

One-Shot CFTP; application to a class of Truncated Gaussian Densities

A.Beskos and G.O.Roberts*

July 5, 2006

Abstract

In this paper, we introduce a new method for perfect simulation of multivariate densities. We use One-Shot CFTP (Roberts and Rosenthal, 2002) together with a monotone coupler for the Gibbs sampler, and implement the algorithm within the Read-Once CFTP protocol (Wilson, 2000b). We illustrate our method by simulating efficiently from high-dimensional truncated normal distributions using the Gibbs sampler.

Keywords : One-Shot coupling; Read-Once coupling from the past; truncated gaussian densities.

AMS Subject Classification: Primary: 65C40, Secondary 65C60.

1 Introduction

Although Perfect Simulation (Propp and Wilson, 1996) is now well-known and has been used in a variety of MCMC contexts, it has proved difficult to apply effectively in high-dimensional situations without specific symmetry to guide the algorithm construction. Thus in particular MCMC problems from Bayesian statistics are difficult for Perfect Simulation (see Murdoch and Green, 1998, for a first attempt). Furthermore, continuous state space problems are intrinsically difficult for any technique based on coupling, since intricate constructions are needed to ensure coupling at all, let alone allow the construction of efficient coalescence rules for complex Markov chains.

In this paper we shall bring together and extend a collection of generic techniques for explicit coupling of continuous state space Markov chains. We will show how these methods can be applied effectively for high-dimensional Gibbs samplers and related algorithms. We illustrate our methods through an example which will be explored in detail through the paper and involves a gaussian density restricted to an arbitrary rectangular region. However we hope to illustrate that our methods have great potential in more general situations.

Our example will achieve exact sampling from multivariate gaussian densities restricted to rectangular regions of the form:

$$\pi(\mathbf{x}) \propto \exp \left\{ -\frac{1}{2}(\mathbf{x} - \mu)^\top Q (\mathbf{x} - \mu) \right\} \cdot \mathbb{I}_S(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d \quad (1)$$

for $\mu \in \mathbb{R}^d$, $Q = (q_{ij}) \in \mathbb{R}^{d \times d}$ a positive definite, symmetric matrix and $S = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{a} \preceq \mathbf{x} \preceq \mathbf{b}\}$, $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$. (\preceq) is the natural partial ordering on \mathbb{R}^d , \mathbb{I}_S the indicator function of S , \mathbb{R} the

*Department of Mathematics and Statistics, Lancaster University, U.K., email: a.beskos@lancaster.ac.uk, g.o.roberts@lancaster.ac.uk

extended real line $\mathbb{R} \cup \{-\infty, \infty\}$ and \top denotes matrix transpose. We call Q the precision matrix. Considering location-scale re-parameterisation we can assume that $\mu \equiv \mathbf{0}$ and $q_{ii} \equiv 1$, $1 \leq i \leq d$.

The abbreviation TN will refer to an arbitrary member of the family of distributions with densities of the type (1). When a random vector \mathbf{X} is distributed according to π we write $\mathbf{X} \sim TN_d(\mu, Q^{-1})|_a^b$. Whenever some vector $\mathbf{u} \in \mathbb{R}^d$ is mentioned u_i denotes its i -th element and \mathbf{u}_{-i} the complete vector without its i -th element, $1 \leq i \leq d$. We use bold fonts for vectors preserving the right to drop this convention whenever needed.

Our methodology deals most easily with cases when the precision matrix is *Stieltjes* (Varga, 1962, ch.3):

Definition 1 *A matrix $V = (v_{ij}) \in \mathbb{R}^{d \times d}$ is a Stieltjes matrix if it is symmetric, positive definite with $v_{ij} \leq 0$ for all $i \neq j$.*

A Stieltjes precision matrix induces a stochastically monotone (w.r.t. (\preceq)) Gibbs sampler on π and this is precisely the property that makes possible the development of the Perfect Simulation algorithms that follow.

Econometrics (Geweke, 1991; Hajivassiliou and McFadden, 1998) and signal processing (Djuric and Kotecha, 1999; Rodriguez-Yam et al., 2002; Troughton and Godsill, 1999) literature provides several cases when TN s arise and different methods for sampling from them. The univariate case is easy to deal with using appropriate rejection sampling methods, see for example Geweke (1991), Robert (1995). Rejection sampling algorithms in multivariate cases involve comparison functions like the corresponding untruncated density, the product of univariate exponential (Hajivassiliou et al., 1996) or Gaussian densities restricted to the support of the target TN and sequentially generated densities (Hajivassiliou and McFadden, 1998). All these techniques perform poorly in high dimensions or for constraint regions away from the bulk of the underlying untruncated density. Many have recommended the use of the Gibbs sampler (Djuric and Kotecha, 1999; Geweke, 1991; Hajivassiliou and McFadden, 1998; Robert, 1995; Rodriguez-Yam et al., 2002) or some other MCMC technique (Troughton and Godsill, 1999). These methods return correlated draws from a distribution that approximates (in a suitable sense) the actual target distribution. Recently, a Perfect Simulation method for TN s restricted to $[0, \infty)^d$ has been presented (Philippe and Robert, 2003) which exploits a monotonicity property induced by the MCMC technique of Slice Sampler. Its algebraic difficulties render it manageable only for $d \leq 3$. It is also not exact in the sense that a vector with sufficiently big values is used instead of $\{+\infty\}^d$.

Our methodology, as applied to a TN , lies close to the Gibbs sampler approach and could be thought of as a way of turning the approximate results of the Gibbs method into exact ones. The Gibbs sampler when applied to a TN exploits the fact that the univariate full conditional distributions are still TN s. Indeed:

$$\text{if } \mathbf{X} \sim \pi \text{ then } X_i | \mathbf{X}_{-i} = \mathbf{x}_{-i} \sim TN_1(\mu_i(\mathbf{x}_{-i}), 1)|_{a_i}^{b_i}, \text{ for } \mu_i(\mathbf{x}_{-i}) = - \sum_{j:j \neq i}^d q_{ij} x_j \quad (2)$$

The algorithms to be presented follow the Read-Once coupling from the past (Ro-CFTP) protocol (Wilson, 2000b), so we begin by presenting in Section 2 a version of the Ro-CFTP idea called *One-Shot CFTP* and introduced in Roberts and Rosenthal (2002). In Section 3 we introduce the coupling technique of the *Monotone Independence Coupler (MIC)* necessary for the development of the Perfect Simulation algorithms that follow.

Two algorithms will be presented. The *case-specific* (Section 4) one is the first that comes in mind under the certain structure of the precision matrix and relies on the cdf Φ of the $\mathcal{N}(0, 1)$ distribution ($\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution with mean μ and variance σ^2) and its inverse Φ^{-1} . Values of $\Phi(x)$ and $\Phi^{-1}(u)$ for big $|x|$ and $u \downarrow 0$, $u \uparrow 1$ must be treated with caution

because of the finite accuracy of computing systems and this sometimes restricts the efficiency of the algorithm.

The *pure* algorithm (Section 5) does not use at any point Φ or its quantiles. It is more complicated than the case-specific one but more general. Although strictly speaking it is not needed for the truncated gaussian example (apart from the numerical issues mentioned above which nonetheless can sometimes be critical for the applicability of the algorithm, see the last example in Subsection 7.2) we will describe it in detail and implement it with a view to demonstrating the general applicability of our methodology. Both algorithms exploit the Stieltjes property of the precision matrix; the case-specific algorithm by constructing a monotone Gibbs sampler and the pure algorithm by devising convergent upper and lower dominating processes for the Gibbs sampler Markov chain.

In Section 6 we show that the applicability of the above algorithms can be easily extended to a class of truncated gaussian densities wider than the one with Stieltjes precision matrix and that similar algorithms can be devised even for non-gaussian densities. In Section 7 we run the algorithms and present several results concerning their correctness and efficiency. It turns out that they draw efficiently from the target distribution even for high dimensions. We finish with some conclusions in Section 8.

2 One-Shot CFTP

The One-Shot coupling has been introduced in Roberts and Rosenthal (2002) in the context of obtaining computable bounds for the convergence of Markov chains. Our approach will be to use the method as an ingredient in a Ro-CFTP algorithm. The construction proceeds as follows. Suppose we have two Markov chains $(X_n)_0^\infty, (Y_n)_0^\infty$, say on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$, governed by the same transition probability kernel but with different initial distributions which are *coupled* in a way that $|X_n - Y_n| \rightarrow 0$ as $n \rightarrow \infty$ (note that we drop the bold font representation of vectors for this section). An attempt to unite the two paths is made when their distance is small and this occasionally leads to stringent bounds on the total variation distance $\sup_{A \in \mathcal{B}(\mathbb{R}^d)} |\mathbb{P}(X_n \in A) - \mathbb{P}(Y_n \in A)|$. We adjust the One-Shot coupling idea within the context of the Ro-CFTP algorithm.

In the sequel, we adopt the Stochastic Recursive Sequence representation of a Markov chain and use *random update functions* as means of describing the dynamics of the chain and, within a simulation context, constructing sample paths of the chain for different starting points using same random elements (what is called *flows* in the Perfect Simulation literature). Assume that we need to sample from a (possibly unnormalised) density π on \mathbb{R}^d w.r.t. some reference measure. For convenience we denote by π both this density and the probability measure induced by it. Let S be the support of π .

The CFTP protocol (Propp and Wilson, 1996) involves the development of an ergodic (usually discrete-time) Markov chain with stationary distribution π and the parallel realisation of paths of the chain starting from everywhere in the support of π in a way that it is possible all paths to merge into a single trajectory in finite time. The paths are run forward with starting points taken increasingly back in time until a random moment $T < 0$ is found when they all get projected onto a single point at time 0. This single output is proved to be distributed according to π . During this repeated excursion into the past random seeds used for the realisation of the paths are stored and recalled any time paths coming from deeper into the past meet the time instances where these seeds were first used.

Unlike the basic CFTP idea, the Ro-CFTP technique does not require the storage of random seeds since it evolves forward in time. It relies on the development of a number, say k , of random update functions $\phi_i, 1 \leq i \leq k$, and their synthesis to a compound random update function which

in our case will have the structure:

$$\Phi(x, \overline{U}_k) := \phi_k(\cdots(\phi_2(\phi_1(x, \overline{U}_1), \overline{U}_2), \dots, \overline{U}_k), \quad x \in S$$

or in an abbreviated form:

$$\Phi(x) = \phi_k \circ \cdots \circ \phi_2 \circ \phi_1(x), \quad x \in S \quad (3)$$

We have denoted by U_i the random elements needed for the first time at the development of ϕ_i . Also, $\overline{U}_i := (U_1, U_2, \dots, U_i)$, $1 \leq i \leq k$. The update rule Φ must *preserve* π :

$$\text{if } \mathcal{P}(x, dy) := P(\Phi(x) \in dy), \quad x \in S \quad \text{then} \quad \int_S \pi(dx) \mathcal{P}(x, dy) = \pi(dy) \quad (4)$$

or in a less troublesome notation $\pi\mathcal{P} = \pi$. We call the synthesis (3) a *block*. In other words, $\phi_1, \phi_2, \dots, \phi_k$ combined produce the update rule Φ that will generate a Markov chain with invariant measure π . The mapping $x \mapsto \Phi(x)$ evolves paths of this chain starting from anywhere in the state space S to the end of the block.

It is critical that we are able to locate the (random) image $\Phi(S) = \{\Phi(x); x \in S\}$ and construct the block in a way that it can project all paths onto a single point. To this end, most of the ϕ_i are built in a way that yields a monotonicity (e.g. $\phi_i(x) \preceq \phi_i(y)$ when $x \preceq y$) or a coalescence property. The monotonicity property could provide upper and lower paths that make possible the identification of a superset of $\Phi(S)$ after carrying out the updates ϕ_i only for these sandwiching paths while the coalescence property makes possible the event of a block being coalescent, i.e. $\Phi(S)$ is a singleton.

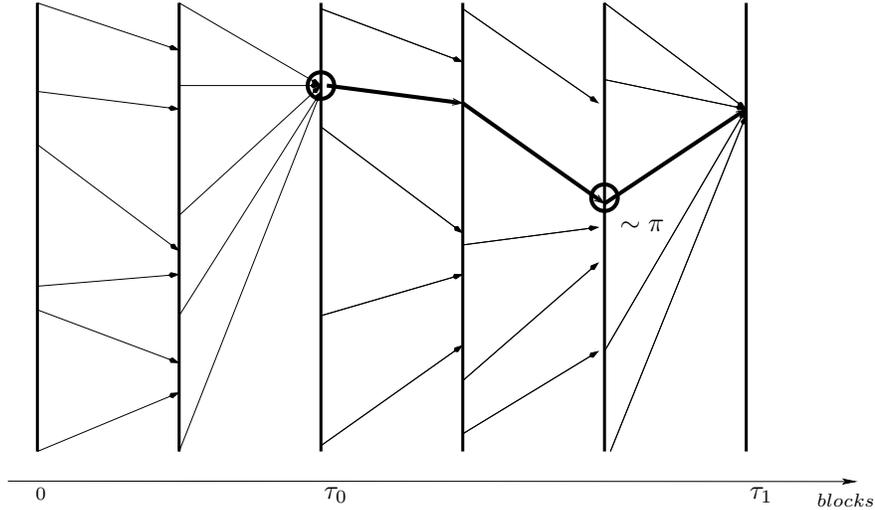


Figure 1: A realisation of successive independent Φ -blocks. The thick trajectory represents the coalescence path initiated with the singleton outputted by the first successful block. The instances of this path one block-step before every successful block are i.i.d. observations from π .

The Ro-CFTP protocol involves the repeated, forward realisation of a number of independent Φ -blocks and the consideration of their image $\Phi(S)$. In most cases $\Phi(S)$ is difficult to determine and, subsequently, it is impossible to decide whether the coalescence event $\{\Phi(S) \text{ is a singleton}\}$ has taken place or not after running a block. Instead, it can be easier to determine whether a

sub-event, say D , of the actual coalescence event has occurred or not. For the Ro-CFTP method to be possible it is then necessary that:

$$P(D) =: \rho > 0 \tag{5}$$

Assume that Φ^i is the i -th iteration of a block. Let \mathbf{w} be the observation achieved at the first instance of a *successful* block, i.e. a block when D occurred, say at the τ_0 -th iteration ($\mathbf{w} \equiv \Phi^{\tau_0}(S)$). We call the path initiated by the singleton \mathbf{w} of the first successful block and constructed by its updates thereafter the *coalescence path*. Define:

$$\tau_i = \min \{ j > \tau_{i-1} : \Phi^j \text{ is successful} \}, \quad i \geq 1$$

Theorem 1 *The $\Phi^{\tau_i-1} \circ \dots \circ \Phi^{\tau_0+1} \circ \Phi^{\tau_0}(S)$, $i \geq 1$, are independent draws from π .*

Proof: In the Appendix.

In other words, the instances of the coalescence path one block-step before every successful block are i.i.d. observations from π , see Fig.1. Thus, for each block we need to know just the location of the coalescence path and the status of the block (successful or not).

An intuitive proof of the Ro-CFTP protocol is given in Wilson (2000b) and an algebraic one in Breyer and Roberts (2001). Our presentation of the Ro-CFTP algorithm is a little more general than in the first appearance of the algorithm (Wilson, 2000b) since we shall need to be able to monitor some arbitrary sub-event of the actual coalescence event for determining whether a block is successful or not. To demonstrate the validity of this compromise we have included in the Appendix a sketch of the proof of Theorem 1 in line with the one given in Breyer and Roberts (2001) except for some small changes to apply to our case.

A common situation when we resort to a sub-event of $\{\Phi(S) \text{ is a singleton}\}$ is when we can only locate a set $\Phi' \supseteq \Phi(S)$. In this case $D \equiv \{\Phi' \text{ is a singleton}\}$. Provided that marginally the updates ϕ_i are Markovian (as it will be the case for the algorithms to be presented), if we can determine that a block will be unsuccessful before running all the steps within that block it is then not necessary to continue keeping track of all the paths started from S ; it suffices to update only the coalescence path with respect to the distributions $P_i(x, dy) := P(\phi_i(x) \in dy)$ for the single current location x of the coalescence path from the time instance when failure was inspected to the end of the block. This follows from the simple consideration that the coalescence path marginally is updated according to the P_i 's anyway and that for an unsuccessful block the Ro-CFTP algorithm needs only the update of the coalescence path.

In summary, to sample from π we run the blocks and keep track of the coalescence path. The instances of this path one block before each successful one constitute i.i.d. draws from π . The algorithm runs in finite time (in terms of the iterations of the blocks needed) since from (5) it is clear that the number of blocks necessary to get n draws from π is distributed according to a negative binomial distribution with mean $(n + 1)/\rho$. The term One-Shot CFTP refers to the specific structure we impose on the block; we attempt coalescence only once at the last step of the block. In the case of TNs , an appropriate Independence sampler step at the beginning of the block forces paths starting from anywhere in the state space into a bounded set while a number of Gibbs sampler steps bring the paths into a small, identifiable hyper-rectangle ensuring that the final coalescence step has big probability of being successful. The construction of a block of the One-Shot algorithm on TNs will be presented in detail in the sections that follow. Fig.3 at pg.10 shows the phases in the formation of such a block. The meaning of the symbols on the graph is explained in the sequel.

3 The Monotone Independence Coupler

The Monotone Independence Coupler (*MIC*) exploits the MCMC technique of Independence Sampling to turn stochastic domination into almost surely domination and raise at the same time a coalescence event of positive probability. It is presented in Theorem 2. Its relevance to the Gibbs sampler and *TN*s is explained in Subsection 3.1. The *Independence Coupler* has been proposed as a coupling method in Breyer and Roberts (2000); the additional adjective emphasises the monotonicity property induced by a certain version of this coupling technique that we devised.

We begin by presenting the Independence Coupler. Let $P(x, dy)$ be a probability kernel on $X \times \mathcal{E}$ which for any x in some set X defines a probability measure on a sufficiently regular measurable space (E, \mathcal{E}) . Assume that $p(x, y)$ is the density of $P(x, dy)$ w.r.t. some reference measure on (E, \mathcal{E}) and q some other density w.r.t. the same measure. Let $x \mapsto F_x, x \in X$, be a random function that uses random seeds defined on the probability triple $(\Omega, \mathcal{F}, \mathbb{P})$ such that marginally $\mathbb{P}(F_x \in dy) = P(x, dy)$. Draw $Y \sim q$ and $U \sim \mathcal{U}_{[0,1]}$ independently ($\mathcal{U}_{[\alpha, \beta]}$ is the uniform distribution on $[\alpha, \beta]$). Consider the modification:

$$C \circ F_x = \begin{cases} Y & \text{if } p(x, F_x) q(Y) U \leq p(x, Y) q(F_x) \\ F_x & \text{otherwise} \end{cases}, \quad x \in X$$

Then for any $x \in X$ it is true that $\mathbb{P}(C \circ F_x \in dy) = P(x, dy)$. The proof is given in Breyer and Roberts (2000). The intuition is that $C \circ F_x$ is derived after drawing F_x from $p(x, \cdot)$ and then performing an Independence sampler step with $p(x, \cdot)$ as the stationary density.

The *MIC* is a special case of the Independence Coupler. It exploits a property sufficient to induce stochastic ordering. We denote by $\mathcal{B}(\mathbb{R})$ the Borel σ -algebra on \mathbb{R} .

Proposition 1 *Let $P(x, dy)$ be a probability kernel on $X \times \mathcal{B}(\mathbb{R})$, $X \subseteq \mathbb{R}$, and $p(x, y)$ its density w.r.t. some reference measure. Assume that p has the properties:*

- (i) *all $p(x, \cdot), x \in X$, have common support $S \subseteq \mathbb{R}$.*
- (ii) *$p(x_2, y)/p(x_1, y)$ is increasing in $y \in S$ when $x_1 \leq x_2$.*

Then P is monotone, that is if D_x is the distribution function that corresponds to $P(x, dy), x \in X$, then for $x_1 \leq x_2, D_{x_1} \geq D_{x_2}$.

Proof:

Let $x_1 \leq x_2$ be elements of S . Consider the set $A = \{y \in \mathbb{R} : p(x_1, y) > p(x_2, y)\}$. In case that $A = \emptyset$ and there exists t such that $D_{x_1}(t) < D_{x_2}(t)$ then $D_{x_1}(\infty) < D_{x_2}(\infty)$ which cannot be true. In case that $A \neq \emptyset$ let $\delta = \sup A$. If $\delta = \infty$, then $p(x_1, y) > p(x_2, y)$ for all $y \in S$, so $D_{x_1} \geq D_{x_2}$. Else if $\delta < \infty$ and there exists $t \geq \delta$ such that $D_{x_1}(t) < D_{x_2}(t)$ then $D_{x_1}(\infty) < D_{x_2}(\infty)$.

□

We can now present the Monotone Independence Coupler. It operates on monotone probability kernels to induce coalescence.

Theorem 2 *Consider a probability kernel $P(x, dy)$ on $X \times \mathcal{B}(\mathbb{R})$, $X \subseteq \mathbb{R}$, with density $p(x, y)$ that satisfies properties (i) and (ii) of Proposition 1. Assume that F_x is a random map such that marginally $F_x \sim P(x, dy)$ for all $x \in X$ and $F_{x_1} \leq F_{x_2}$ for $x_1 \leq x_2$. Get some $x^* \in X$ and draw $Y \sim P(x^*, dy)$ and $U \sim \mathcal{U}_{[0,1]}$ independently. Define the modified random map:*

$$C \circ F_x = \begin{cases} Y & \text{if } p(x, F_x) p(x^*, Y) U \leq p(x, Y) p(x^*, F_x) \\ F_x & \text{otherwise} \end{cases} \quad (6)$$

Then:

(i) Marginally, $P(C \circ F_x \in dy) = P(x, dy)$, $x \in X$.

(ii) if $x_1 \leq x_2$ then $C \circ F_{x_1} \leq C \circ F_{x_2}$.

Proof: In the Appendix.

Note that $p(x, \cdot)$ need only be known up to a normalisation constant. It is straightforward that if there exist $x_l, x_u \in X$ such that $x_l \leq x \leq x_u$ for any $x \in X$, then $C \circ F_{x_l} \leq C \circ F_x \leq C \circ F_{x_u}$, so after realising $C \circ F_x$ only for two x 's we can obtain bounds for the updates of all points in X . Clearly, when X is bounded the map $x \mapsto C \circ F_x$ has positive probability of being coalescent. Since $C \circ F_x$ remains increasing in x the *MIC* can be applied repeatedly using each time independent proposed variates; for example $C \circ C \circ F_x \sim P(x, dy)$ will be increasing in x . The effect of the *MIC* depends heavily on using a proposal $p(x^*, \cdot)$ of the same type with $p(x, \cdot)$. In different case, result (ii) of Theorem 2 would not be generally true.

A firm prerequisite for Theorem 2 is that it is possible to obtain a-priori a random map F_x that realises the stochastic ordering of the kernel $P(x, dy)$. Given that, the *MIC* is easy to apply. Although in theory this is always possible using the inverse cdf method (see for example ch.1 of Thorisson, 2000) in practise difficulties related with the availability of certain inverse cdfs are often encountered.

There are also other coupling methods that induce coalescence. The splitting technique (see for instance ch.1 of Thorisson, 2000) obtains the largest possible probability for the coalescence event and so does the layered multi-shift coupler (Wilson, 2000a). However, both methods have limitations in their use.

The layered multi-shift coupler requires that for any $x \in X$, and $F_x \sim P(x, dy)$, the distribution of $F_x - x$ to be independent of x , a strong random-walk assumption. In the case of *TNs*, the layer multi-shift coupler cannot be applied.

For the splitting technique, one needs to find $r(y) := \inf_{x \in C} p(x, y)$, $y \in \mathbb{R}$, and its integral $\int_{\mathbb{R}} r(y) dy$ for a suitably chosen small set $C \subseteq X$ and simulate from the densities proportional to $r(\cdot)$ and $p(x, \cdot) - r(\cdot)$; the calculations and simulations required create difficulties at the application of the method. *MIC* is computationally more efficient than the splitting construction (at least in the case of *TNs*) and benefits from an automatic implementation which does not require application-specific tuning. In fact it can be thought of as a kind of random splitting construction where the smoothness of the transition kernel $P(x, \cdot)$ as a function of x automatically constructs (in a random fashion) an appropriate small set for coupling purposes.

Example: To illustrate the use of the *MIC* assume that $\{\mathcal{N}(\mu, \sigma^2); \mu \in M \subset \mathbb{R}\}$ is a family of univariate Gaussian distributions with mean μ and constant variance σ^2 . It can be equivalently represented by the (unnormalised) kernel $p(\mu, y) = \exp\{-(y - \mu)^2/2\sigma^2\}$, $y \in \mathbb{R}$, which satisfies both conditions of Proposition 1. A trivial way to realise the stochastic ordering of the specified family w.r.t. μ is by setting $F_\mu = \mu + \sigma Z$ for a common $Z \sim \mathcal{N}(0, 1)$. Theorem 2 now ascertains that if some μ^* is picked up and a variate $Y \sim \mathcal{N}(\mu^*, \sigma^2)$ is proposed the modified map $C \circ F_\mu$ will remain increasing in μ . The same procedure can be applied repeatedly with independent Y 's (Fig.2).

3.1 The MIC in the context of *TNs*

In the Perfect Simulation algorithms to be presented the *MIC* is applied to the univariate updates of a Gibbs sampler step at the end of the block of the One-Shot algorithm. According to the One-Shot rule, this final step is the only one within the block that attempts to merge the paths started from everywhere in the state space at the beginning of the block. The analysis that follows is thus focused on coalescence.

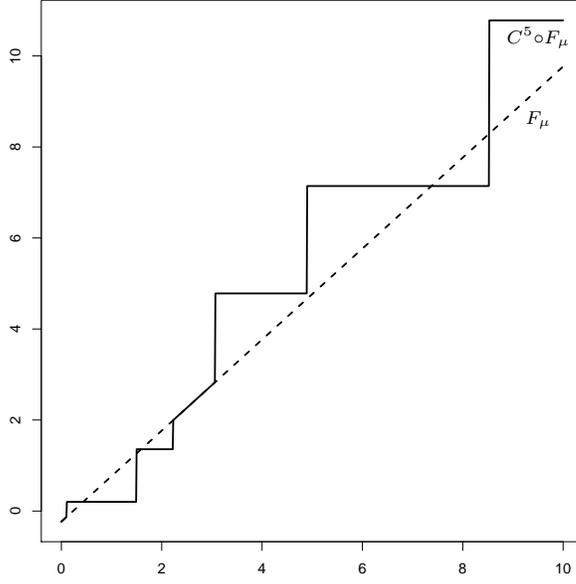


Figure 2: The dashed line depicts the map $F_\mu = \mu + Z \sim \mathcal{N}(\mu, 1)$, $\mu \in [0, 10]$ whereas the solid line the modified map $C^5 \circ F_\mu$ after a 5-fold implementation of the *MIC*. $C^5 \circ F_\mu \sim \mathcal{N}(\mu, 1)$ remains increasing in μ .

To update the i -th element of some \mathbf{x} according to the Gibbs scheme on $TN_d(\mathbf{0}, Q^{-1})|_{\mathbf{a}}^{\mathbf{b}}$ we need to sample from the univariate $TN_1(\mu_i(\mathbf{x}_{-i}), 1)|_{\alpha_i}^{\beta_i}$ for $\mu_i(\mathbf{x}_{-i}) = -\sum_{j:j \neq i} q_{ij} x_j$. Certainly, we will have to update a set of points so we need to devise a random map $F(\mu; \alpha, \beta) \sim TN_1(\mu, 1)|_{\alpha}^{\beta}$ for all μ in some appropriate interval $M = [\mu_l, \mu_u]$. We will give the main idea for the generation of $F(\mu; \alpha, \beta)$ and the way that we check if $\mu \mapsto F(\mu; \alpha, \beta)$ is coalescent for the case of both algorithms (the case-specific and the pure algorithm). Details on the incorporation of this univariate update rule in the general frame of the algorithms are given in the following sections. Note that the family $\{TN_1(\mu, 1)|_{\alpha}^{\beta}, \mu \in M\}$ can be represented by the kernel:

$$p(\mu, y) = \exp\left\{-\frac{(y - \mu)^2}{2}\right\} \mathbb{I}_{[\alpha, \beta]}(y), \quad y \in \mathbb{R}, \quad \text{for } \mu \in M \quad (7)$$

which satisfies both conditions of Proposition 1.

In the case-specific algorithm the use of the cdf of the $\mathcal{N}(0, 1)$ and its inverse can yield a random map $F(\mu; \alpha, \beta) \sim p(\mu, \cdot)$ increasing in $\mu \in M$. Theorem 2 now ascertains that we can produce a modification $C \circ F(\mu; \alpha, \beta) \sim p(\mu, \cdot)$ that remains increasing in μ and has the additional property of inducing coalescence. Clearly, if $C \circ F(\mu_l; \alpha, \beta) = C \circ F(\mu_u; \alpha, \beta)$ then coalescence has happened.

In the pure algorithm, we will be in position to produce a random map $F(\mu; \alpha, \beta) \sim p(\mu, \cdot)$ piece-wise increasing in μ . Considering each piece separately we can use the *MIC* and obtain a modification $C \circ F(\mu; \alpha, \beta) \sim p(\mu, \cdot)$ that remains piece-wise increasing in μ . Recall that the modified map is generated after proposing a $Y \sim p(\mu^*, \cdot)$ for some $\mu^* \in \mathbb{R}$ instead of $F(\mu; \alpha, \beta)$. It is important that we use the same Y for all the increasing pieces since this makes it possible for $C \circ F(\mu; \alpha, \beta)$ to be coalescent. Clearly, we need to determine $C \circ F(\mu; \alpha, \beta)$ explicitly only for the μ 's at the thresholds of the increasing pieces of $F(\mu; \alpha, \beta)$ to decide if coalescence occurred.

4 The Case-Specific Algorithm

This is the case when the cdf Φ of the $\mathcal{N}(0, 1)$ and its inverse Φ^{-1} are used. Thus, it involves an algorithm that does not readily extend beyond the gaussian family. The One-Shot algorithm block involves three phases. For each phase we are going to give the update rule and describe the way we locate in practise the position of the sample paths having started from everywhere in the support of the target density π given at (1) after carrying out the certain update. Notice that we always have to monitor the updates of the coalescence path and therefore apply explicitly the provided update rules for the current location of this single trajectory.

In what follows in this section and Section 5 it is necessary that we are provided with a fast generator of draws from the univariate distributions $TN_1(\mu, 1)|_\alpha^\beta$ and the one with density:

$$q(x) \propto \exp\left(-\frac{|x|}{\epsilon}\right), \quad x \in [\alpha, \beta] \quad (8)$$

for $\epsilon > 0$ and $\alpha, \beta \in \bar{\mathbb{R}}$, $\alpha < \beta$. We denote this distribution by $\text{Exp}(\epsilon)|_\alpha^\beta$. An efficient and numerically stable algorithm for sampling from the $TN_1(\mu, 1)|_\alpha^\beta$ for any values of the parameters is presented in the Appendix. For the case of the $\text{Exp}(\epsilon)|_\alpha^\beta$ a simple algorithm involves well-known methods for simulating truncated univariate exponential densities, so we avoid the details. From now on it is assumed that we can call on the subroutines `tr_normal`(α, β, μ) and `tr_exp`(α, β, ϵ) to draw from these two univariate distributions.

Phase 1: Forcing Paths into a Compact Set

Consider the density:

$$q(\mathbf{x}) \propto \exp\left(-\sum_{i=1}^d \frac{|x_i|}{\epsilon}\right) \cdot \mathbb{I}_S(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d$$

where $S = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{a} \preceq \mathbf{x} \preceq \mathbf{b}\}$ is identical to the support of π and $\epsilon > 0$. Sampling from q is straightforward after calling d times on the `tr_exp`(a_i, b_i, ϵ) method. Assume that $B \sim q$ and $U \sim \mathcal{U}_{[0,1]}$ independently. Consider the random update function:

$$\phi_{\mathcal{I}}(\mathbf{x}) = \begin{cases} B & \text{if } \pi(\mathbf{x})q(B)U \leq \pi(B)q(\mathbf{x}) \\ \mathbf{x} & \text{otherwise} \end{cases}, \quad \mathbf{x} \in S \quad (9)$$

This is just an Independence sampler step known to preserve π . It is easy to prove that if the support of π allows the consideration of the limit $|\mathbf{x}| \rightarrow \infty$ then:

$$\lim_{|\mathbf{x}| \rightarrow \infty} \frac{\pi(\mathbf{x})}{q(\mathbf{x})} = 0 \quad (10)$$

i.e. q is heavier tailed than π . From (9), (10), it is now clear that points sufficiently distanced from the bulk of the target distribution π accept B rendering the image $\phi_{\mathcal{I}}(S)$ a bounded set. More analytically, assume that R_B is the set of points not accepting B . Then:

$$R_B = \left\{ \mathbf{x} \in S : U > \frac{\pi(B)q(\mathbf{x})}{\pi(\mathbf{x})q(B)} \right\} = \left\{ \mathbf{x} \in S : \epsilon \mathbf{x}^\top Q \mathbf{x} - 2 \sum_{i=1}^d |x_i| < c \right\}$$

for $c = 2\epsilon \log U - 2 \sum |B_i| + \epsilon B^\top Q B$. The following proposition helps in locating a superset of R_B .

Proposition 2 Let $V \in \mathbb{R}^{d \times d}$ be a symmetric, positive definite matrix. Then, for any $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{x}^\top V \mathbf{x} \geq \lambda_V \mathbf{x}^\top \mathbf{x}$, where $\lambda_V > 0$ is the minimum eigenvalue of V .

Proof: In the Appendix.

This Proposition yields that $R_B \subseteq \{\mathbf{x} \in S : \lambda_Q \epsilon \mathbf{x}^\top \mathbf{x} - 2 \sum |x_i| < c\}$, where λ_Q is the minimum eigenvalue of Q , and after setting $\epsilon = 1/\lambda_Q$ and adjusting c for this value of ϵ :

$$\begin{aligned} R_B \subseteq \{\mathbf{x} \in S : \mathbf{x}^\top \mathbf{x} - 2 \sum |x_i| < c\} &= \{\mathbf{x} \in S : \sum (|x_i| - 1)^2 < c + d\} \subseteq \\ &\subseteq \{\mathbf{x} \in S : |x_i| \leq \sqrt{c+d} + 1, 1 \leq i \leq d\} \end{aligned}$$

If $c + d < 0$ then $R_B = \emptyset$ and all the state space is projected onto B . Consider the sets $R^i = \{x \in [a_i, b_i] : |x| \leq \sqrt{c+d} + 1\}$, $1 \leq i \leq d$. Clearly, the image $\phi_{\mathcal{I}}(S)$ is a subset of the hyper-rectangle, call it H , determined by the extreme points \mathbf{x}_l and \mathbf{x}_u with individual elements $x_{l,i} = \min\{\psi_{l,i}, B_i\}$ and $x_{u,i} = \max\{\psi_{u,i}, B_i\}$ for $\psi_{l,i} = \inf R^i$ and $\psi_{u,i} = \sup R^i$, $1 \leq i \leq d$. As

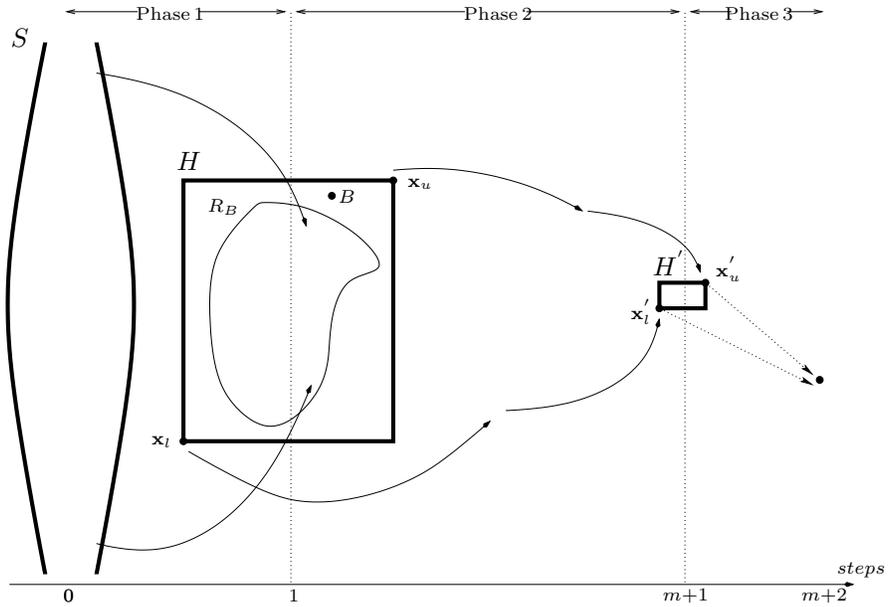


Figure 3: A Block of the One-Shot Algorithm on TNs . An Independence sampler step (Phase 1) brings paths into the bounded set H , m Gibbs steps (Phase 2) force paths into the smaller set H' and the attempt for coalescence takes place at an additional Gibbs step (Phase 3).

shown at Phase 1 of Fig.3 paths from everywhere in S are located into the identifiable bounded set H after a single Independence sampler step.

Phase 2: Bridging the Gap

This part of the block makes use of a number of Gibbs sampler steps. The objective is to force paths from H into some smaller hyper-rectangle H' . The critical point is that when the precision matrix Q is Stieltjes the Gibbs sampler on π exhibits a monotonicity property that makes necessary (for finding H') the monitoring of only two paths, an upper and a lower starting

from \mathbf{x}_u and \mathbf{x}_l respectively. Recall that if $\mathbf{X} \sim \pi$ then $X_i | \mathbf{X}_{-i} = \mathbf{x}_{-i} \sim TN_1(\mu_i(\mathbf{x}_{-i}), 1)|_{\alpha_i}^{b_i}$ for $\mu_i(\mathbf{x}_{-i}) = -\sum_{j:j \neq i} q_{ij} x_j$. To sample from this conditional density we resort to the inverse cdf method. The inverse cdf of the $TN_1(\mu, 1)|_{\alpha}^{\beta}$ is:

$$D^{-1}(u; \mu, \alpha, \beta) := \Phi^{-1} \{u \Phi(\beta - \mu) + (1 - u) \Phi(\alpha - \mu)\} + \mu, \quad u \in [0, 1] \quad (11)$$

and is increasing in μ , i.e. $D^{-1}(\cdot; \mu_1, \alpha, \beta) \leq D^{-1}(\cdot; \mu_2, \alpha, \beta)$ for $\mu_1 \leq \mu_2$.

The update of the i -th element of \mathbf{x} under the Gibbs scheme can be given by the random map:

$$\phi_g^i(\mathbf{x}) := (x_1, \dots, x_{i-1}, D^{-1}(\xi; \mu_i(\mathbf{x}_{-i}), \alpha_i, b_i), x_{i+1}, \dots, x_d)$$

for $\xi \sim \mathcal{U}_{[0,1]}$. When $\mathbf{x} \preceq \mathbf{y}$ it is true that $\mu_i(\mathbf{x}_{-i}) \leq \mu_i(\mathbf{y}_{-i})$, so $\phi_g^i(\mathbf{x}) \preceq \phi_g^i(\mathbf{y})$. A full sweep of the Gibbs sampler will be:

$$\phi_g(\mathbf{x}) := \phi_g^d \circ \phi_g^{d-1} \circ \dots \circ \phi_g^2 \circ \phi_g^1(\mathbf{x})$$

and clearly $\phi_g(\mathbf{x}) \preceq \phi_g(\mathbf{y})$ for $\mathbf{x} \preceq \mathbf{y}$. In general, many iterations, say m , of the Gibbs sampler will be carried out. The overall random map for this part of the block will be:

$$\phi_{\mathcal{G}}(\mathbf{x}) := \phi_g \circ \phi_g \circ \dots \circ \phi_g(\mathbf{x}) = \phi_g^{(m)}(\mathbf{x}), \quad \mathbf{x} \in H \quad (12)$$

Since $\phi_{\mathcal{G}}(\mathbf{x}_l) \preceq \phi_{\mathcal{G}}(\mathbf{x}) \preceq \phi_{\mathcal{G}}(\mathbf{x}_u)$ for all $\mathbf{x} \in H$, when the Gibbs steps are finished all paths have been forced into the tractable hyper-rectangle H' determined by $\mathbf{x}'_l = \phi_{\mathcal{G}}(\mathbf{x}_l)$ and $\mathbf{x}'_u = \phi_{\mathcal{G}}(\mathbf{x}_u)$. That means that $\phi_{\mathcal{G}} \circ \phi_{\mathcal{T}}(S) \subseteq H' \equiv \phi_{\mathcal{G}}(H)$, see Fig.3 - Phase 2.

Phase 3 : One-Shot Coalescence

We proceed with one more Gibbs step adjoined with an effort to achieve coalescence of the paths. As pointed out in Subsection 3.1, the idea is to use the Monotone Independence Coupler (*MIC*) at each of the d univariate updates of a full Gibbs sweep. Coalescence is declared when all d attempts are successful. An alternative choice would be the splitting technique, its implementation however would have been much more demanding.

To perform a univariate update of the Gibbs sampler for all points in some certain set we need to draw from $TN_1(\mu, 1)|_{\alpha}^{\beta}$, or equivalently from the kernel $p(\mu, \cdot)$ given in (7), for any μ in some set $M = [\mu_l, \mu_u]$ determined by the extreme points of the set and appropriate α, β . For instance, at the first univariate update we need to construct a map for all $\mathbf{x} \in H'$, so $\mu_l = \mu_1(\mathbf{x}'_{l,-1})$ and $\mu_u = \mu_1(\mathbf{x}'_{u,-1})$, for $\mu_i(\mathbf{x}_{-i})$ defined in (2).

The *MIC* involves two phases. First we construct a random map $F(\mu; \alpha, \beta) \sim p(\mu, \cdot)$ exactly as in the previous phase of the algorithm; we set $F(\mu; \alpha, \beta) = D^{-1}(\xi; \mu, \alpha, \beta)$ for $\xi \sim \mathcal{U}_{[0,1]}$ and D^{-1} defined in (11). Recall that $F(\mu; \alpha, \beta)$ is increasing in μ . Then we pick up some $\mu^* \in \mathbb{R}$ and draw $Y \sim p(\mu^*, \cdot)$ and $U \sim \mathcal{U}_{[0,1]}$ independently. Consider the modified map (for simplicity we write F_{μ} instead of $F(\mu; \alpha, \beta)$):

$$C \circ F_{\mu} = \begin{cases} Y & \text{if } p(\mu, F_{\mu}) p(\mu^*, Y) U \leq p(\mu, Y) p(\mu^*, F_{\mu}) \\ F_{\mu} & \text{otherwise} \end{cases}$$

We have already pointed out that the kernel p satisfies both conditions of Proposition 1, so Theorem 2 guarantees that $C \circ F_{\mu} \sim p(\mu, \cdot)$ and is increasing in μ . At the actual implementation of the algorithm we choose $\mu^* = (\mu_l + \mu_u)/2$ so that the probability of coalescence is relatively high and evaluate $C \circ F_{\mu}$ only for $\mu = \mu_l$ and $\mu = \mu_u$. These two values are the updates of a certain co-ordinate of the sandwiching paths and will be used at the next univariate update to find the new boundaries μ_l, μ_u .

For the general algorithm, we perform the i -th univariate update of the final Gibbs step via the random map:

$$\phi_c^i(\mathbf{x}) := (x_1, \dots, x_{i-1}, C \circ F(\mu_i(\mathbf{x}_{-i}); a_i, b_i), x_{i+1}, \dots, x_d)$$

For $\mathbf{x} \preceq \mathbf{y}$ it is clear that $\mu_i(\mathbf{x}_{-i}) \leq \mu_i(\mathbf{y}_{-i})$, so $\phi_c^i(\mathbf{x}) \preceq \phi_c^i(\mathbf{y})$. The complete update rule is:

$$\phi_C(\mathbf{x}) := \phi_c^d \circ \phi_c^{d-1} \circ \dots \circ \phi_c^2 \circ \phi_c^1(\mathbf{x}), \quad \mathbf{x} \in H' \quad (13)$$

and is increasing in \mathbf{x} . Someone needs to consider its effect only on the two extreme paths to determine if coalescence is achieved. If at some intermediate univariate update the corresponding elements of the extreme paths do not couple we declare the block unsuccessful and update only the coalescence path according to the transition kernel determined by the Gibbs sampler (see the discussion in Section 2).

Synthesis

The compound update rule for a block will be $\Phi = \phi_C \circ \phi_G \circ \phi_I$. It preserves π since all its individual update functions defined in (9), (12) and (13) do. Because of the Independence sampler step and the fact that we can only locate a superset H of $\phi_I(S)$ it is obvious that $\Phi(S)$ is intractable. As already shown, we can easily obtain some $\Phi' \supseteq \Phi(S)$ determined as $\Phi' = \phi_C(\phi_G(H))$. Φ' can be a singleton with positive probability increasing in the number of Gibbs steps within each block.

It is important that the finite accuracy of the computing systems restricts the availability of the cdf Φ of the standard normal distribution and its inverse Φ^{-1} . The double-precision GSL library for the C-language returns values for $\Phi(x)$ only when $|x| < 38$. For instance, when the inverse cdf formula (11) is used for the $TN_1(0, 1)|_{40}^{50}$ it returns $+\infty$. If it happens that a value $\Phi(x)$ for x outside of the range $[-38, 38]$ is needed we instantly declare the block unsuccessful (we cannot realise the monotonicity property anymore) and update only the coalescence path until the end of the block. The validity of this compromise is justified after declaring a block successful if the event $D = \{\Phi' \text{ is a singleton}\} \cap \{\text{at no point of the block is a value } \Phi(x) \text{ for } |x| > 38 \text{ requested}\}$ occurs and use it as the sub-event of the actual coalescence event $\{\Phi(S) \text{ is a singleton}\}$ at the demonstration of the Ro-CFTP protocol in Section 2.

5 The Pure Algorithm

Recall that the target distribution is π given in (1) for $\mu = \mathbf{0}$, support S and Stieltjes precision matrix $Q = (q_{ij})$ with $q_{ii} = 1$, $1 \leq i \leq d$. This time the use of Φ or Φ^{-1} is not considered yielding the prospect of implementing the resulted algorithm in a context more general than that of TNs . The phases in the construction of a block are the same as for the case-specific algorithm. Again, for each phase we give the official update rule and describe the way we identify in practise the position of the paths after that update has taken place. Note that the information we need at the end of a block is the location of the coalescence path (so it is always presumed that we explicitly apply the update rules for it) and whether it is successful or not.

Phase 1: Forcing the Paths into a Compact Set

It is identical to Phase 1 of the case-specific algorithm. It yields the random map ϕ_I (9) that projects all elements of S onto a hyper-rectangle H determined by two identifiable points \mathbf{x}_l and \mathbf{x}_u , see Fig.3 - Phase 1.

Phase 2: Bridging the Gap

Assume that a path is located at some $\mathbf{x} \in S$ and the update of the i -th element of \mathbf{x} is about to take place via Gibbs sampler. The full conditional distribution of the i -th element of a random vector distributed according to π is given in (2). The case-specific algorithm exploited the fact that the inverse cdf of this univariate distribution is increasing in \mathbf{x} and implemented the Gibbs steps only for the sandwiching paths found at $\mathbf{x}_l, \mathbf{x}_u$ before the effect of the Gibbs sampler. Without Φ, Φ^{-1} realising the stochastic ordering of the family $\{TN_1(\mu, 1)|_\alpha^\beta, \mu \in M \subseteq \mathbb{R}\}$ is impossible. Instead we opt for an adaptive rejection sampling scheme. It yields two bounding paths that set off from \mathbf{x}_l and \mathbf{x}_u , squeeze all paths between them but are technically constructed, i.e. they do not represent actual realisations of the underlying Markov chain but a selection of instances among several paths.

Let ϕ_g^i denote the update rule under the i -th univariate Gibbs step. To construct ϕ_g^i two methods are considered. The first (Fig.4) provides a random map $R_1(\mu; \alpha, \beta) \sim TN_1(\mu, 1)|_\alpha^\beta$ for all $\mu \in [\mu_l, \mu_u] \subseteq [\alpha, \beta]$. It involves rejection sampling that uses as envelopes the corresponding untruncated densities. If $(Z + \mu) \sim \mathcal{N}(\mu, 1)$ lies in $[\alpha, \beta]$ then $Z + \mu$ is accepted as a draw for $TN_1(\mu, 1)|_\alpha^\beta$. It uses the same Z for all μ 's and is adaptive since the μ 's that accept $Z + \mu$ are located and a subset of $[\mu_l, \mu_u]$ is considered for the rest of the algorithm. The algorithm terminates when $R_1(\mu; \alpha, \beta)$ is constructed for all $\mu \in [\mu_l, \mu_u]$.

```

set  $\mu_{\min} = \mu_l, \mu_{\max} = \mu_u$ 
while ( $\mu_{\min} \neq \mu_{\max}$ ) repeat
  sample increment  $Z \sim \mathcal{N}(0, 1)$ 
  if ( $Z + \mu_{\min} \leq \beta$  and  $Z + \mu_{\max} \geq \alpha$ )
    if ( $Z + \mu_{\min} \geq \alpha$  and  $Z + \mu_{\max} \leq \beta$ )
      store  $(\mu_{\min}, Z + \mu_{\min}), (\mu_{\max}, Z + \mu_{\max}),$  set  $\mu_{\max} = \mu_{\min}$ 
    else if ( $Z + \mu_{\min} \geq \alpha$ )
      store  $(\mu_{\min}, Z + \mu_{\min}), (\beta - Z, \beta),$  set  $\mu_{\min} = \beta - Z$ 
    else if ( $Z + \mu_{\max} \leq \beta$ )
      store  $(\alpha - Z, \alpha), (\mu_{\max}, \mu_{\max} + Z),$  set  $\mu_{\max} = \alpha - Z$ 
return stored points

```

Figure 4: The algorithm that yields a piece-wise increasing random map $\mu \mapsto R_1(\mu; \alpha, \beta)$ with the marginal property that $R_1(\mu; \alpha, \beta) \sim TN_1(\mu, 1)|_\alpha^\beta$ for all μ in $[\mu_l, \mu_u] \subseteq [\alpha, \beta]$.

The second method generates a map $R_2(\mu; \alpha, \beta) \sim TN_1(\mu, 1)|_\alpha^\beta$ for any μ in some interval $[\mu_l, \mu_u]$ when $\mu_u \leq \alpha$ or $\mu_l \geq \beta$. It is a case of *Rejection Coupler* (Murdoch and Green, 1998). The unnormalised density of the $TN_1(\mu, 1)|_\alpha^\beta$ distribution $p(\mu, y) = \exp\{-(y - \mu)^2/2\} \mathbb{I}_{[\alpha, \beta]}(y)$ has the property:

$$\text{if } \mu_1 \leq \mu_2 \leq \alpha \text{ or } \beta \leq \mu_2 \leq \mu_1 \text{ then } p(\mu_1, \cdot) \leq p(\mu_2, \cdot) \quad (14)$$

The algorithm that constructs R_2 (for the case when $\mu_u \leq \alpha$) involves adaptive rejection sampling; use $p(\mu_u, \cdot)$ as an envelope for all the others $p(\mu, \cdot), \mu \in [\mu_l, \mu_u]$, locate the minimum μ that accepts a proposal from $p(\mu_u, \cdot)$ and set that as the new μ_u . It terminates when a draw from $p(\mu_l, \cdot)$ is obtained. For the case when $\mu_l \geq \beta$ we follow the same procedure with the apparent symmetrical considerations. The pseudo-algorithm in Fig.5 is for $\mu_u \leq \alpha$. Fig.6 shows a realisation of the random functions $R_1(\mu; \alpha, \beta), R_2(\mu; \alpha, \beta)$ for the case when $\alpha = -1, \beta = 1$ and $\mu \in [-2, 2]$.

```

set  $\mu_{\min} = \mu_l, \mu_{\max} = \mu_u$ 
while ( $\mu_{\min} \neq \mu_{\max}$ ) repeat
  sample  $Y \sim p(\mu_{\max}, \cdot), U \sim \mathcal{U}_{[0,1]}$ 
  set  $\mu_{\max} = \min\{\mu : \mu_{\min} \leq \mu \leq \mu_{\max} \text{ and } U \leq p(\mu, Y)/p(\mu_{\max}, Y)\}$ 
  store  $(\mu_{\max}, Y)$ 
return stored points

```

Figure 5: The algorithm that generates the piece-wise constant random map $\mu \mapsto R_2(\mu; \alpha, \beta)$ with the marginal property that $R_2(\mu; \alpha, \beta) \sim TN_1(\mu, 1)|_{\alpha}^{\beta}$ for all μ in $[\mu_l, \mu_u] \subseteq (-\infty, \alpha]$.

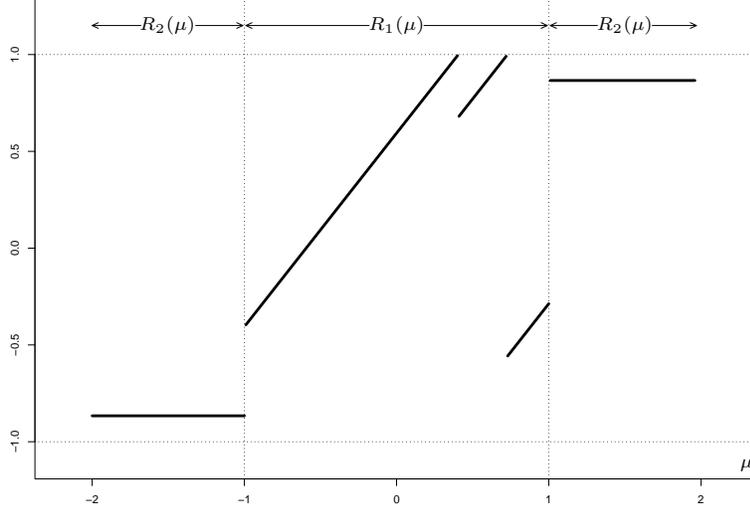


Figure 6: A realisation of the random update maps $R_1(\mu; -1, 1) \sim TN_1(\mu, 1)|_{-1}^1$ for $\mu \in [-1, 1]$ and $R_2(\mu; -1, 1) \sim TN_1(\mu, 1)|_{-1}^1$ for $\mu \in [-2, -1] \cup [1, 2]$.

The overall update rule for the i -th univariate update of a Gibbs step is defined as:

$$\phi_g^i(\mathbf{x}) = \begin{cases} (x_1, \dots, x_{i-1}, R_1(\mu_i(\mathbf{x}_{-i}); a_i, b_i), x_{i+1}, \dots, x_d), & a_i \leq \mu_i(\mathbf{x}_{-i}) \leq b_i \\ (x_1, \dots, x_{i-1}, R_2(\mu_i(\mathbf{x}_{-i}); a_i, b_i), x_{i+1}, \dots, x_d), & \mu_i(\mathbf{x}_{-i}) > b_i \text{ or } \mu_i(\mathbf{x}_{-i}) < a_i \end{cases}$$

where we have assumed that ϕ_g^i applies to a set points within a bounding rectangle so that it is possible to identify the appropriate bounds μ_l, μ_u for the construction of R_1 and R_2 . For instance at the first univariate update of the first Gibbs step this rectangle is H . As the following update rules take place this bounding area is determined by a pair of bounding paths whose construction is detailed in the next paragraph.

A full Gibbs step is given as:

$$\phi_g(\mathbf{x}) := \phi_g^d \circ \phi_g^{d-1} \circ \dots \circ \phi_g^1(\mathbf{x})$$

Carrying out m Gibbs steps according to ϕ_g corresponds to using the random map:

$$\phi_G(\mathbf{x}) := \phi_g \circ \phi_g \circ \dots \circ \phi_g(\mathbf{x}) = \phi_g^{(m)}(\mathbf{x}), \quad \mathbf{x} \in H \quad (15)$$

Clearly, $\phi_G(\mathbf{x})$ is not monotone in \mathbf{x} (see Fig.6) so locating the position of the paths after applying it is not straightforward. To overcome this inconvenience we develop a pair of bounding paths.

We denote by R the complete update map produced after combining R_1 and R_2 . Let $\mathbf{y}_u^{(i,j)}$, $\mathbf{y}_l^{(i,j)}$ be the position of the upper and lower bounding paths after the j -th univariate update of the i -th Gibbs step has taken place, $1 \leq i \leq m$, $0 \leq j \leq d$. These paths begin from the edges of the hyper-rectangle H determined by the Independence sampler step, i.e. $\mathbf{y}_u^{(1,0)} \equiv \mathbf{x}_u$, $\mathbf{y}_l^{(1,0)} \equiv \mathbf{x}_l$, and are developed in a way that $\mathbf{y}_u^{(i,j)}$ ($\mathbf{y}_l^{(i,j)}$) is identical to $\mathbf{y}_u^{(i,j-1)}$ ($\mathbf{y}_l^{(i,j-1)}$) except for its j -th element which is given as the sup $A_{i,j}$ (inf $A_{i,j}$) for the set $A_{i,j}$ defined as $A_{i,j} = \{R(\mu_j(\mathbf{y}_{-j}); a_j, b_j); \mathbf{y}_l^{(i,j-1)} \preceq \mathbf{y} \preceq \mathbf{y}_u^{(i,j-1)}\}$, under the convention that $(i, d) \equiv (i+1, 0)$.

Essentially, at each univariate update of the Gibbs sampler we locate the minimum and the maximum value of the univariate map R that updates the certain co-ordinate for all the points between the current location of the bounding paths and use these values to update the bounding paths. The definition of R renders the identification of these values an easy task. This procedure yields a tractable hyper-rectangle H' determined by the extreme points $\mathbf{x}'_l := \mathbf{y}_l^{(m,d)}$ and $\mathbf{x}'_u := \mathbf{y}_u^{(m,d)}$ that includes the position of all paths started from H after the effect of these m Gibbs steps.

Phase 3: One-Shot Coalescence

As with the case-specific algorithm we try to couple the univariate updates of an additional Gibbs step. Then we were able to construct a random map $F(\mu; \alpha, \beta) \sim TN_1(\mu, 1)|_\alpha^\beta$ (equivalently $F(\mu; \alpha, \beta) \sim p(\mu, \cdot)$ for the kernel p defined in (7)) increasing in μ rendering the effect of the *MIC* straightforward. Now we follow a different procedure.

As in the previous phase of this algorithm we combine, if necessary, R_1, R_2 to get the random map $R(\mu; \alpha, \beta) \sim p(\mu, \cdot)$. The generation of R_1, R_2 ensures that R is piece-wise increasing in μ , see Fig.6. We choose $\mu^* \in \mathbb{R}$ and draw $Y \sim p(\mu^*, \cdot)$ and $U \sim \mathcal{U}_{[0,1]}$ independently. Consider the modified map:

$$C \circ R(\mu; \alpha, \beta) = \begin{cases} Y & \text{if } \mu \in [\alpha, \beta] \text{ and } p(\mu, R_\mu) p(\mu^*, Y) U \leq p(\mu, Y) p(\mu^*, R_\mu) \\ R_\mu & \text{if } \mu \in [\alpha, \beta] \text{ and } p(\mu, R_\mu) p(\mu^*, Y) U > p(\mu, Y) p(\mu^*, R_\mu) \\ R_\mu & \text{otherwise} \end{cases} \quad (16)$$

where we wrote R_μ instead of $R(\mu; \alpha, \beta)$. The new update rule implements the *MIC* on R only for $\mu \in [\alpha, \beta]$. For $\mu < \alpha$ or $\mu > \beta$ the definition of R itself provides a chance for coalescence without the need for modification. From Theorem 2(i) it is clear that $C \circ R(\mu; \alpha, \beta) \sim p(\mu, \cdot)$ for any μ considered. Theorem 2(ii) ascertains that $C \circ R(\mu; \alpha, \beta)$ remains piece-wise increasing in μ .

Assume that a pair of bounding paths (the construction of which we explain in the next paragraph) determine that we need to use (16) for all $\mu \in [\mu_l, \mu_u]$. A careful consideration of (16) indicates that if $\mu_u \leq \alpha$ or $\mu_l \geq \beta$ then we use the Rejection Coupler as the means for coalescence. If $[\mu_l, \mu_u] \subseteq [\alpha, \beta]$ then we apply the *MIC* for $\mu^* = (\mu_l + \mu_u)/2$ and we can determine if the modified update $C \circ R(\mu; \alpha, \beta)$ is coalescent after realising it only for the extreme points of the increasing pieces of $R(\mu; \alpha, \beta)$. In any other case coalescence cannot happen, so we declare the block unsuccessful and update just the coalescence path to the end of the block according to the marginal kernel of the Gibbs sampler.

We do not need to resort to a sophisticated construction of bounding paths in the way described in the previous part of the algorithm. Recall that this is an One-Shot algorithm, so if a univariate Gibbs update fails to merge the element it updates for all points in an area of interest then we declare the block unsuccessful and update only the coalescence path. Clearly, given that after the i -th Gibbs update all univariate updates have been coalescent the area we mention above is the hyper-rectangle determined by a pair of points equal to $\mathbf{x}'_l, \mathbf{x}'_u$ except for their first i -elements which have been updated and are identical.

In the general context of the algorithm, the i -th update of a Gibbs step is given by the random map:

$$\phi_c^i(\mathbf{x}) := (x_1, \dots, x_{i-1}, C \circ R(\mu_i(\mathbf{x}_{-i}); a_i, b_i), x_{i+1}, \dots, x_d) \quad (17)$$

The complete update rule is:

$$\phi_C(\mathbf{x}) := \phi_c^d \circ \phi_c^{d-1} \circ \dots \circ \phi_c^2 \circ \phi_c^1(\mathbf{x}), \quad \mathbf{x} \in H' \quad (18)$$

Synthesis

The random update map for a block of the pure algorithm is the synthesis $\Phi = \phi_C \circ \phi_G \circ \phi_I$, where the constituent update maps are defined in (9), (15) and (18). It has to be mentioned that, in theory, the hyper-rectangle H determined after the Independence sampler step could be extremely large for some blocks and this could create some problems related with the time required to carry out the Rejection Coupler scheme of the subsequent Gibbs steps. For example, it could happen that at a univariate update of the Gibbs step it is required that we construct the map $R_2(\mu; 0, 10) \sim TN_1(\mu, 1)|_0^{10}$ for all $\mu \in [10, 2000]$ which would require considerable time. To avoid such problems we could instantly declare a block unsuccessful when the map $R_2(\mu; \alpha, \beta)$ should be determined for $\mu \in [\mu_l, \mu_u]$ when $\max\{(\mu_u - \beta), (\alpha - \mu_l)\}$ is bigger than some threshold of our choice. For the cases considered in Section 7 when we run the algorithms for various target distributions, such problems with the Independence sampler step did not occur so we avoid any further reference to this subtle point.

In total, for the pure algorithm a block is declared successful when the sub-event $D = \{\text{the bounding processes coalesce}\}$ of the actual coalescence event $\{\Phi(S) \text{ is a singleton}\}$ takes place.

6 Extensions

The truncated gaussian distribution (1) is used throughout this paper mostly for reasons of demonstrating the applicability of the *MIC* monotone coupling technique introduced in Section 3 and the various continuous state space coupling constructions of Sections 4 and 5 in a realistic setting. In Subsection 6.2 we give a hint about how similar constructions can be applied to other families of distributions. Before that we make a simple observation about the gaussian case.

6.1 Extensions within the Gaussian family

Although the preceding algorithms appear to be applicable to a certain class of *TNs*, those with a Stieltjes precision matrix, a superset of this class can be treated in a similar way. Consider the following two structures of a $d \times d$, real, symmetric, positive definite matrix:

$$\begin{pmatrix} 1 & - & + & \dots & - & - \\ - & 1 & + & \dots & - & - \\ + & + & 1 & \dots & + & + \\ - & - & + & \dots & - & - \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ - & - & + & \dots & - & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & - & - & \dots & - & - \\ - & 1 & - & \dots & - & - \\ - & - & 1 & \dots & - & - \\ - & - & - & \dots & - & - \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ - & - & - & \dots & - & 1 \end{pmatrix}$$

If $\mathbf{X} \sim TN_d(\mathbf{0}, Q^{-1})|_{\mathbf{a}}^{\mathbf{b}}$ with Q having the left side structure then $\mathbf{X}' = (X_1, X_2, -X_3, X_4, \dots, X_d)$ is distributed according to $TN_d(\mathbf{0}, Q_o^{-1})|_{\mathbf{a}'}^{\mathbf{b}'}$ with \mathbf{a}' , \mathbf{b}' identical to \mathbf{a} , \mathbf{b} respectively except for $a'_3 = -b_3$ and $b'_3 = -a_3$ and Q_o a matrix with the right side structure, i.e. a Stieltjes matrix (it

can be shown that Q_o remains positive definite). This observation ascertains that our methodology could be implemented on a class of TN s wider than the one so far considered. In the following definition $r_i : \mathbb{R}^{d \times d} \mapsto \mathbb{R}^{d \times d}$ represents a map which when applied to a $d \times d$ matrix changes the sign of the non-diagonal elements of the i -th row and column, $1 \leq i \leq d$, and r_0 the identity mapping that maps a matrix to itself.

Definition 2 *The Positive Association Equivalence Class (PAEC) of truncated Gaussian distributions includes distributions identified as $TN_d(\mu, Q^{-1})|_{\mathbf{a}}^{\mathbf{b}}$ for some $d \in \{1, 2, \dots\}$, $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$, $\mu \in \mathbb{R}^d$ and $Q \in \mathbb{R}^{d \times d}$ a symmetric and positive definite matrix for which there exists a sequence of operators $(r_{i_j})_{j=1}^k$ such that:*

$$r_{i_1} \circ r_{i_2} \circ \dots \circ r_{i_k}(Q)$$

is a Stieltjes matrix, $\{i_1, i_2, \dots, i_k\} \subseteq \{0, 1, 2, \dots, d\}$.

This sign transformation is called *Parity Switching* in Roberts and Tweedie (2005). Clearly, the pure and the case-specific algorithms can be applied to any member of this class of TN s. Actually, even when $TN_d(\mu, Q^{-1})|_{\mathbf{a}}^{\mathbf{b}} \notin PAEC$ it is still possible to call on the same algorithms provided that:

$$\sum_{j:j \neq i} |q_{ij}| < 1 \quad \text{for all } 1 \leq i \leq d \quad (19)$$

While nothing changes for the Independence sampler step a modification of the Gibbs steps is necessary. If the i -th updating of a Gibbs step is taking place with the paths found in the hyper-rectangle determined by some extreme points $\mathbf{x}_l, \mathbf{x}_u$ then a map $F_\mu \sim TN_1(\mu, 1)|_{a_i}^{b_i}$ for all μ such that:

$$-\sum_{j:j \neq i} q_{ij} \{x_{l,j} \mathbb{I}(q_{ij} \leq 0) + x_{u,j} \mathbb{I}(q_{ij} > 0)\} \leq \mu \leq -\sum_{j:j \neq i} q_{ij} \{x_{u,j} \mathbb{I}(q_{ij} \leq 0) + x_{l,j} \mathbb{I}(q_{ij} > 0)\}$$

is constructed in the way described in Phase 2 of the case-specific and pure algorithms. The i -th elements of $\mathbf{x}_l, \mathbf{x}_u$ are updated with the minimum and the maximum value of the map F_μ respectively. Condition (19) guarantees the convergence of the bounding paths constructed by the update rule F_μ .

6.2 Further Directions for Extensions

The Perfect Simulation algorithms of Sections 4 and 5 can apply in a similar way to any multivariate distribution of arbitrary dimension d for which:

P₁) for any $1 \leq i \leq d$, the density of the distribution of $X_i | \mathbf{X}_{-i} = \mathbf{x}_{-i}$ can be written as $p_i(\epsilon_i(\mathbf{x}_{-i}), \cdot)$ where $\epsilon_i : \mathbb{R}^{d-1} \mapsto \mathbb{R}$, is locally bounded, and

P₂) the density $p_i(\epsilon, \cdot)$ in P₁ is stochastically monotone in ϵ , $1 \leq i \leq d$.

The stochastic ordering in P₂ can be increasing in the parameter, i.e. $p_i(\epsilon_2, y)/p_i(\epsilon_1, y)$ being increasing in y for $\epsilon_1 \leq \epsilon_2$, as in the case of a TN , or decreasing, i.e. $p_i(\epsilon_2, y)/p_i(\epsilon_1, y)$ decreasing in y for $\epsilon_1 \leq \epsilon_2$, as it will be the case for the examples that follow.

These are not the most general conditions possible, but generality comes at the expense of applicability in this problem, so we concentrate on the family of distributions with properties P₁ and P₂ for which we can demonstrate that the algorithms described so far in the paper can be readily generalised. We remark here that P₁ and P₂ are implied by the stronger though possibly easy to check condition that for a density f :

$$\frac{\partial^2}{\partial x_i \partial x_j} \log f(\mathbf{x})$$

is either non-negative for all pairs pair $1 \leq i < j \leq d$ and all \mathbf{x} or non-positive. That this condition implies P_1 and P_2 is easily verified using the FKG inequality (see for example Karlin, 1968). Note that this condition is trivially satisfied for a TN .

Based on the case of a TN , we will provide the guidelines for the development of an One-Shot algorithm on an arbitrary multivariate density, say f , satisfying P_1 and P_2 . For the Independence sampler step we need to find a proposal density with heavier tails than f ; the corresponding update function will then force all paths into a compact set, see Phase 1 of Section 4. This step might even not be necessary, see the examples that follow.

Assume now that before a Gibbs step that will update the i -th co-ordinate of the paths, all paths are located in a compact set, say G . The Gibbs sampler step will require the construction of a random map $\epsilon \mapsto F_\epsilon$ for all ϵ in $E = [\inf_{\mathbf{x} \in G} \epsilon_i(\mathbf{x}_{-i}), \sup_{\mathbf{x} \in G} \epsilon_i(\mathbf{x}_{-i})]$, with the marginal property that $F_\epsilon \sim p_i(\epsilon, \cdot)$ such that it is possible to identify the $\inf_{\epsilon \in E} F_\epsilon$, $\sup_{\epsilon \in E} F_\epsilon$; these extremes will be the updates of the i -th co-ordinate of some lower and upper bounding paths. If the inverse cdf of $p_i(\epsilon, \cdot)$ is available we easily obtain an F_ϵ monotone in ϵ , see Phase 2 of Section 4. Otherwise, it could be possible to construct F_ϵ using some of the advanced coupling techniques of Phase 2 of Section 5 (adaptive rejection sampling, Rejection Coupler).

The final phase of the block involves the attempt for coalescence after appropriately modifying an additional Gibbs step. We follow exactly the same rules as for the case of a TN . If the random maps for the univariate updates of the Gibbs step are monotone we use *MIC* (Phase 3, Section 4); else if they are constructed using adaptive rejection sampling or Rejection Coupler we modify them using *MIC* to induce coalescence or do not modify them and hope for the Rejection Coupler to provide coalescence respectively (Phase 3, Section 5).

For this construction to apply, the upper and lower bounding paths developed within the block must converge. This will depend on the specific density under consideration and could be checked theoretically, see Møller (1999) for the case of the multivariate gamma conditional distribution defined in the sequel, or even experimentally, i.e. from preliminary runnings of the Gibbs steps.

The family of distributions with the properties P_1 and P_2 includes many probability models used in the literature; we will now give two examples.

6.2.1 Multivariate Gamma Conditionals Distribution

The so-called autogamma distribution has already been subject of the Perfect Simulation literature, see Møller (1999). It includes as a specific case a bayesian hierarchical model on a dataset on pump reliability originally presented in Gelfand and Smith (1990) and used among others by Rosenthal (1995); Murdoch and Green (1998); Breyer and Roberts (2001). The density of the autogamma distribution writes as:

$$f(\mathbf{x}) \propto \exp \left\{ - \sum_{i=1}^d \epsilon_i x_i - \sum_{i < j} \epsilon_{ij} x_i x_j \right\} \prod_{i=1}^d x_i^{\delta_i - 1}, \quad \mathbf{x} \in \mathbb{R}^d, \mathbf{x} \succeq \mathbf{0} \quad (20)$$

for $\epsilon_i > 0$, $\delta_i > 0$ and $\epsilon_{ij} = \epsilon_{ji} \geq 0$. If $\mathbf{X} \sim f$ then :

$$X_i | \mathbf{X}_{-i} = \mathbf{x}_{-i} \sim \Gamma(\delta_i, \epsilon_i(\mathbf{x}_{-i})), \quad \epsilon_i(\mathbf{x}_{-i}) := \epsilon_i + \sum_{j:j \neq i} \epsilon_{ij} x_j \quad (21)$$

$\Gamma(\delta, \epsilon)$ denotes the gamma distribution with mean δ/ϵ . Property P_1 is satisfied for $p_i(\epsilon_i(\mathbf{x}_{-i}), \cdot)$ the density of $\Gamma(\delta_i, \epsilon_i(\mathbf{x}_{-i}))$. Also, $p_i(\epsilon, \cdot)$ is stochastically decreasing in ϵ so P_2 holds.

The Independence sampler step is not necessary. For the Gibbs steps we construct the random map $F_\epsilon \sim p_i(\epsilon, \cdot)$ by setting $F_\epsilon = G/\epsilon$ for $G \sim \Gamma(\delta_i, 1)$. After one Gibbs sweep the upper

bounding path initiated at $\{+\infty\}^d$ will have finite values and all paths will have been forced into a compact set; just notice that $\epsilon_i(\mathbf{x}_{-i}) \geq \epsilon_i > 0$, thus the possible values of ϵ are lower bounded and F_ϵ will be upper bounded. As shown in Møller (1999) the bounding paths updated via the supremum and infimum of F_ϵ converge.

Were we given a truncated version of (20), we could still build a Perfect Simulation algorithm using the adaptive rejection sampling techniques of Section 5.

6.2.2 Multivariate Exponential Conditionals Distribution

For details on this family of distributions see for instance ch.8 of Arnold et al. (1999) and Arnold and Strauss (1988). The density is as follows:

$$f(\mathbf{x}) = \exp \left\{ - \sum_{\mathbf{v} \in \xi_d} \lambda_{\mathbf{v}} \left(\prod_{i=1}^d x_i^{v_i} \right) \right\}, \quad \mathbf{x} \in \mathbb{R}^d, \mathbf{x} \succeq \mathbf{0} \quad (22)$$

where $\xi_d := \{\mathbf{v} \in \mathbb{R}^d : v_i = 0 \text{ or } 1 \text{ for all } 1 \leq i \leq d\}$, $\lambda_{\mathbf{v}}$ for $\mathbf{v} \neq \mathbf{0}$ non-negative parameters and λ_0 a normalizing constant. If $\mathbf{X} \sim f$ then:

$$X_i | \mathbf{X}_{-i} = \mathbf{x}_{-i} \sim \text{Exp}(1/\epsilon_i(\mathbf{x}_{-i})), \quad \epsilon_i(\mathbf{x}_{-i}) := \sum_{\mathbf{v} \in \xi_d, v_i=1} \lambda_{\mathbf{v}} \left(\prod_{j:j \neq i} x_j^{v_j} \right)$$

with $\text{Exp}(1/\epsilon)$, $\epsilon > 0$, the exponential distribution with mean $1/\epsilon$. Property P₁ holds for $p_i(\epsilon_i(\mathbf{x}_{-i}), \cdot)$ the density of $\text{Exp}(1/\epsilon_i(\mathbf{x}_{-i}))$; property P₂ also holds since $p_i(\epsilon, \cdot)$ is stochastically decreasing in ϵ .

It is not simple to give a general rule for the Independence sampler step for any values of the parameters $\lambda_{\mathbf{v}}$. In fact, for a sub-family of (22) defined in the sequel this step is not necessary. Assuming that the paths can be projected onto a hyper-rectangle with lower point \mathbf{x}_l of positive co-ordinates and upper point \mathbf{x}_u of co-ordinates $x_{u,j} < \infty$, $1 \leq j \leq d$, the Gibbs steps are then straightforward; we construct the random map $F_\epsilon \sim p_i(\epsilon, \cdot)$ for any required ϵ in a bounded interval E using the inverse cdf method:

$$F_\epsilon = -\log(1 - \xi)/\epsilon, \quad \xi \sim \mathcal{U}_{[0,1]} \quad (23)$$

F_ϵ is decreasing in ϵ , so we can easily find $\inf_{\epsilon \in E} F_\epsilon$ and $\sup_{\epsilon \in E} F_\epsilon$ which are used for the update of the i -th co-ordinate of the lower and upper bounding paths initiated from \mathbf{x}_l and \mathbf{x}_u respectively. The coalescence step will be a simple application of *MIC*.

We return to the Independence sampler step. Consider the densities (see ch.8 of Arnold et al., 1999):

$$f(\mathbf{x}) \propto \exp \left\{ - \sum_{i=1}^d x_i - \delta \prod_{i=1}^d x_i \right\}, \quad \mathbf{x} \in \mathbb{R}^d, \mathbf{x} \succeq \mathbf{0} \quad (24)$$

generated by (22) under the specifications $\lambda_{\mathbf{v}} = \delta > 0$ for $\mathbf{v} = (1, 1, \dots, 1)$, $\lambda_{\mathbf{v}} = 1$ for $\mathbf{v} \in \xi_d$ such that $\sum v_i = 1$, and $\lambda_{\mathbf{v}} = 0$ for all other $\mathbf{v} \neq \mathbf{0}$. In this case

$$\epsilon_i(\mathbf{x}_{-i}) = 1 + \delta \prod_{j:j \neq i} x_j \geq 1. \quad (25)$$

Consider a Gibbs sampler step under the rule (23) for all points in the domain of (24); from (25), F_ϵ will need to apply to all ϵ in $E = [1, \infty)$. Yet, $\inf_{\epsilon \in E} F_\epsilon = \lim_{\epsilon \rightarrow \infty} F_\epsilon = 0$ and $\sup_{\epsilon \in E} F_\epsilon = F_1$ so after one full Gibbs sweep the upper bounding path initiated at $\{+\infty\}^d$ will have finite co-ordinates; the rest of the Gibbs steps will be carried out in the regular way. It can be shown that the bounding paths updated by the supremum and infimum of the maps F_ϵ converge.

7 Simulation Results

For this section we use the abbreviations *RS*:=Rejection Sampling, *PA*:=Pure Algorithm and *CSA*:=Case-Specific Algorithm.

7.1 Running the Algorithms

To try to eliminate the possibility of coding error, we first applied *PA* and *CSA* to a distribution which can be routinely simulated using *RS*. Thus we consider the tri-variate distribution:

$$TN_3(\mathbf{0}, Q^{-1})|_{\mathbf{a}}^{\mathbf{b}}, \text{ for } \mathbf{a}=\{-\infty\}^3, \mathbf{b}=\{+\infty\}^3 \text{ and } Q=(q_{ij}) \text{ with } q_{ii}=1, q_{ij}=-0.4 \text{ if } i \neq j \quad (26)$$

and get two samples each of size 50,000. We used 9 Gibbs steps within each block for *PA* and 5 Gibbs steps for *CSA* before the attempt for coalescence. The marginal distribution for each co-ordinate of the target distribution is just $\mathcal{N}(0, 2.1426)$. The left qq-plot in Fig.7 and the Kolmogorov-Smirnov (KS) tests (Table 1) on all univariate samples are reassuring as to the correctness of our algorithms.

Table 1: p-values for the KS test on two samples each of size 50,000 from (26) after applying *PA* and *CSA*. The test was performed separately for each of the three co-ordinates. The null hypothesis in all cases is that the draws are from $\mathcal{N}(0, 2.1426)$.

Co-ordinate	1st	2nd	3rd
PA	0.11	0.40	0.95
CSA	0.35	0.65	0.31

To make sure that the algorithms are correct even when the state space is truncated we sample 50,000 draws from the distribution:

$$TN_3(\mathbf{0}, Q^{-1})|_{\mathbf{a}}^{\mathbf{b}}, \text{ for } \mathbf{a}=(0, 0, 0), \mathbf{b}=(10, 10, 10), Q=(q_{ij}) \text{ with } q_{ii}=1, q_{ij}=-0.4 \text{ if } i \neq j \quad (27)$$

(5 and 3 Gibbs steps for *PA* and *CSA* respectively) and carry out similar error-checking tests, see the right qq-plot of Fig.7 and Table 2. This time the marginal distributions for each co-ordinate do not have a simple form so the results are compared with draws obtained from an *RS* that uses the corresponding untruncated density as envelope function. Sampling from unconstrained multivariate Gaussian densities is carried out via the Cholesky decomposition of the variance-covariance matrix, see for instance ch.XI of Devroye (1986).

We repeated the same testing procedure for different dimensions and truncation regions (chosen in a way that the benchmark rejection sampling is not too inefficient) and did not inspect any flaws.

7.2 Efficiency evaluation

We compare *PA* and *CSA* with *RS* that proposes from the corresponding untruncated density. We consider a best case scenario for *RS* with $\mathbf{a} = \{0\}^d$ and $\mathbf{b} = \{10\}^d$; a big proportion of the probability mass for the envelope density of *RS* is in the truncation region so the probability that a draw from this envelope density gets accepted is big compared to other cases. It is interesting to incorporate in this comparison different positive correlation levels. An easy way to achieve

Table 2: p-values for the KS test on three samples each of size 50,000 from (27) after applying *PA*, *CSA* and *RS*. The test was performed separately for each of the three co-ordinates and for all possible pairs of algorithms. The null hypothesis in all cases is that the observations are from the same distribution.

co-ordinate	PA-RS	PA-CSA	CSA-RS
1st	0.58	0.67	0.67
2nd	0.80	0.97	0.44
3rd	0.75	0.55	0.27

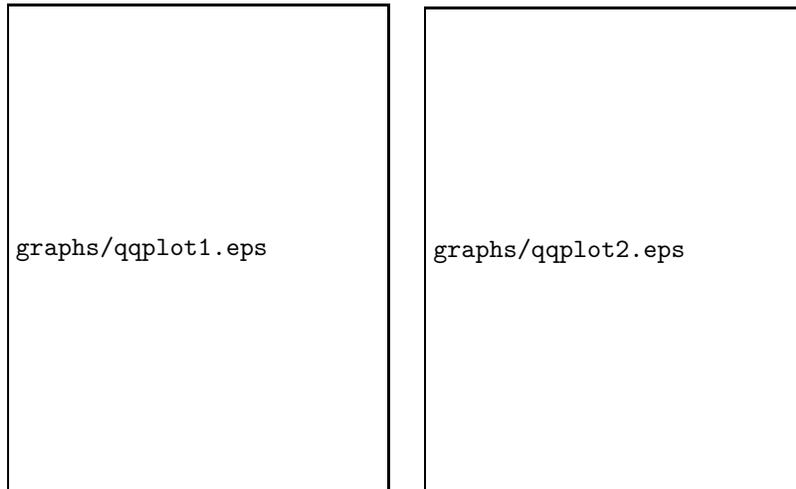


Figure 7: On the left, a qq-plot for the first co-ordinate of a sample of size 50,000 after implementing *PA* on (26). The observations have been divided with $\sqrt{2.1426}$ so that the estimated quantiles are comparable with those of the $\mathcal{N}(0,1)$. On the right, a qq-plot comparing the quantiles for the distribution of the first co-ordinate of a random vector distributed according to (27) as estimated from two samples of size 50,000 from *PA* and *CSA*.

that is by considering Stieltjes precision matrices $Q=(q_{ij})$ with $q_{ii} = 1$ and $q_{ij} = -c/(d-1)$ for some $0 < c < 1$ when $i \neq j$. This construction leads to appropriate symmetric, positive definite matrices. As c increases higher positive correlation levels are explored.

Table 3 shows the number of draws/sec after running the algorithms for different dimensions and different levels of positive correlation. The column of the positive correlations shows the value of c considered when constructing the precision matrix with the method described above. The table also gives in parentheses the number of Gibbs steps within a block (excluding the final coalescence Gibbs step) for each case of the *PA* and *CSA*. The number of Gibbs steps chosen for each algorithm is the one that maximised its efficiency in terms of the computational time to obtain a given number of samples. All the algorithms are written in C-language and were executed on an Athlon personal computer, running at 1500MHz in a Linux environment.

The cases considered in Table 3 are convenient not only for *RS* but also for *CSA*. *PA* can outperform *CSA* when truncations at more heterogeneous regions are considered. For example,

Table 3: Draws/sec for *RS* that proposes from the corresponding unconstrained density, *PA* and *CSA* when used on $TN_d(\mathbf{0}, Q^{-1})|_{\mathbf{b}}^{\mathbf{a}}$ for $\mathbf{a} = \{0\}^d$ and $\mathbf{b} = \{10\}^d$ for different d 's and levels of positive correlation. In parentheses the number of Gibbs steps within a block for *PA* and *CSA*.

		dimensionality				
		10	25	75	100	
correlation level	RS					
	PA					
	CSA					
	0.2		362	0	0	0
			7,712 (1)	2,421 (2)	435 (3)	264 (3)
			6,671 (1)	2,139 (1)	508 (2)	492 (2)
	0.5		1,173	0.22	0	0
			4,519 (3)	1,221 (5)	224 (7)	128 (8)
			4,563 (2)	1,453 (3)	337 (3)	208 (4)
	0.8		5,882	27	0	0
			1,266 (11)	154 (17)	4 (50)	1 (200)
			2,325 (6)	667 (7)	167 (8)	120 (8)

we applied *PA* and *CSA* to the 50-dimensional $TN_{50}(\mathbf{0}, Q^{-1})|_{\mathbf{b}}^{\mathbf{a}}$ for $\mathbf{a} = \{-40\}^{25} \times \{40\}^{25}$ and $\mathbf{b} = \{-20\}^{25} \times \{60\}^{25}$ and $Q = (q_{ij})$ with $q_{ii} = 1$ and $q_{ij} = -0.8/49$ for $i \neq j$. *CSA* is then inefficient because at the first Gibbs steps values $\Phi(x)$ from the cdf of the $\mathcal{N}(0, 1)$ distribution for $|x| > 38$ are usually requested (see the discussion at the end of Section 4). *PA* returns approximately 50 draws/sec and it needs only 7 Gibbs steps to deliver blocks with probability of being successful close to 0.8. The small number of Gibbs steps needed is due to the fact that for truncations distanced from the origin a huge proportion of the probability mass of the target Gaussian density is found near the boundary closest to the origin, so the Gibbs sampler is rapidly convergent.

8 Conclusions

We have introduced a flexible CFTP methodology for MCMC algorithms. We also devised a powerful coupling technique (the Monotone Independence Coupler) which can be applied in various settings. The algorithms we presented use the Ro-CFTP protocol and rely on the construction of blocks consisting of three different stages (forcing compactness, bridging the gap, and attempting for one-shot coalescence). The construction of our coupling block allows many options for extension and variation which might be appropriate in different examples. We have presented our methods in a fairly general way, and though applied only to the *TN* family of target densities, we hope we have managed to illustrate the general applicability of the techniques introduced.

We have found that implementation on Gibbs samplers with dimension up to 100 is well within the feasibility of the method, though we have not as yet carried out more extensive simulation studies on distributions obtained (for instance) as posterior distributions in Bayesian analysis.

The Read-Once CFTP idea circumvents the need for storing random seeds, and thus programming is generally easier and less prone to errors. The One-Shot idea spares time from attempting for coalescence when its probability is small. Note that the coalescence step, need not be monotone; assume that at the final step of a block of a Ro-CFTP algorithm it is possible

that we use the splitting technique, i.e. the kernel $P(x, dy)$ under consideration can split as $\rho \nu(dy) + (1 - \rho) Q(x, dy)$ for any x between some appropriately constructed bounding paths. With probability ρ all paths are projected onto a draw from ν and the block is successful whereas with probability $1 - \rho$ we just declare the block unsuccessful and simply update the coalescence path according to the residual kernel $Q(x, dy)$.

There is nothing in the general methodology which restricts us to the use of Gibbs samplers. Independence Couplers can be constructed in a natural way for general Metropolis-Hastings kernels (see Breyer and Roberts, 2001). We did however exploit, for the example with the TN s, a monotonicity property of the Gibbs sampler and this was critical for the efficiency of the bounding techniques. Ongoing work will investigate the scope of the method for more general cases.

Acknowledgements

The first author would like to acknowledge financial support from the Greek State Scholarships Foundation.

APPENDIX

Proof of Theorem 1:

Recall that $\mathcal{P}(x, dy) = P(\Phi(x) \in dy)$. Let $\nu(dy) := P(\Phi(x) \in dy | D)$ independent of x . The kernel \mathcal{P} can split as following:

$$\mathcal{P}(x, dy) = \rho \nu(dy) + (1 - \rho) Q(x, dy) \quad (28)$$

for $\rho = P(D)$ and $Q(x, dy) \equiv P(\Phi(x) \in dy | D^c)$ the residual kernel.

We will need some transition (probability) kernel operations; the product of two transition kernels P_1, P_2 is the transition kernel $P_1 P_2(x, dy) = \int_z P_1(x, dz) P_2(z, dy)$, the power of a transition kernel is defined inductively as $P^n = P^{n-1} P$ with $P^0(x, dy) \equiv \mathbb{I}_{\{x \in dy\}}$ and the product of a probability measure λ with a transition kernel P is the probability measure $\lambda P(dx) = \int_z \lambda(dz) P(z, dx)$.

Let $W^i = \Phi^{\tau_i - 1} \circ \dots \circ \Phi^{\tau_0 + 1} \circ \Phi^{\tau_0}(S)$, $i \geq 1$, be the random variable that refers to the position of the coalescence path one block-step before the $(i + 1)$ -th successful block and Δt_i be the number of (unsuccessful) blocks between this successful block and the immediately previously successful one. It is then true that:

$$P(W^i \in dw) = \sum_{j=0}^{\infty} P(\Delta t_i = j) P(W^i \in dw | \Delta t_i = j) = \sum_{j=0}^{\infty} (1 - \rho)^j \rho \nu Q^j(dw) \equiv \pi(dw) \quad (29)$$

The last and critical part of (29) is obtained algebraically after using (28) and that $\pi \mathcal{P} = \pi$, see Breyer and Roberts (2001) for details. The independence among the draws follows from the *regeneration* of the coalescence path after successful blocks, i.e. tours of the coalescence path at the time instances between a successful block and one block before the next successful one are i.i.d..

Proof of Theorem 2:

(i) It is just a case of the Independence Coupler.

(ii) Get $F_x \sim P(x, dy)$. Let $R_1 := \{x \in X : x \leq x^* \text{ and } F_x < Y\}$, $R_2 := \{x \in X : x^* \leq x \text{ and } F_x > Y\}$ and $A := X \cap (R_1 \cup R_2)^c = \{x \in X : x \leq x^* \text{ and } F_x \geq Y\} \cup \{x \in X : x^* \leq x \text{ and } F_x \leq Y\}$. Define also

$$G(x, Y) := \frac{p(x, Y)p(x^*, F_x)}{p(x, F_x)p(x^*, Y)}, \quad x \in X$$

Choose any $x_1, x_2 \in X$ with $x_1 \leq x_2$. To simplify the notation we write F_1, F_2 instead of F_{x_1}, F_{x_2} respectively. We examine the following cases:

Case 1: $x_1, x_2 \in A$.

If $x \in A$ then $G(x, Y) = \frac{p(x^*, F_x)}{p(x, F_x)} / \frac{p(x^*, Y)}{p(x, Y)} \geq 1$ from property (ii) of $p(x, y)$ given in Proposition 1, so $C \circ F_1 = C \circ F_2 = Y$.

Case 2: $x_1, x_2 \in R_1$.

We again exploit the monotonicity property of $p(x, y)$. Recall that $F_1 \leq F_2 < Y$.

$$\frac{G(x_2, Y)}{G(x_1, Y)} = \frac{\frac{p(x_2, Y)p(x^*, F_2)}{p(x_2, F_2)}}{\frac{p(x_1, Y)p(x^*, F_1)}{p(x_1, F_1)}} \geq \frac{\frac{p(x_2, F_2)p(x^*, F_2)}{p(x_2, F_2)}}{\frac{p(x_1, F_2)p(x^*, F_1)}{p(x_1, F_1)}} = \frac{p(x^*, F_2)p(x_1, F_1)}{p(x_1, F_2)p(x^*, F_1)} \geq \frac{p(x^*, F_1)p(x_1, F_1)}{p(x_1, F_1)p(x^*, F_1)} = 1$$

Since $G(x_2, Y) \geq G(x_1, Y)$ it is clear from (6) that if x_1 is mapped to Y ($U \leq G(x_1, Y)$) so is x_2 . This excludes, from the definition of R_1 , the only possibility of getting $C \circ F_1 > C \circ F_2$.

Case 3: $x_1, x_2 \in R_2$.

Exactly as in the previous case:

$$\frac{G(x_2, Y)}{G(x_1, Y)} = \frac{\frac{p(x_2, Y)p(x^*, F_2)}{p(x_2, F_2)}}{\frac{p(x_1, Y)p(x^*, F_1)}{p(x_1, F_1)}} \leq \frac{\frac{p(x_2, F_1)p(x^*, F_2)}{p(x_2, F_2)}}{\frac{p(x_1, F_1)p(x^*, F_1)}{p(x_1, F_1)}} = \frac{p(x_2, F_1)p(x^*, F_2)}{p(x^*, F_1)p(x_2, F_2)} \leq \frac{p(x_2, F_2)p(x^*, F_2)}{p(x^*, F_2)p(x_2, F_2)} = 1$$

Once again, after considering the definition of R_2 and that if x_2 accepts Y so does x_1 we get that $C \circ F_1 \leq C \circ F_2$.

From the definition of the sets A, R_1, R_2 and the result for the case when $x \in A$ it is straightforward that the only case that $x_1 \leq x_2$ can lead to $C \circ F_1 > C \circ F_2$ is when $x_1, x_2 \in R_1$ or $x_1, x_2 \in R_2$. For instance, if $x_1 \in R_1$ and $x_2 \in R_2$ then clearly $C \circ F_1 \leq Y \leq C \circ F_2$, if $x_1 \in R_1$ and $x_2 \in A$ then $C \circ F_1 \leq Y = C \circ F_2$, and so on. The preceding results demonstrate that even when $x_1, x_2 \in R_1$ or $x_1, x_2 \in R_2$ monotonicity, i.e. $C \circ F_1 \leq C \circ F_2$ for $x_1 \leq x_2$, is preserved.

The algorithm that draws from $TN_1(\mu, 1)|_\alpha^\beta$:

The method `tr_normal` (α, β, μ) which samples from the $TN_1(\mu, 1)|_\alpha^\beta$ is given as pseudo-algorithm in Fig.8. To avoid unnecessary details in the figure `REJ(f(x), g(x))` implies a draw from f after performing rejection sampling with g as the comparison density ($f \leq g$). We use $f(x)|_\gamma^\delta$ to represent $f(x)\mathbb{I}_{[\gamma, \delta]}(x)$ for a functional f and reals $\gamma < \delta$.

The algorithm incorporates four separate rejection sampling algorithms. The acceptance probability $r_{\alpha, \beta}$ for each case and its minimum value are given in the following table (to simplify the demonstration we have set $\mu = 0$ so the target density is proportional to $e^{-x^2/2}\mathbb{I}_{[\alpha, \beta]}(x)$):

```

tr_normal( $\alpha, \beta, \mu$ )
{
  set  $b := \beta - \mu$  and  $a := \alpha - \mu$ 
  if ( $a \geq 1/2$ )  $y = \text{REJ}(\exp(-x^2/2) |_{\alpha}^b, \exp(-ax + a^2/2) |_{\alpha}^b)$ 
  else if ( $b \leq -1/2$ )  $y = \text{REJ}(\exp(-x^2/2) |_{\alpha}^b, \exp(-bx + b^2/2) |_{\alpha}^b)$ 
  else if ( $b - a \leq 3/2$ )  $y = \text{REJ}(\exp(-x^2/2) |_{\alpha}^b, 1 |_{\alpha}^b)$ 
  else  $y = \text{REJ}(\exp(-x^2/2) |_{\alpha}^b, \exp(-x^2/2))$ 
  return  $y + \mu$ 
}

```

Figure 8: The method for sampling from $TN_1(\mu, 1) |_{\alpha}^{\beta}$. It incorporates four different rejection sampling algorithms. In all cases drawing from the envelope densities is straightforward.

envelope	called on	$r_{\alpha, \beta}$	inf (attained at)
$e^{-\alpha x + a^2/2} \mathbb{I}_{[\alpha, \beta]}(x)$	if $\alpha \geq 1/2$	$\frac{\alpha \sqrt{2\pi} (\Phi(\beta) - \Phi(\alpha))}{\exp(-\alpha^2/2) - \exp(a^2/2 - \alpha\beta)}$	0.44^a $(\alpha, \beta) = (1/2, \infty)$
$e^{-\beta x + \beta^2/2} \mathbb{I}_{[\alpha, \beta]}(x)$	else if $\beta \leq -1/2$	$\frac{(-\beta) \sqrt{2\pi} (\Phi(\beta) - \Phi(\alpha))}{\exp(-\beta^2/2) - \exp(-\alpha\beta + \beta^2/2)}$	0.44^b $(\alpha, \beta) = (-\infty, -1/2)$
$\mathbb{I}_{[\alpha, \beta]}(x)$	else if $(\beta - \alpha) \leq 3/2$	$\sqrt{2\pi} \frac{\Phi(\beta) - \Phi(\alpha)}{\beta - \alpha}$	0.47^c $(\alpha, \beta) = (1/2, 2)$
$e^{-x^2/2}$	else	$\Phi(\beta) - \Phi(\alpha)$	0.29^d $(\alpha, \beta) = (1/2, 2)$

^a r is $\downarrow \beta$ (decreasing in β) for constant α and $\uparrow \alpha$ (increasing in α) for constant $\beta - \alpha$.

^b r is $\uparrow \alpha$ for constant β and $\downarrow \beta$ for constant $\beta - \alpha$.

^cFor constant $\beta - \alpha \equiv c$, r is $\uparrow \beta \in (-1/2, c/2]$ and $\downarrow \beta \in [c/2, 1/2 + c]$, so $r_{\alpha, \beta} \geq r_{1/2, 1/2+c}$ which is $\downarrow c$, so

$r_{\alpha, \beta} \geq r_{1/2, 2}$.

^d $r_{\alpha, \beta} \geq r_{\alpha, \alpha+3/2}$ which is $\downarrow \alpha$.

The methodology is close to the one proposed in Geweke (1991) except for avoiding using the cdf of the $\mathcal{N}(0, 1)$ distribution when choosing among available comparison functions. That guarantees the unrestricted applicability of the resulted algorithm. In Robert (1995) a slightly different methodology is proposed which yields better acceptance probabilities for the proposed variates but as already noted in Geweke (1991) the increased algebraic structure of this algorithm renders it equivalent to ours in terms of computational time.

Proof of Proposition 2:

Certainly, $\lambda_V > 0$ since V is positive definite. Also, there exists an orthogonal matrix $T \in \mathbb{R}^{d \times d}$ ($T^\top T = I$) such that $T^\top \Lambda T = V$ where $\Lambda = (\lambda_{ij})$ is a diagonal matrix with diagonal elements the eigenvalues of V . Then, for $\mathbf{y} = T\mathbf{x}$:

$$\mathbf{x}^\top V \mathbf{x} = (T\mathbf{x})^\top \Lambda (T\mathbf{x}) = \sum_{i=1}^d \lambda_{ii} y_i^2 \geq \lambda_V \sum_{i=1}^d y_i^2 = \lambda_V \mathbf{y}^\top \mathbf{y} = \lambda_V \mathbf{x}^\top \mathbf{x}$$

References

- Arnold, B., Castillo, E., and Sarabia, J. (1999). *Conditional Specification of Statistical Models*. Springer.
- Arnold, B. C. and Strauss, D. (1988). Bivariate distributions with exponential conditionals. *J. Amer. Statist. Assoc.*, 83(402):522–527.
- Breyer, L. and Roberts, G. (2000). Some multi-step coupling constructions for Markov chains. Technical report, Lancaster University.
- Breyer, L. and Roberts, G. (2001). Catalytic perfect simulation. *Methodology and Computing in Applied Probability*, 3(2):161–177.
- Devroye, L. (1986). *Non-uniform Random Variate Generation*. Springer-Verlag.
- Djuric, P. M. and Kotecha, J. H. (1999). Gibbs sampling approach for generation of truncated multivariate gaussian random variables. *Proceedings of IEEE ICASSP-99*, 3:1757–1760.
- Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *J. Amer. Statist. Assoc.*, 85(410):398–409.
- Geweke, J. (1991). Efficient simulation from the multivariate normal and Student- t distributions subject to linear constraints. In Keramidas, E. M., editor, *Computing Science and Statistics. Proceedings of the 23rd Symposium on the Interface*, pages 571–578. Interface Foundation of North America.
- Hajivassiliou, V., McFadden, D., and Ruud, P. (1996). Simulation of multivariate normal rectangle probabilities and their derivatives: Theoretical and computational results. *Journal of Econometrics*, 72:85–134.
- Hajivassiliou, V. A. and McFadden, D. L. (1998). The method of simulated scores for the estimation of LDV models. *Econometrica*, 66:863–896.
- Karlin, S. (1968). *Total positivity. Vol. I*. Stanford University Press, Stanford, Calif.
- Møller, J. (1999). Perfect simulation of conditionally specified models. *Journal of the Royal Statistical Society B*, 61(1):251–264.
- Murdoch, D. and Green, P. (1998). Exact sampling from a continuous state space. *Scandinavian Journal of Statistics*, 25(3):483–502.
- Philippe, A. and Robert, C. P. (2003). Perfect simulation of positive gaussian distributions. *Statistics and Computing*, 13:179–186.
- Propp, J. and Wilson, D. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9:223–252.
- Robert, C. (1995). Simulation of truncated normal variables. *Statistics and Computing*, 5:121–125.
- Roberts, G. and Rosenthal, J. (2002). One-Shot coupling for certain stochastic recursive sequences. *Stochastic Processes and their Applications*, 99:195–208.
- Roberts, G. and Tweedie, R. (2005). Understanding MCMC. To be published.

- Rodriguez-Yam, G. A., Davis, R. A., and Scharf, L. L. (2002). A bayesian model and Gibbs sampler for hyperspectral imaging. *Proceedings 2002 IEEE Sensor Array and Multichannel Signal Processing Workshop*, pages 105–109.
- Rosenthal, J. S. (1995). Minorization conditions and convergence rates for Markov chain Monte Carlo. *J. Amer. Statist. Assoc.*, 90(430):558–566.
- Thorisson, H. (2000). *Coupling, Stationarity and Renegeration*. Springer.
- Troughton, P. T. and Godsill, S. (1999). MCMC methods for restoration of quantised time series. *Proceedings of the IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, 2:447–451.
- Varga, R. S. (1962). *Matrix Iterative Analysis*. Prentice-Hall Inc.
- Wilson, D. (2000a). Layered multishift coupling for use in perfect sampling algorithms (with a primer on CFTP). In Madras, N., editor, *Monte Carlo Methods*, volume 26 of *Fields Institute Communications*, pages 141–176. American Mathematical Society.
- Wilson, D. B. (2000b). How to couple from the past using a read-once source of randomness. *Random Structures and Algorithms*, 16(1):85–113.