5]$  and  $z_i^3 \sim \mathcal{N}(-4, \sigma = 4)$ . The values of  $y_i$  were generated with  $\beta_1 = -9, \beta_2 = 20, \beta_3 = 27$  and we assumed a Gaussian prior with  $\sigma^2 = 1$ . Note that the means of the sampled  $\beta_i$ 's are different from the  $\beta_i$ 's used to generate the data, due to the prior which pulls the  $\beta_i$ 's towards zero.

A more quantitative comparison between the HMC and the Gibbs samplers is presented in Figure 4, which compares the autocorrelation functions (ACFs) for two of the variables of the Probit model, and in Table 2, which compares, for the same variables, the Effective Sample Factors (ESFs) and the Effective Sample Size (ESS) normalized by CPU time. For the latter, we see a remarkable difference of two to three orders of magnitude between the two samplers.

A model related to the Probit is the Tobit model for censored data [Tobin, 1958], which is a linear regression model where negative values are not observed:

$$y_i = \begin{cases} y_i^* & \text{for } y_i^* > 0, \\ 0 & \text{for } y_i^* \leq 0, \end{cases} \quad (3.18)$$

where

$$y_i^* = \mathbf{z}_i \cdot \boldsymbol{\beta} + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma). \quad (3.19)$$

The likelihood of a pair  $(y_i, \mathbf{z}_i)$  is

$$p(y_i | \mathbf{z}_i, \boldsymbol{\beta}, \sigma) = \begin{cases} \frac{e^{-\frac{(y_i - \mathbf{z}_i \cdot \boldsymbol{\beta})^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} & \text{for } y_i > 0, \\ \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^0 dw_i e^{-\frac{(w_i - \mathbf{z}_i \cdot \boldsymbol{\beta})^2}{2\sigma^2}} & \text{for } y_i = 0, \end{cases} \quad (3.20)$$

	Avg. CPU time		$w_{101}$	$\beta_2$
HMC	235.5 secs.	ESF	<b>1.96 (1.28/7.03)</b>	<b>2.65 (1.28/3.69)</b>
		ESS/CPU	<b>49.9 (32.6/178.6)</b>	<b>67.69 (32.6/92.84)</b>
Gibbs	651.7 secs.	ESF	0.037 (0.027/0.054)	0.0051 (0.0036/0.0068)
		ESS/CPU	0.34 (0.22/0.50)	0.047 (0.033/0.063)

**Table 2: Efficiency of HMC vs. Gibbs in the Probit example.** *The results summarize 10 runs of each sampler, and correspond to the means of the variables  $w_{101}$  and  $\beta_2$  from the 803-dimensional Probit example. The autocorrelation functions of these variables are shown in Figure 4. In each run we used 6,000 samples, after discarding 2,000 samples as burn-in. Again, the CPU time for each run is random, but its variability across runs is negligible due to the large number of samples. For ESF and ESS/CPU, we show median (first/third quartiles). ESS/CPU is the Effective Sample Size in units of the CPU runtime. For Gibbs we used the algorithm of [Damien and Walker, 2001]. Note that the efficiency criteria are fairly variable across runs, but the HMC ESS/CPU is consistently two to three orders of magnitude bigger than Gibbs. The Gibbs sampler shows a big difference in the efficiency between the two variables (cf. their ACFs in Figure 4), while this difference is minor in the HMC case. The HMC runtime can be reduced further by parallelizing the computation of the travel times to hit each wall.*

and the posterior probability for  $\beta$  is

$$p(\beta|\mathbf{Y}, \mathbf{Z}, \sigma) \propto p(\beta|\sigma) \prod_{i=1}^N p(y_i|\mathbf{z}_i, \beta, \sigma) \quad (3.21)$$

$$\propto p(\beta|\sigma) \prod_{i, y_i > 0} e^{-\frac{(y_i - \mathbf{z}_i \cdot \beta)^2}{2\sigma^2}} \prod_{i, y_i = 0} \int_{-\infty}^0 dw_i e^{-\frac{(w_i - \mathbf{z}_i \cdot \beta)^2}{2\sigma^2}} \quad (3.22)$$

As in (3.8), this can be treated as a marginal distribution over the variables  $w_i$ , with the joint distribution for  $(\beta, w_i)$  a truncated multivariate Gaussian.

One can similarly consider multivariate Probit [Ashford and Sowden, 1970] and Tobit models. For the former, the Bayesian approach has been shown to be superior to Maximum Likelihood [Geweke et al., 1994].

### 3.2 Sample paths in a Brownian bridge

A family of cases where a simple special structure for  $\mathbf{M}$  arises are linear-Gaussian state-space models with constraints on the hidden variables. For concreteness, we will study the case of the Brownian bridge. Consider the following discrete stochastic process

$$V_t = V_{t-1} + \varepsilon_t \quad \varepsilon_t \sim \mathcal{N}(0, \sigma^2) \quad t = 1, \dots, T \quad (3.23)$$

$$V_0 = L \leq H \quad (3.24)$$

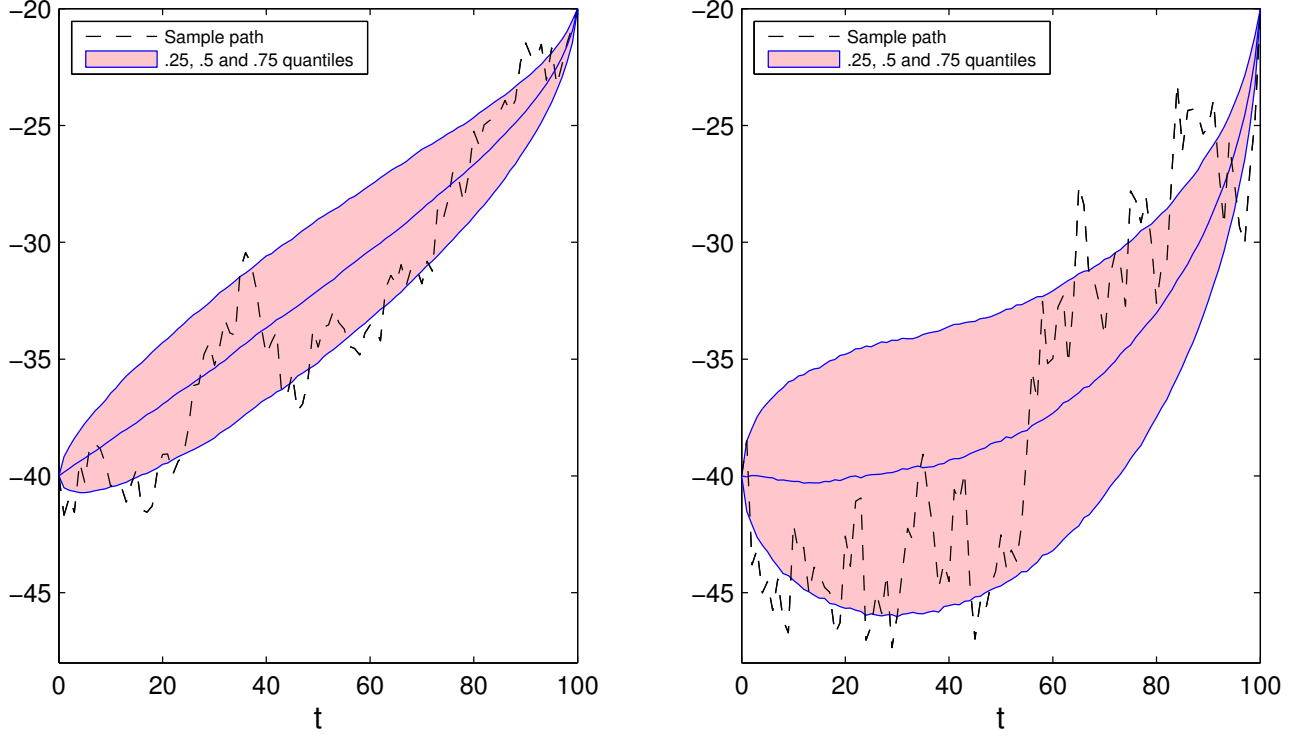
$$V_t < V_T \quad \text{for } t < T \quad (3.25)$$

$$V_T = H \quad (3.26)$$

The starting point  $V_0 = L$  is fixed and the number of steps  $T$  until the first hit of  $H$  is a random variable. This process is called the Brownian bridge and has applications, among others, in finance







**Figure 5: Sample paths in a Brownian bridge.** Each figure shows one sample path and the median and .25 and .75 quantiles at each  $t$  from 15,000 sample paths (after 500 burn-in paths) of a Brownian bridge model with  $L = -40$ ,  $H = -20$  and  $T = 100$ . Left: low noise,  $\sigma^2 = 1$ . Right: high noise,  $\sigma^2 = 5$ .

To obtain a sparse constraint matrix, it is convenient to use the B-spline basis, in which only four elements in the vector  $\phi(z_j)$  are non-zero for any  $j$  (see, e.g. [De Boor, 2001] for details). In a Bayesian approach, we are interested in sampling from the posterior distribution

$$p(\mathbf{a}, \sigma^2 | \mathbf{Y}, \mathbf{X}, \lambda) \propto p(\mathbf{Y} | \mathbf{X}, \mathbf{a}, \sigma^2) p(\mathbf{a} | \lambda, \sigma^2) p(\sigma^2), \quad (3.37)$$

where we defined

$$\mathbf{Y} = (y_1, \dots, y_N), \quad (3.38)$$

$$\mathbf{X} = (x_1, \dots, x_N). \quad (3.39)$$

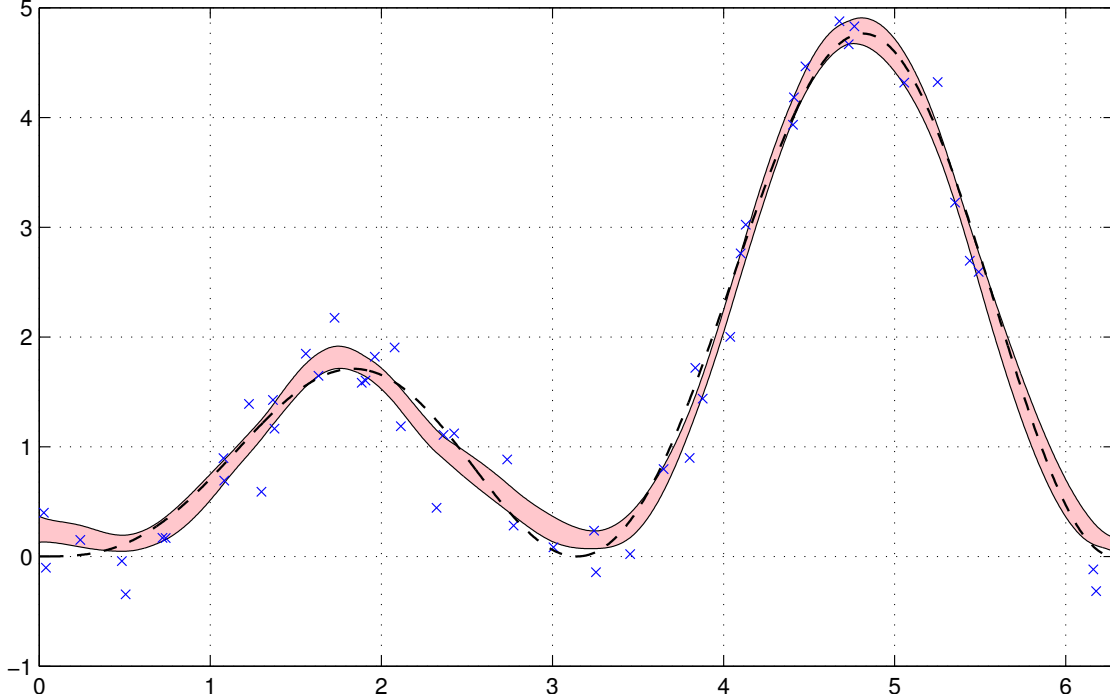
The likelihood is

$$p(\mathbf{Y} | \mathbf{X}, \mathbf{a}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \mathbf{a} \cdot \phi(x_i))^2\right), \quad (3.40)$$

and for the prior on  $\mathbf{a}$  we consider

$$p(\mathbf{a} | \lambda, \sigma^2) \propto \left(\frac{\lambda}{\sigma^2}\right)^{\frac{N+4}{2}} \exp\left(-\frac{\lambda}{2\sigma^2} \int_0^h dx (\mathbf{a} \cdot \phi''(x))^2\right), \quad (3.41)$$

$$\propto \left(\frac{\lambda}{\sigma^2}\right)^{\frac{N+4}{2}} \exp\left(-\frac{\lambda}{2\sigma^2} \mathbf{a}^T \mathbf{K} \mathbf{a}\right), \quad (3.42)$$



**Figure 6: Bayesian splines for positive functions.** The crosses show 50 samples from  $y_i = x_i \sin^2(x_i) + \varepsilon_i$ , where  $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$  with  $\sigma^2 = .09$ . The values of  $x_i$  were sampled uniformly from  $[0, 2\pi]$ . The curve  $f(x) = x \sin^2(x)$  is shown as a dashed line. The shaded band shows the splines built with coefficients from the .25 and .75 quantiles of samples from the posterior distribution of  $\mathbf{a}$  in (3.37). We used a Jeffreys prior for  $\sigma^2$  [Jeffreys, 1946] and imposed the positivity constraints (3.34) at 100 points spread uniformly in  $[0, 2\pi]$ . The smoothness parameter  $\lambda$  was estimated as  $\hat{\lambda} = 0.0067$  by maximizing the marginal likelihood (empirical Bayes criterion), using a Monte Carlo EM algorithm. The mean of the samples of  $\sigma^2$  was  $\hat{\sigma}^2 = 0.091$ . The spline computations were performed with the “fda” MATLAB package [Ramsay et al., 2009].

where  $\mathbf{K} \in \mathbb{R}^{N+4, N+4}$  has entries

$$K_{sr} = \int_0^h dx \phi_s''(x) \phi_r''(x). \quad (3.43)$$

The prior (3.41)-(3.42) is standard in the spline literature and imposes a  $\lambda$ -dependent penalty on the roughness of the estimated polynomial, with a bigger  $\lambda$  corresponding to a smoother solution. This penalty helps to avoid overfitting the data [Green and Silverman, 1994].

We can Gibbs sample from the posterior (3.37) by alternating between the conditional distributions of  $\sigma^2$  and  $\mathbf{a}$ . The latter is a TMG with

$$\log p(\mathbf{a} | \sigma^2, \mathbf{X}, \mathbf{Y}, \lambda) \propto -\frac{1}{2\sigma^2} \mathbf{a}^T (\mathbf{M} + \lambda \mathbf{K}) \mathbf{a} + \frac{1}{\sigma^2} \mathbf{a}^T \cdot \mathbf{r}, \quad s = 1 \dots N + 4, \quad (3.44)$$

constrained by (3.34), and we defined

$$\mathbf{M} = \sum_{i=1}^N \boldsymbol{\phi}(x_i) \boldsymbol{\phi}(x_i)^T \in \mathbb{R}^{N+4, N+4}, \quad (3.45)$$

$$\mathbf{r} = \sum_{i=1}^N y_i \boldsymbol{\phi}(x_i) \in \mathbb{R}^{N+4}. \quad (3.46)$$

In the B-spline basis, the matrices  $\mathbf{M}$  and  $\mathbf{K}$  in (3.44) have a banded form [De Boor, 2001]. As in Example 3.2 above, this allows us to speed up the runtime from  $O(N^2)$  to  $O(N)$  for each sample. Figure 6 shows an example for the function  $f(x) = x \sin^2(x)$ , with  $N = 50$  points sampled as

$$y_i = x_i \sin^2(x_i) + \varepsilon_i \quad \varepsilon_i \sim \mathcal{N}(0, \sigma) \quad \sigma^2 = .09, \quad (3.47)$$

and with the  $x_i$  sampled uniformly from  $[0, 2\pi]$ .

### 3.4 Bayesian reconstruction of quantized stationary Gaussian processes

Suppose that we are given  $N$  values of a function  $f(x_i)$ ,  $i = 1, \dots, N$ , that takes discrete values in a set  $\{q_k\}$ ,  $k = 1, \dots, K$ . We assume that this is a quantized projection of a sample  $y(x_i)$  from a stationary Gaussian process with a known translation-invariant covariance kernel of the form

$$(M^{-1})_{ij} = \Sigma_{ij} = K(|x_i - x_j|), \quad (3.48)$$

and the quantization follows a known rule of the form

$$f(x_i) = q_k \quad \text{if} \quad z_k \leq y(x_i) < z_{k+1}. \quad (3.49)$$

We are interested in sampling the posterior distribution

$$p(y(x_1), \dots, y(x_N) | f(x_1), \dots, f(x_N)), \quad (3.50)$$

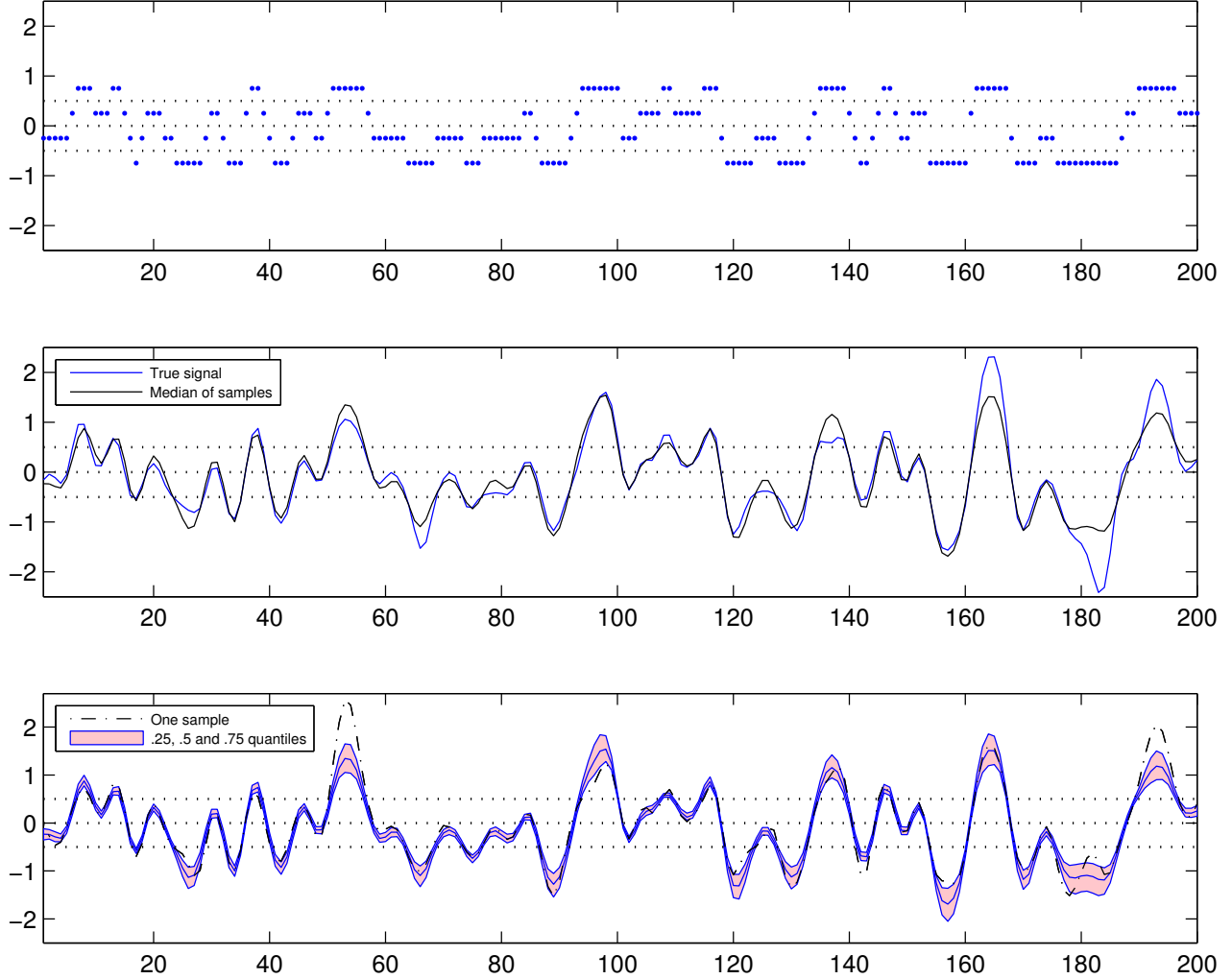
which is a Gaussian with covariance (3.48), truncated by the quantization rules (3.49). In this case, we can exploit the Toeplitz form of the  $N$ -by- $N$ -covariance matrix  $\Sigma_{ij}$  to reduce the runtime of the HMC sampler. The idea is to embed  $\Sigma$  into a circulant matrix as (see e.g., [Chu and George, 1999])

$$C = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \cdot & \cdot & \Sigma_{1(N-1)} & \Sigma_{1N} & \Sigma_{1(N-1)} & \cdot & \cdot & \Sigma_{12} \\ \Sigma_{12} & \Sigma_{11} & \cdot & \cdot & \Sigma_{1(N-2)} & \Sigma_{1(N-1)} & \Sigma_{1N} & \cdot & \cdot & \Sigma_{13} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \in \mathbb{R}^{(2N-2) \times (2N-2)} \quad (3.51)$$

As is well known, a circulant matrix is diagonalized by the discrete Fourier transform matrix  $O$  as

$$C = O^T \Lambda O, \quad (3.52)$$

where  $O^T O = 1$  and  $\Lambda$  is the diagonal matrix of the eigenvalues. We can exploit this structure in the frame in which the Hamiltonian is given by (2.31). Recall that at each iteration we must sample



**Figure 7: Bayesian reconstruction of quantized functions.** *Upper panel: a function quantized at  $N = 200$  points, taking four possible values. Middle panel: true function and median of the posterior samples. Lower panel: one posterior sample and median and .25 and .75 quantiles of the posterior samples. The three dotted lines, at  $y = 0, \pm 0.5$ , separate the four quantization regions. We used 5,000 samples, after discarding 500 as burn-in. The  $N = 200$  values of the true function were sampled from a Gaussian process with kernel given by (3.54) with  $\sigma^2 = 0.6, \eta^2 = 0.2$ . Using the translation invariance of this kernel, the covariance matrix of the Gaussian can be embedded into a circulant matrix, and this allows to sample the initial velocity at every iteration of the HMC sampler in  $O(N \log N)$  time instead of  $O(N^2)$ .*

the initial velocity  $\dot{X}(0)$  from an  $N$ -dimensional Gaussian with covariance  $\Sigma$ . We can do this by starting with a  $(2N - 2)$ -dimensional sample  $\varepsilon$  from a unit-covariance Gaussian and computing

$$O^T \Lambda^{1/2} \varepsilon. \quad (3.53)$$

The first  $N$  elements of this vector are samples from a Gaussian with covariance  $\Sigma$  as desired. The multiplication by  $O^T$  in (3.53) can be performed with a Fast Fourier Transform, and this reduces the runtime for each  $\tilde{X}(0)$  sample from  $O(N^2)$  to  $O(N \log N)$ . Figure 7 shows an example of a quantized and reconstructed function evaluated at  $N = 200$  points. The function was sampled from a Gaussian process with kernel

$$K(|x_i - x_j|) = \sigma^2 e^{-\frac{|x_i - x_j|^2}{2\eta^2}} \quad (3.54)$$

with  $\sigma^2 = 0.6$ ,  $\eta^2 = 0.2$ , and the quantization was performed by splitting the target space into four regions separated by  $y = 0, \pm 5$ .

This technique can be easily extended to functions defined in higher dimensional spaces, such as images in 2D, where the quantization acts as a lossy compression technique and the estimated values  $E(y(x_i)|f(x_1), \dots, f(x_N))$  can be thought of as the decoded (decompressed) image.

## 4 The Bayesian Lasso

The techniques introduced above can also be used to sample from multivariate distributions whose log density is piecewise quadratic, with linear or elliptical boundaries between the piecewise regions. Instead of presenting the most general case, let us elaborate the details for the example of the Bayesian Lasso [Park and Casella, 2008, Hans, 2009, Polson and Scott, 2011].

We are interested in the posterior distribution of the coefficients  $\boldsymbol{\beta} \in \mathbb{R}^d$  and  $\sigma^2$  of a linear regression model

$$y_n = \mathbf{z}_n \cdot \boldsymbol{\beta} + \varepsilon_n \quad \varepsilon_n \sim \mathcal{N}(0, \sigma^2) \quad n = 1, \dots, N, \quad (4.1)$$

Defining

$$\mathbf{Y} = (y_1, \dots, y_N) \quad (4.2)$$

$$\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_N), \quad (4.3)$$

we want to sample from the posterior distribution

$$p(\boldsymbol{\beta}, \sigma^2 | \mathbf{Y}, \mathbf{Z}, \lambda) \propto p(\mathbf{Y} | \mathbf{Z}, \boldsymbol{\beta}, \sigma^2) p(\boldsymbol{\beta} | \lambda, \sigma^2) p(\sigma^2), \quad (4.4)$$

with prior density for the coefficients

$$p(\boldsymbol{\beta} | \lambda, \sigma^2) = \left( \frac{\lambda}{2\sigma^2} \right)^d \exp \left( -\frac{\lambda}{\sigma^2} \sum_{i=1}^d |\beta_i| \right). \quad (4.5)$$

This prior is called the *lasso* (for ‘least absolute shrinkage and selection operator’) and imposes a  $\lambda$ -dependent sparsening penalty in the maximum likelihood solutions for  $\boldsymbol{\beta}$  [Tibshirani, 1996].

We can Gibbs sample from the posterior (4.4) by alternating between the conditional distributions of  $\sigma^2$  and  $\boldsymbol{\beta}$ . The latter is given by

$$-\log p(\boldsymbol{\beta} | \mathbf{Y}, \mathbf{Z}, \sigma^2, \lambda) = \frac{1}{2\sigma^2} \sum_{n=1}^N (\mathbf{z}_n \cdot \boldsymbol{\beta} - y_n)^2 + \frac{\lambda}{\sigma^2} \sum_{i=1}^d |\beta_i| + \text{const.} \quad (4.6)$$

$$= \frac{1}{2\sigma^2} \boldsymbol{\beta}^T \mathbf{M} \boldsymbol{\beta} - \frac{1}{\sigma^2} \sum_{i=1}^d L^i(s_i) \beta_i + \text{const.} \quad (4.7)$$

where we defined

$$\mathbf{M} = \sum_{n=1}^N \mathbf{z}_n \mathbf{z}_n^T \in \mathbb{R}^{d \times d} \quad (4.8)$$

$$L^i(s_i) = \sum_{n=1}^N y_n(z_i)_n - \lambda s_i \quad i = 1, \dots, d. \quad (4.9)$$

with

$$s_i = \text{sign}(\beta_i). \quad (4.10)$$

Sampling  $\boldsymbol{\beta}$  from (4.7) was considered previously via Gibbs sampling, either expressing the Laplace prior (4.5) as mixtures of Gaussians [Park and Casella, 2008] or Bartlett-Fejer kernels [Polson and Scott, 2011], or directly from (4.7) [Hans, 2009].

In order to apply Hamiltonian Monte Carlo we consider the Hamiltonian

$$H = \frac{1}{2\sigma^2} \boldsymbol{\beta}^T \mathbf{M} \boldsymbol{\beta} - \frac{1}{\sigma^2} \sum_{i=1}^d L^i(s_i) \beta_i + \frac{\sigma^2}{2} \mathbf{S}^T \mathbf{M}^{-1} \mathbf{S}. \quad (4.11)$$

Note that we did not map the coordinates to a canonical frame, as in Section 2.1. Instead, we chose a momenta mass matrix  $\sigma^{-2} \mathbf{M}$ , which is equal to the precision matrix of the coordinates. This choice leads to the simple equations

$$\ddot{\beta}_i = -\beta_i + \mu_i(\mathbf{s}), \quad (4.12)$$

where

$$\mu_i(\mathbf{s}) = \sum_{j=1}^d M_{ij}^{-1} L^j(s_j). \quad (4.13)$$

The solution to (4.12) is

$$\beta_i(t) = \mu_i(\mathbf{s}) + a_i \sin(t) + b_i \cos(t), \quad (4.14)$$

$$= \mu_i(\mathbf{s}) + A_i \cos(t + \varphi_i), \quad (4.15)$$

where

$$A_i = \sqrt{a_i^2 + b_i^2}, \quad (4.16)$$

$$\tan \varphi_i = -\frac{a_i}{b_i}. \quad (4.17)$$

The constants  $a_i, b_i$  in (4.14) can be expressed in terms of the initial conditions as

$$b_i = \beta_i(0) - \mu_i(\mathbf{s}) \quad (4.18)$$

$$a_i = \dot{\beta}_i(0) \quad (4.19)$$

$$= M_{ij}^{-1} p^j(0). \quad (4.20)$$

As in Section 2.1, we start by sampling  $\mathbf{S}$  from  $p(\mathbf{S}) = \mathcal{N}(0, \sigma^{-2}\mathbf{M})$  and let the particle move during a time  $T = \pi/2$ . The trajectory of the particle is given by (4.15) until a coordinate crosses any of the  $\beta_i = 0$  planes, which happens at the smallest time  $t > 0$  such that

$$0 = \mu_i(\mathbf{s}) + A_i \cos(t + \varphi_i), \quad i = 1, \dots, d. \quad (4.21)$$

(Note that had we transformed the coordinates  $\beta$  to a canonical frame, each condition here would have involved a sum of  $d$  terms; thus the parameterization we use here leads to sparser, and therefore faster, computations.) Suppose the constraint is met for  $i = j$  at time  $t = t_j$ . At this point  $\beta_j$  changes sign, so the Hamiltonian (4.11) changes by replacing

$$L^j(s_j) \longrightarrow L^j(-s_j) = L^j(s_j) + 2s_j\lambda, \quad (4.22)$$

which in turn changes the values of  $\mu_i(\mathbf{s})$ 's in (4.13). Note from (4.12) that this causes a jump in  $\dot{\beta}(t_j)$ . Using the continuity of  $\beta(t_j)$ ,  $\dot{\beta}(t_j)$  and the updated  $\mu_i(\mathbf{s})$ 's, we can compute new values for  $a_i$  and  $b_i$  as in (4.18) and (4.19) to continue the trajectory for times  $t > t_j$ .

We have found the efficiency of this algorithm comparable to other methods to sample from the Bayesian Lasso model, e.g., [Park and Casella, 2008]. The real advantage of our approach would be when the coefficients  $\beta_i$ 's have additional constraints, as in the tree shrinkage model [LeBlanc and Tibshirani, 1998], the hierarchical Lasso [Bien et al., 2012], or when some of the coefficients are constrained to be positive. In these cases, it is very easy to combine this algorithm with the imposition of constraints of the previous Section.

Finally, the piecewise linear log-density (4.6) is continuous with discontinuous derivative, but we can also consider discontinuous log-densities defined piecewise. In these cases, the velocity is not continuous across the boundary of two regions, but jumps in such a way that the total energy is conserved. The extension of the basic method to this case is straightforward.

## Acknowledgements

This work was supported by an NSF CAREER grant, a McKnight Scholar award, NSF grant IIS-0904353 and by the Defense Advanced Research Projects Agency (DARPA) MTO under the auspices of Dr. Jack Judy, through the Space and Naval Warfare Systems Center, Pacific Grant/Contract No. N66001-11-1-4205. This material is based upon work supported by, or in part by, the U. S. Army Research Laboratory and the U. S. Army Research Office under contract number W911NF-12-1-0594. AP is supported by the Swartz Foundation. We thank Matt Hoffman, Alexandro Ramirez, Carl Smith and Eftychios Pnevmatikakis for helpful discussions.

## References

- J.H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association*, pages 669–679, 1993.
- JR Ashford and RR Sowden. Multi-variate probit analysis. *Biometrics*, pages 535–546, 1970.
- M. N. Bannerman, R. Sargant, and L. Lue. Dynamo: a free  $o(n)$  general event-driven molecular dynamics simulator. *Journal of Computational Chemistry*, 32(15):3329–3338, 2011.
- J. Bien, J. Taylor, and R. Tibshirani. A Lasso for Hierarchical Interactions. *Arxiv preprint arXiv:1205.5050*, 2012.
- M.H. Chen and J. Deely. Application of a new Gibbs Hit-and-Run sampler to a constrained linear multiple regression problem. Technical report, Technical Report 92-21, Purdue University, Center for Statistical Decision Sciences and Department of Statistics, 1992.
- E. Chu and A. George. *Inside the FFT black box: serial and parallel fast Fourier transform algorithms*. CRC, 1999.
- DR Cox and N. Wermuth. On some models for multivariate binary variables parallel in complexity with the multivariate Gaussian distribution. *Biometrika*, 89(2):462–469, 2002.
- P. Damien and S.G. Walker. Sampling truncated normal, beta, and gamma densities. *Journal of Computational and Graphical Statistics*, 10(2):206–215, 2001.
- C. De Boor. *A practical guide to splines*. Springer Verlag, 2001.
- S. Duane, A.D. Kennedy, B.J. Pendleton, and D. Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.
- N. Ellis and R. Maitra. Multivariate Gaussian simulation outside arbitrary ellipsoids. *Journal of Computational and Graphical Statistics*, 16(3):692–708, 2007.
- Lawrence J Emrich and Marion R Piedmonte. A method for generating high-dimensional multivariate binary variates. *The American Statistician*, 45(4):302–304, 1991.
- Alan E Gelfand, Adrian FM Smith, and Tai-Ming Lee. Bayesian analysis of constrained parameter and truncated data problems using gibbs sampling. *Journal of the American Statistical Association*, 87(418):523–532, 1992.
- A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin. *Bayesian data analysis*. CRC press, 2004.
- J. Geweke. Efficient simulation from the multivariate normal and student-t distributions subject to linear constraints and the evaluation of constraint probabilities. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, pages 571–578, 1991.
- J. Geweke, M. Keane, and D. Runkle. Alternative computational approaches to inference in the multinomial probit model. *The review of economics and statistics*, pages 609–632, 1994.



- P. Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer, 2003.
- P.J. Green and B.W. Silverman. *Nonparametric regression and generalized linear models: a roughness penalty approach*, volume 58. Chapman & Hall/CRC, 1994.
- JM Hammersley and KW Morton. A new Monte Carlo technique: antithetic variates. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 52, pages 449–475. Cambridge Univ Press, 1956.
- C. Hans. Bayesian lasso regression. *Biometrika*, 96(4):835–845, 2009.
- D. Herbison-Evans. Solving quartics and cubics for graphics. Technical report, Technical Report TR-94-487, Basser Department of Computer Science, University of Sidney, Sidney, Australia, 1994.
- M.D. Hoffman and A. Gelman. The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Arxiv preprint arXiv:1111.4246*, 2011.
- J.A. Izaguirre and S.S. Hampton. Shadow hybrid Monte Carlo: an efficient propagator in phase space of macromolecules. *Journal of Computational Physics*, 200(2):581–604, 2004.
- Harold Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 186(1007):pp. 453–461, 1946. ISSN 00804630. URL <http://www.jstor.org/stable/97883>.
- AD Kennedy. The theory of hybrid stochastic algorithms. In *NATO ASIB Proc. 224: Probabilistic Methods in Quantum Field Theory and Quantum Gravity*, volume 1, page 209, 1990.
- AD Kennedy and KM Bitar. An exact Local Hybrid Monte Carlo algorithm for gauge theories. *Nuclear Physics B-Proceedings Supplements*, 34:786–788, 1994.
- J.H. Kotecha and P.M. Djuric. Gibbs sampling approach for generation of truncated multivariate gaussian random variables. In *Proceedings., 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1757–1760. IEEE, 1999.
- M. LeBlanc and R. Tibshirani. Monotone shrinkage of trees. *Journal of Computational and Graphical Statistics*, pages 417–433, 1998.
- J.S. Liu. *Monte Carlo strategies in scientific computing*. Springer, 2008.
- R.M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 54: 113–162, 2010.
- B. Neelon and D.B. Dunson. Bayesian isotonic regression and trend analysis. *Biometrics*, 60(2): 398–406, 2004.
- L. Paninski, J.W. Pillow, and E.P. Simoncelli. Maximum likelihood estimation of a stochastic integrate-and-fire neural encoding model. *Neural computation*, 16(12):2533–2561, 2004.

- T. Park and G. Casella. The Bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.
- N.G. Polson and J.G. Scott. The Bayesian Bridge. *Arxiv preprint arXiv:1109.2279*, 2011.
- J.O. Ramsay, G. Hooker, and S. Graves. *Functional data analysis with R and MATLAB*. Springer Verlag, 2009.
- C.E. Rasmussen. Gaussian processes to speed up Hybrid Monte Carlo for expensive Bayesian integrals. In *Bayesian Statistics 7: Proceedings of the 7th Valencia International Meeting*, pages 651–659. Oxford University Press, 2003.
- C.P. Robert. Simulation of truncated normal variables. *Statistics and computing*, 5(2):121–125, 1995.
- C.P. Robert and G. Casella. *Monte Carlo statistical methods*. Springer Verlag, 2004.
- G. Rodriguez-Yam, R.A. Davis, and L.L. Scharf. Efficient Gibbs sampling of truncated multivariate normal with application to constrained linear regression. *Unpublished Manuscript*, 2004. <http://www.stat.columbia.edu/~rdavis/papers/CLR.pdf>.
- R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58:267–288, 1996.
- J. Tobin. Estimation of relationships for limited dependent variables. *Econometrica: Journal of the Econometric Society*, 26(1):24–36, 1958.