An Auto-Validating, Trans-Dimensional, Universal Rejection Sampler for Locally Lipschitz Arithmetical Expressions[*]

Raazesh Sainudiin[†]
Laboratory for Mathematical Statistical Experiments, Christchurch Centre, and Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand
r.sainudiin@math.canterbury.ac.nz

Thomas York
Department of Biological Statistics and Computational Biology, Cornell University, Ithaca, U.S.A.
tly2@cornell.edu

Abstract

The sample space of a trans-dimensional random vector is a union of spaces with different dimensions. We introduce a trans-dimensional extension of the rejection sampler of von Neumann. Our construction of the rejection sampler is based on interval analysis and provides a universal method that is capable of producing independent and identically distributed (IID) samples from a large class of trans-dimensional target densities with locally Lipschitz arithmetical expressions. We illustrate the efficiency of the sampler by theory and by examples in up to ten dimensions. Our sampler is immune to the 'pathologies' of some infamous densities that were previously considered unsamplable and can rigorously draw IID trans-dimensional posterior samples from small binomial partition models and phylogenetic tree spaces.

Keywords: interval analysis, perfect trans-dimensional sampling, computer-aided proofs in statistics
AMS subject classifications: 65C05,65C60,65G40,65C99,65G20

1 Introduction

Obtaining independent and identical (IID) samples or realizations from a random vector T with probability density function $f^{\cdot}$, denoted $T \sim f^{\cdot}$, is a basic problem in

[*] Submitted: September 30, 2011.Revised: January 30, 2013 and February 21, 2013; Accepted: April 3, 2013.

[†] Corresponding Author

computational statistics. The density $f^{\cdot}(t) : \mathbb{T} \subset \mathbb{R}^d \to [0, \infty)$ allows one to obtain $\mathbb{P}(T \in B) = \int_B f^{\cdot}(t)dt$, the probability that T belongs to any Borel set B. The density is absolutely continuous with respect to λ^d, the product of d Lebesgue measures, i.e., $f^{\cdot} \ll \lambda^d$, and integrates to 1, i.e., $\int_{\mathbb{T}} f^{\cdot}(t)dt = 1$. A *sampler* is a randomized algorithm that transforms independent and identically distributed (IID) samples from M, the uniformly distributed random variable on the unit interval, to those from the desired random object, say the random vector T with density $f^{\cdot}$.

In Bayesian estimation, we want to draw IID samples from a target posterior density $f^{\cdot}$ and more generally, in multivariate simulation, we want to draw IID samples from a random vector T with probability density $f^{\cdot}$. These samples allow insights into the nature of the random vector itself. They are often used to estimate an integral of interest about the random vector, say, $\mathbb{E}_{f^{\cdot}}(h(T)) := \int_{\mathbb{T}} h(t)f^{\cdot}(t)dt$, where $h(t) : \mathbb{T} \to \mathbb{R}$ is bounded and $\mathbb{E}_{f^{\cdot}}(h^2(T)) < \infty$, using the estimator $\widehat{h}_n = n^{-1}\sum_{i=1}^n h(t_i)$, where $t_1, t_2, \ldots, t_n$ are IID samples from T with density $f^{\cdot}$. For example, such integrals of interest can be the posterior mean given by $\int_{\mathbb{T}} tf^{\cdot}(t)dt$ with $h(t) = t$, or the probability of an event A given by $\mathbb{P}(A) := \int_{\mathbb{T}} \mathbb{1}_A(t)f^{\cdot}(t)dt$, with $h(t) = \mathbb{1}_A(t)$, where $\mathbb{1}_A(t)$ equals 1 if $t \in A$ and 0 otherwise. Due to the strong law of large numbers our estimator $\widehat{h}_n$ converges to the desired $\mathbb{E}_{f^{\cdot}}(h(T))$ with probability 1 as the number of samples n approaches infinity. Furthermore, the condition $\mathbb{E}_{f^{\cdot}}(h^2(T)) < \infty$ ensures the asymptotic normality of our estimator due to the central limit theorem and provides a straightforward calculation of a confidence interval.

The sample space of a trans-dimensional random vector is the union of two or more spaces of different dimensions. A more challenging problem is to simulate IID samples in a trans-dimensional setting where samples can come from spaces with different dimensions. In some applications such as Bayesian model selection, we are interested in such trans-dimensional simulation. Suppose we have a finite set of probability models labelled by the set $\mathbb{K} = \{0, 1, \ldots, \ell-1, \ell, \ell+1, \ldots, i\}$. For each $k \in \mathbb{K}$, let

$$^kT \sim {}^kf^{\cdot}({}^kt) : {}^k\mathbb{T} \to \mathbb{R}, {}^k\mathbb{T} \subseteq \mathbb{R}^{d_k}, \quad d_k \geq 1 \ .$$

We say the *labelled point* ^kt belonging to the *labelled sample space* $^k\mathbb{T}$ is a realisation of the *labelled random vector* ^kT of dimension d_k that is distributed according to the *labelled density* $^kf^{\cdot}$. This is called a trans-dimensional setting because the model dimensions $d_0, d_1, \ldots, d_i$ may not be the same.

We are interested in simulating IID samples from the *randomly labelled random vector* ^KT with the possibly trans-dimensional density $^{\mathbb{K}}f^{\cdot}({}^kt) : {}^{\mathbb{K}}\mathbb{T} \to \mathbb{R}$. The domain is given by the union of labelled sample spaces as

$$^{\mathbb{K}}\mathbb{T} := {}^0\mathbb{T} \cup {}^1\mathbb{T} \cup \cdots \cup {}^{\ell-1}\mathbb{T} \cup {}^{\ell}\mathbb{T} \cup {}^{\ell+1}\mathbb{T} \cup \cdots \cup {}^i\mathbb{T} \ , \tag{1}$$

and the density of ^KT is obtained by normalizing as

$$^KT \sim {}^{\mathbb{K}}f^{\cdot}({}^kt) := \frac{{}^{\mathbb{K}}f({}^kt)}{N_{{}^{\mathbb{K}}f}} \ , \tag{2}$$

where,

$$
\begin{aligned}
{}^{\mathbb{K}}\!f({}^{k}t) \quad &:= \quad \sum_{\ell=0}^{i} {}^{\ell}\!f({}^{k}t) \mathbb{1}_{{}^{\ell}\mathbb{T}}({}^{k}t) := \begin{cases} {}^{0}\!f({}^{k}t) & \text{if } k=0 \\ {}^{1}\!f({}^{k}t) & \text{if } k=1 \\ \vdots & \\ {}^{\ell-1}\!f({}^{k}t) & \text{if } k=\ell-1 \\ {}^{\ell}\!f({}^{k}t) & \text{if } k=\ell \\ {}^{\ell+1}\!f({}^{k}t) & \text{if } k=\ell+1 \\ \vdots & \\ {}^{i}\!f({}^{k}t) & \text{if } k=i \end{cases} , \qquad (3) \\
N_{\mathbb{K}_f} \quad &:= \quad \sum_{k=0}^{i} \int_{{}^{k}\mathbb{T}} {}^{k}\!f({}^{k}t) \; d({}^{k}t) \ . \qquad (4)
\end{aligned}
$$

Note that the sum in (3) is really a convenient notation for uniting the target densities over the labelled spaces in ${}^{\mathbb{K}}\mathbb{T}$ as shown by the i cases on the right hand side of (3) and one need not have ${}^{\ell}\!f({}^{k}t)$ defined when $\ell \neq k$. Here, ${}^{k}\!f({}^{k}t)$ is absolutely continuous with respect to λ^{d_k}, the product of d_k many Lebesgue measures, i.e., ${}^{k}\!f({}^{k}t) \ll \lambda^{d_k}$, and ${}^{k}\mathbb{T} \subseteq \mathbb{R}^{d_k}$. When $|\{d_k : k \in \mathbb{K}\}| > 1$ the randomly labelled random vector ${}^{K}T$ becomes a *trans-dimensional random vector*. The challenge is to draw n possibly trans-dimensional samples ${}^{k_1}t_1, {}^{k_2}t_2, \ldots, {}^{k_n}t_n$ without any knowledge of the normalising constant $N_{\mathbb{K}_f}$ since the target density ${}^{\mathbb{K}}\!f^{\cdot}$ is typically only known up to this normalising constant as the target shape ${}^{\mathbb{K}}\!f({}^{k}t)$.

Example 1 (A trans-dimensional uniform random vector)**.** *Let* $\mathbb{K} = \{0, 1\}$ *with labelled domains* ${}^{0}\mathbb{T} = \{{}^{0}t = ({}^{0}t_1, {}^{0}t_2) \in [0,1]^2 : {}^{0}t_1 = {}^{0}t_2\}$ *and* ${}^{1}\mathbb{T} = \{{}^{1}t = ({}^{1}t_1, {}^{1}t_2) \in [0,1]^2\}$*. Let the model-labelled densities be* ${}^{0}\!f({}^{k}t) = \mathbb{1}_{{}^{0}\mathbb{T}}({}^{k}t) \ll \lambda^1$ *and* ${}^{1}\!f({}^{k}t) = \mathbb{1}_{{}^{1}\mathbb{T}}({}^{k}t) \ll \lambda^2$ *corresponding to the uniform densities over* ${}^{0}\mathbb{T}$ *and* ${}^{1}\mathbb{T}$*, respectively. Thus,* $N_{\mathbb{K}_f} = 1 + 1 = 2$*. Our trans-dimensional uniform random vector that is equally likely to be realised uniformly at random on* ${}^{0}\mathbb{T}$ *or* ${}^{1}\mathbb{T}$ *has density:*

$$
{}^{\mathbb{K}}\!f^{\cdot}({}^{k}t) = \frac{1}{2}\left(\mathbb{1}_{{}^{0}\mathbb{T}}({}^{k}t) + \mathbb{1}_{{}^{1}\mathbb{T}}({}^{k}t)\right) \ .
$$

The algorithm to simulate samples from this randomly labelled random vector ${}^{K}T$ *is simple: if* m_1 *is less that* $1/2$ *then return* m_2 *else return* (m_2, m_3) *where* m_1, m_2, m_3 *are IID samples from the Uniform distribution on* $[0,1]$*.*

In an inferential context, it is convenient to let ${}^{k}\!f({}^{k}t)$, the shape of the target density for each model k, have three components: (i) p_k, the prior probability, such that, $\sum_{k=0}^{i} p_k = 1$, (ii) the likelihood ${}^{k}L({}^{k}t) \propto \mathbb{P}(\text{data}|{}^{k}t) : {}^{k}\mathbb{T} \to \mathbb{R}$ and (iii) the prior density ${}^{k}q({}^{k}t) : {}^{k}\mathbb{T} \to \mathbb{R}$, i.e.,

$$
{}^{k}\!f({}^{k}t) := p_k \; {}^{k}L({}^{k}t) \; {}^{k}q({}^{k}t) \ .
$$

Example 2 (Same or different coins)**.** *Suppose I give you a coin. You observe the outcomes of* n_1 *independent and identical tosses with it. You return the coin back to me. Now I give you a possibly different coin. Your task is to determine if the second coin is the same (or has the same bias) as the first coin by tossing the second coin as before* n_2 *times.*

Let the labelled domains for the two models be ${}^0\mathbb{T} = \{{}^0t = ({}^0t_1, {}^0t_2) \in [0,1]^2 : {}^0t_1 = {}^0t_2\}$ *and* ${}^1\mathbb{T} = \{{}^1t = ({}^1t_1, {}^1t_2) \in [0,1]^2\}$*. Let the uniform prior densities be* ${}^0q({}^kt) = 1\!\!1_{{}^0\mathbb{T}}({}^kt) \ll \lambda^1$ *and* ${}^1q({}^kt) = 1\!\!1_{{}^1\mathbb{T}}({}^kt) \ll \lambda^2$*. Let the model priors be* p_0 *and* p_1*. Let the number of Heads in the first and second sequence of tosses be* x_1 *and* x_2*, respectively. Finally, the trans-dimensional posterior density* ${}^{\mathbb{K}}f^{\cdot}({}^kt)$ *is proportional to:*

$$\begin{aligned}{}^{\mathbb{K}}f({}^kt) &= p_0 \binom{n_1}{x_1} ({}^0t_1)^{x_1+x_2} \binom{n_2}{x_2} (1-{}^0t_1)^{n_1+n_2-x_1-x_2} 1\!\!1_{{}^0\mathbb{T}}({}^kt) \\ &+ p_1 \left(\binom{n_1}{x_1} ({}^1t_1)^{x_1} (1-{}^1t_1)^{n_1-x_1} \binom{n_2}{x_2} ({}^1t_2)^{x_2} (1-{}^1t_2)^{n_2-x_2} 1\!\!1_{{}^1\mathbb{T}}({}^kt) \right) \ .\end{aligned}$$

It is more difficult to produce IID samples from the target density in Example 2 when compared to that from Example 1.

Enclosure methods that rely on machine interval arithmetic — validated computer arithmetic that encloses or bounds all numerical errors — have become an important tool in computer-aided proofs in analysis. Some examples where these methods have been applied include proofs of the Feigenbaum conjectures [31], the double bubble conjecture [21], the existence of the Lorenz attractor [50] and the Kepler conjecture [19]. In this paper, we employ auto-validating enclosure methods via interval analysis to the fundamental computational statistical problem of simulating independent samples from a *trans-dimensional random vector* – a randomly labelled random vector whose probability density with respect to Lebesgue measure is given by a locally Lipschitz arithmetical expression over a set of finite dimensional Euclidean spaces with possibly different dimensions.

1.1 Classical Samplers

The classical methods for simulating IID samples from a random variable T with density $f^{\cdot}(t)$ and distribution function $F^{\cdot}(t) := \int_{\{s:s\leq t\}} f^{\cdot}(s)ds$, are the *Inversion Sampler* (InS), the *Rejection Sampler* (RS) of von Neumann [51] and their variants. These samplers transform IID samples from M, the uniformly distributed random variable on the unit interval, to those from the desired T.

The algorithm for InS is to return $F^{\cdot[-1]}(m)$, where m is a sample from M and $F^{\cdot[-1]}(u) : [0,1] \to \mathbb{R}$ is the inverse of the distribution function of T. Thus, if one knows the inverse of the distribution function for a univariate random variable then InS is the simplest way to produce a sample from T. For most random variables, it is difficult to compute $F^{\cdot}$ and often the target density $f^{\cdot}(t)$ is only known up to a normalising constant. Furthermore, the idea of inversion sampling cannot be easily extended to multivariate random vectors unless their density is given by a product of univariate densities with invertible distribution functions.

The algorithm for RS is also relatively simple and generalizes to a multivariate setting. Unlike InS, RS can produce IID samples from the target density $f^{\cdot}(t) := f(t)/(N_f)$ by only evaluating the *target shape* $f(t)$ — without knowing the normalising constant $N_f := \int_{\mathbb{T}} f(t)dt$. Briefly, the idea behind RS is as follows: produce a point uniformly distributed in the $(d+1)$-dimensional region under an envelope function that is strictly greater than or equal to the target shape and if this point is below the target shape then accept its first d coordinates in $\mathbb{T}$ as a sample from T, otherwise reject it and try again. However, the limiting step in RS is the construction of an envelope function $\widehat{g}(t)$ that is not only greater than the target shape $f(t) := N_f f^{\cdot}(t)$ at every $t \in \mathbb{T} \subseteq \mathbb{R}^d$, but also easy to normalise and draw samples from. Moreover, a practical and efficient envelope function has to be as close to the target shape as possible from above. RS in a more general setting is detailed in Algorithm 1.

One can employ well-known computational statistical techniques, such as, importance sampling, residual sampling, squeeze principle and alias method to improve the efficiency of our exact samplers. Since these techniques for improving efficiency are not the main focus of this paper we refer the reader to [4] for an introduction to these methods.

InS and RS are said to be *exact samplers* because they produce IID samples from the desired target. For several targets of interest it is difficult to construct an exact sampler. Another class of Monte Carlo methods can produce dependent samples from a Markov chain whose stationary distribution is the target density of interest. Such samplers are not exact because they do not produce IID samples. They include the *Metropolis-Hastings Sampler* [38, 22] and the *Gibbs Sampler* [13, 10] among others, and are collectively termed as Monte Carlo Markov chain (MCMC) methods. The reversible-jump MCMC sampler of Green [18] is capable of producing dependent trans-dimensional samples.

These dependent samplers are easier to construct and implement in multivariate settings when compared to exact samplers. However they are only asymptotically valid (as the number of samples go to infinity) and it is nontrivial to guarantee that an MCMC algorithm has converged to its stationary distribution [27]. Such guarantees are necessary to produce valid confidence intervals for $\mathbb{E}_{f^{\cdot}}(h(T))$ based on samples from $f^{\cdot}$. For specific targets that form the stationary distribution of suitably well-structured Markov chains, sophisticated coupling techniques [41] can be used to efficiently obtain IID samples. In general with MCMC one is at the mercy of heuristic convergence diagnostics [12] to determine the burn-in period (how long should one run the Markov chain to ensure that the chain has converged to the stationary distribution) and thin-out times (how to sub-sample the states visited by the chain to reduce the dependence between consecutive samples) and therefore cannot guarantee IID samples from them. Thus, we return to the rejection sampler, also known as the fundamental theorem of simulation, and develop universal methods that extend the class of densities to which it can be applied.

Next we review some samplers in the literature that are closer to our approach in terms of needing to bound or enclose the target densities. *Universal* or *automatic* or *black-box* samplers can be used to generate IID samples from any density in a restricted class. One of the earliest such samplers using *transformed density rejection* was due to Devroye [4, Ch. VII]. A density is said to be logconcave or $\mathcal{T}$-concave if it is concave after being transformed by the logarithm or the function $\mathcal{T}$, respectively. Adaptive rejection sampling is a universal sampler to obtain IID samples from univariate logconcave [14, 16] and $\mathcal{T}$-concave densities [24] and have been recently extended to a larger class of univariate densities [17, 37]. These methods can be used to construct a universal algorithm that is applicable to a large class of unimodal distributions, including the normal, beta, gamma, and t-distribution. A subsequent generalisation through a Metropolis step [15] produced dependent samples from univariate non-logconcave targets.

Multivariate universal extensions of the rejection sampler was successful in restricted classes; orthounimodal [5], bivariate $\mathcal{T}$-concave [25] and multivariate $\mathcal{T}$-concave [32] densities. More recently, Martino and Míguez [36, 37] give a generalised adaptive rejection sampling algorithm for a richer class of univariate densities and use it to produce dependent samples from multivariate targets by Gibbs sampling one dimension at a time.

These universal samplers have the following limitations. Multivariate extensions for less restrictive classes of densities are typically via dependent samplers and therefore rely on heuristic diagnostics to assess convergence to the desired stationary distribution. Furthermore, proposals constructed for non-log-concave conditional densities from finitely many points via usual floating-point arithmetic cannot guarantee that the density has not soared between the sampled points. Finally, no universal sampler is available even for the simplest class of trans-dimensional densities. Thus, none of the existing exact universal samplers is capable of producing independent and identical samples from more general classes of possibly trans-dimensional multivariate target densities.

1.2 Our New Approach

Any sampler which uses conventional floating-point arithmetic operates under two `practical` assumptions outlined by Devroye [4, p. 1–2]:

- `A1` exact real operations are possible on a conventional computer with finite memory and
- `A2` IID samples are available from M, the uniform random variable on $[0, 1]$.

Our method relaxes assumption `A1` because it employs interval arithmetic to produce rigorous enclosures of the range of the target shape over each interval vector or box in an adaptive partition of the domain. The interval extended target shape maps boxes in the partition to intervals in $\mathbb{R}$. This image interval provides an upper bound for the global maximum and a lower bound for the global minimum of the target over each element of the partition. We use this information to construct an envelope as a piecewise constant function over the partition. Using the Alias method of Walker [52] we efficiently propose samples from this normalised step-function envelope for von Neumann rejection sampling. By using the notion of model-labelled points and boxes we extend rejection sampling to a trans-dimensional density over finitely many model-labelled domains of possibly distinct finitely many dimensions.

Unlike existing exact samplers, the auto-validating construction of our rejection sampler through interval methods gives an enclosure of the target shape over the entire real continuum in any box of the domain with machine-representable bounds. These enclosures rigorously account for all sources of numerical errors [30, 20] and thereby guarantee IID samples from the desired target. Moreover, the target is allowed to be multivariate and/or non-log-concave and/or trans-dimensional with possibly 'pathological' behaviour, as long as its arithmetical expression has a well-defined interval extension.

We call our method auto-validating because we employ interval methods to rigorously construct the envelope for a large class of target densities, including trans-dimensional densities. The method was already described in a more rudimentary form in [42, 44] and applied to Bayesian phylogenetic estimation in [45]. This sampler is referred to as the Moore rejection sampler (MRS) in honour of Ramon E. Moore who was one of the influential founders of interval analysis [39]. This is the first paper at the interface of computer-aided proofs in analysis and Monte Carlo methods. It unifies the fundamental theorem of simulation, i.e., the rejection sampler of von Neumann [51], with the fundamental theorems of interval analysis to produce the first exact trans-dimensional sampler for any locally Lipschitz target density. Our sampler can produce IID samples from several densities, including the multivariate witch's hat (see Section 5.5) that was previously thought to be unsamplable [29].

1.3 Plan of the Paper

There is currently little overlap between researchers in computational statistics and reliable computing. This paper is an attempt to increase communication between these two communities of researchers. However, we have written it from the perspective of introducing a computational statistician to the core elements in interval analysis that are needed to appreciate the Moore rejection sampler. Care has also been taken to minimize the number of new concepts from computational statistics that a researcher from reliable computing needs to know in order to appreciate the basic ideas from interval analysis that make the Moore rejection sampler work.

The rest of the paper is organised as follows. Trans-dimensional von Neumann Rejection Sampler (TRS) is formulated in Section 2. We give a brief and self-contained introduction to rudimentary interval analysis for the sake of researchers from computational statistics who are likely to be unfamiliar with these notions in Section 3. This Section can be skipped by those who are familiar with interval analysis. We construct our sampler in Section 4. Examples demonstrating the robustness and efficiency of the sampler are discussed in Section 5. We conclude in Section 6.

2 Trans-dimensional Rejection Sampler (TRS)

The rejection sampler (RS) of von Neumann [51] in its original formulation is capable of drawing independent samples from a target random variable or random vector T with density $f^{\cdot}(t) := f(t)/N_f$ but with a fixed dimension n, i.e., $T \sim f^{\cdot}$ and $t \in \mathbb{T} \subseteq \mathbb{R}^n$.

It is more difficult to produce IID samples from the target density in Example 2 when compared to that from Example 1. The trans-dimensional extension of the von Neumann rejection sampler (TRS) can produce samples from ${}^{K}T \sim {}^{\mathbb{K}}f^{\cdot}$ according to Algorithm 1 when provided with (i) a fundamental sampler that can produce independent samples from the Uniform$[0,1]$ random variable M with density given by the indicator function $1\!\!1_{[0,1]}(m) : \mathbb{R} \to \mathbb{R}$, (ii) a target shape ${}^{\mathbb{K}}f({}^{k}t) : {}^{\mathbb{K}}\mathbb{T} \to \mathbb{R}$, (iii) an envelope function $\widehat{{}^{\mathbb{K}}g}({}^{k}t) : {}^{\mathbb{K}}\mathbb{T} \to \mathbb{R}$, such that, for each ${}^{k}t \in {}^{\mathbb{K}}\mathbb{T}$:

$$\widehat{{}^{\mathbb{K}}g}({}^{k}t) := \sum_{\ell \in \mathbb{K}} \widehat{{}^{\ell}g}({}^{k}t)\, 1\!\!1_{{}^{\ell}\mathbb{T}}({}^{k}t) \geq {}^{\mathbb{K}}f({}^{k}t) := \sum_{\ell \in \mathbb{K}} {}^{\ell}f({}^{k}t)\, 1\!\!1_{{}^{\ell}\mathbb{T}}({}^{k}t) \quad , \tag{5}$$

(iv) a normalising constant $N_{\widehat{{}^{\mathbb{K}}g}} := \sum_{k \in \mathbb{K}} \int_{{}^{k}\mathbb{T}} \widehat{{}^{\mathbb{K}}g}({}^{k}t)\ d({}^{k}t)$, (v) a "proposal" density over ${}^{\mathbb{K}}\mathbb{T}$:

$${}^{\mathbb{K}}g({}^{k}t) := \sum_{\ell \in \mathbb{K}} {}^{\ell}g({}^{k}t)\, 1\!\!1_{{}^{\ell}\mathbb{T}}({}^{k}t) = \frac{\widehat{{}^{\mathbb{K}}g}({}^{k}t)}{N_{\widehat{{}^{\mathbb{K}}g}}} = \frac{\sum_{\ell \in \mathbb{K}} \widehat{{}^{\ell}g}({}^{k}t)\, 1\!\!1_{{}^{\ell}\mathbb{T}}({}^{k}t)}{N_{\widehat{{}^{\mathbb{K}}g}}} \quad , \tag{6}$$

from which independent samples can be drawn and finally (vi) ${}^{\mathbb{K}}f({}^{k}t)$ and $\widehat{{}^{\mathbb{K}}g}({}^{k}t)$ must be computable for any ${}^{k}t \in {}^{\mathbb{K}}\mathbb{T}$.

Theorem 1 (Trans-dimensional von Neumann RS)**.** *If the randomly labelled, random vector ${}^{K}T$ is generated according to Algorithm 1 with the envelope function $\widehat{{}^{\mathbb{K}}g}$ satisfying* (5) *and the corresponding proposal density ${}^{\mathbb{K}}g$ of* (6)*, then ${}^{K}T$ is distributed according to the possibly trans-dimensional target density ${}^{\mathbb{K}}f^{\cdot}$.*

Algorithm 1: Trans-dimensional von Neumann Rejection Sampler (TRS)

```
input     : (1) algorithm for computing ${}^{\mathbb{K}}f$;
            (2) samplers for ${}^{K}V \sim {}^{\mathbb{K}}g$ and $M \sim \mathbb{1}_{[0,1]}$;
            (3) algorithm for computing $\widehat{{}^{\mathbb{K}}g}$;
            (4) integer MaxTrials;
output    : (1) possibly one sample ${}^{k}t$ from ${}^{K}T \sim {}^{\mathbb{K}}f^{\cdot}$
            (2) number of trials Trials
initialize: Trials $\leftarrow 0$; Success $\leftarrow$ false; ${}^{k}t \leftarrow \emptyset$;
repeat                                         // propose at most MaxTrials times until acceptance
    ${}^{k}v \leftarrow$ sample(${}^{\mathbb{K}}g$) ;          // draw a sample ${}^{k}v$ from RV ${}^{K}V$ with density ${}^{\mathbb{K}}g$
    $u \leftarrow \widehat{{}^{\mathbb{K}}g}({}^{k}v)$ sample($\mathbb{1}_{[0,1]}$);   // sample $u$ from RV $U$ with density $\mathbb{1}_{[0,\widehat{{}^{\mathbb{K}}g}({}^{k}v)]}$
    if $u \leq {}^{k}f({}^{k}v)$ then                 // accept the proposed ${}^{k}v$ and flag Success
        ${}^{k}t \leftarrow {}^{k}v$; Success $\leftarrow$ true
    end
    Trials $\leftarrow$ Trials $+1$ ;                // track the number of proposal trials so far
until Trials $\leq$ MaxTrials or Success = true;
return ${}^{k}t$ and Trials
```

Proof. Without loss of generality, let ${}^{\mathbb{K}}\mathbb{T} := \left\{{}^{0}\mathbb{T}, {}^{1}\mathbb{T}, \ldots, {}^{i}\mathbb{T}\right\}$ be the labelled domain of the target density ${}^{\mathbb{K}}f^{\cdot}$ with components ${}^{0}f \ll \lambda^{d_0}, {}^{1}f \ll \lambda^{d_1}, \ldots, {}^{i}f \ll \lambda^{d_i}$ be contained in ${}^{\mathbb{K}}\mathbb{R} := \{\mathbb{R}^{d_0}, \mathbb{R}^{d_1} \ldots, \mathbb{R}^{d_i}\}$, the ordered multi-set of finite dimensional Euclidean spaces, i.e., ${}^{k}\mathbb{T} \subseteq \mathbb{R}^{d_k}$, for each k in $\mathbb{K} := \{0, 1, \ldots, i\}$.

For each such k, let us define the following two subsets of ${}^{k}\mathbb{T} \times \mathbb{R}_{+} \subseteq \mathbb{R}^{d_k+1}$,

$$\mathcal{B}(\widehat{{}^{k}g}) = \left\{({}^{k}v, u) : {}^{k}v \in {}^{k}\mathbb{T}, 0 \leq u \leq \widehat{{}^{k}g}({}^{k}v)\right\},$$
$$\mathcal{B}({}^{k}f) = \left\{({}^{k}v, u) : {}^{k}v \in {}^{k}\mathbb{T}, 0 \leq u \leq {}^{k}f({}^{k}v)\right\} \quad ,$$

and collectively over all k, let us define the following two sets:

$$\mathcal{B}(\widehat{{}^{\mathbb{K}}g}) = \mathcal{B}(\widehat{{}^{0}g}) \cup \mathcal{B}(\widehat{{}^{1}g}) \cup \cdots \cup \mathcal{B}(\widehat{{}^{i}g}),$$
$$\mathcal{B}({}^{\mathbb{K}}f) = \mathcal{B}({}^{0}f) \cup \mathcal{B}({}^{1}f) \cup \cdots \cup \mathcal{B}({}^{i}f) \quad .$$

Algorithm 1 first produces a sample from the randomly labelled random vector $({}^{K}V, U)$ that is uniformly distributed in $\mathcal{B}(\widehat{{}^{\mathbb{K}}g})$. We can see this by letting $h({}^{k}v, u)$ denote the joint density of $({}^{K}V, U)$ and $h(u|{}^{k}v)$ denote the conditional density of U given ${}^{K}V = {}^{k}v$. Then,

$$h({}^{k}v, u) = \begin{cases} {}^{\mathbb{K}}g({}^{k}v)\, h(u|{}^{k}v) = \left(\sum_{\ell \in \mathbb{K}} {}^{\ell}g({}^{k}v)\, \mathbb{1}_{{}^{\ell}\mathbb{T}}({}^{k}t)\right)\, h(u|{}^{k}v) & \text{if } ({}^{k}v, u) \in \mathcal{B}(\widehat{{}^{\mathbb{K}}g}) \\ 0 & \text{otherwise .} \end{cases}$$

Since we sample a height u for a given ${}^{k}v$ from the Uniform$[0, \widehat{{}^{\mathbb{K}}g}({}^{k}v)]$ distribution,

$$h(u|{}^{k}v) = \begin{cases} \left(\widehat{{}^{\mathbb{K}}g}({}^{k}v)\right)^{-1} = \left(N_{\widehat{{}^{\mathbb{K}}g}} \sum_{\ell \in \mathbb{K}} {}^{\ell}g({}^{k}v)\, \mathbb{1}_{{}^{\ell}\mathbb{T}}({}^{k}t)\right)^{-1} & \text{if } u \in [0, \widehat{{}^{\mathbb{K}}g}({}^{k}v)] \\ 0 & \text{otherwise.} \end{cases}$$

Therefore,

$$h({}^{k}v,u)=\begin{cases}{}^{\mathbb{K}}g({}^{k}v)\,h(u|{}^{k}v)=\dfrac{{}^{\mathbb{K}}g({}^{k}v)}{N_{\widehat{{}^{\mathbb{K}}g}}\;{}^{\mathbb{K}}g({}^{k}v)}=\left(N_{\widehat{{}^{\mathbb{K}}g}}\right)^{-1} & \text{if } ({}^{k}v,u)\in\mathcal{B}(\widehat{{}^{\mathbb{K}}g})\\ 0 & \text{otherwise .}\end{cases}$$

Thus, we have shown that the joint density of the random vector $({}^{K}V,U)$ initially produced by Algorithm 1 is uniformly distributed on $\mathcal{B}(\widehat{{}^{\mathbb{K}}g})$. The above relationship also makes geometric sense since the volume of $\mathcal{B}(\widehat{{}^{\mathbb{K}}g})$ is exactly $N_{\widehat{{}^{\mathbb{K}}g}}$.

Now, let $({}^{K}T,S)$ be the accepted randomly labelled random vector during the accept/reject step of Algorithm 1, i.e.,

$$({}^{K}T,S)=({}^{K}V,U)\in\mathcal{B}({}^{\mathbb{K}}f)\subseteq\mathcal{B}(\widehat{{}^{\mathbb{K}}g})\ .$$

Then, the uniform distribution of $({}^{K}V,U)$ on $\mathcal{B}(\widehat{{}^{\mathbb{K}}g})$ implies the uniform distribution of $({}^{K}T,S)$ on $\mathcal{B}({}^{\mathbb{K}}f)$. Since the volume of $\mathcal{B}({}^{\mathbb{K}}f)$ is $N_{{}^{\mathbb{K}}f}$, the density of $({}^{K}T,S)$ is identically $1/N_{{}^{\mathbb{K}}f}$ on $\mathcal{B}({}^{\mathbb{K}}f)$ and 0 elsewhere. Hence, the marginal density of ${}^{K}T$ on ${}^{\mathbb{K}}\mathbb{T}$ is

$$\begin{aligned}\int\limits_{0}^{\sum_{\ell\in\mathbb{K}}{}^{\ell}f({}^{k}t)\,\mathbb{1}_{{}^{\ell}\mathbb{T}}({}^{k}t)} 1/N_{{}^{\mathbb{K}}f}\,dh &= 1/N_{{}^{\mathbb{K}}f}\ \int_0^{{}^{\mathbb{K}}f({}^{k}t)}1\,dh\\ &= 1/N_{{}^{\mathbb{K}}f}\int_0^{N_{{}^{\mathbb{K}}f}\,{}^{\mathbb{K}}f^{\cdot}({}^{k}t)}1\,dh,\\ &= {}^{\mathbb{K}}f^{\cdot}({}^{k}t)\ .\end{aligned}$$

Thus, we have shown that the accepted randomly labelled random vector ${}^{K}T$ has the desired density ${}^{\mathbb{K}}f^{\cdot}$. □

Let $\boldsymbol{A}(\widehat{{}^{\mathbb{K}}g})$ be the probability that a labelled point proposed according to the distribution ${}^{\mathbb{K}}g$ gets accepted as an independent sample from ${}^{\mathbb{K}}f^{\cdot}$ through the envelope function $\widehat{{}^{\mathbb{K}}g}$. Observe that the envelope-specific acceptance probability $\boldsymbol{A}(\widehat{{}^{\mathbb{K}}g})$ is the ratio of the integrals

$$\boldsymbol{A}(\widehat{{}^{\mathbb{K}}g})=\frac{N_{{}^{\mathbb{K}}f}}{N_{\widehat{{}^{\mathbb{K}}g}}}:=\frac{\sum_{k\in\mathbb{K}}\int_{{}^{k}\mathbb{T}}{}^{k}f({}^{k}t)\ d({}^{k}t)}{\sum_{k\in\mathbb{K}}\int_{{}^{k}\mathbb{T}}\widehat{{}^{\mathbb{K}}g}({}^{k}t)\ d({}^{k}t)}\ ,$$

and the probability distribution over the number of samples from ${}^{\mathbb{K}}g$ to obtain one sample from ${}^{\mathbb{K}}f^{\cdot}$ is geometrically distributed with mean $1/\boldsymbol{A}(\widehat{{}^{\mathbb{K}}g})$.

The maximum acceptance probability is attained when we can directly sample from ${}^{\mathbb{K}}f^{\cdot}$, i.e., when $\widehat{{}^{\mathbb{K}}g}={}^{\mathbb{K}}g={}^{\mathbb{K}}f={}^{\mathbb{K}}f^{\cdot}$. In practise, we often optimise over a class $\widehat{\mathcal{G}}$ of envelope functions that are easy to sample from. For a class of rejection samplers induced by a given class of envelope functions $\widehat{\mathcal{G}}$, the optimal rejection sampler has the envelope function $\widehat{{}^{\mathbb{K}}g^{*}}\in\widehat{\mathcal{G}}$, such that:

$$\widehat{{}^{\mathbb{K}}g^{*}}:=\operatorname*{argmax}_{\widehat{{}^{\mathbb{K}}g}\in\widehat{\mathcal{G}}}\boldsymbol{A}(\widehat{{}^{\mathbb{K}}g})\ .$$

Note that when there is exactly one model labelled 0, i.e., $\mathbb{K}=\{0\}$, we may ignore the left-super-scripted model label and our trans-dimensional von Neumann rejection sampler or TRS reduces to the classical von Neumann rejection sampler or RS.

In Section 4 we will see an auto-validating method to construct the proposal and envelope required for TRS in order to produce exact IID samples from a large class of possibly trans-dimensional target densities using the theory of interval analysis that is introduced in Section 3. Such a trans-dimensional extension of the classical RS is novel, natural and straightforward to implement with our object-oriented concepts of operatable labelled objects as briefed in Section 4.3.

3 Interval Analysis

This section is a self-contained introduction to interval analysis for researchers in computational statistics who may be unfamiliar with this area. Those who are familiar with interval analysis may skip this Section. Let $\mathbb{IR}$ denote the set of closed real intervals. Let any element of $\mathbb{IR}$ be denoted by $\boldsymbol{x} := [\underline{x}, \overline{x}]$, where, $\underline{x} \leq \overline{x}$ and $\underline{x}, \overline{x} \in \mathbb{R}$. An interval $\boldsymbol{x}$ is called *thin* if $\underline{x} = \overline{x}$ and *thick* if $\underline{x} < \overline{x}$. Let the *width* of $\boldsymbol{x} \in \mathbb{IR}$ be $\operatorname{wid}(\boldsymbol{x}) := \overline{x} - \underline{x}$ and let its *radius* be $\operatorname{rad}(\boldsymbol{x}) := \operatorname{wid}(\boldsymbol{x})/2$. We write $\inf \boldsymbol{x} := \underline{x}$ for the *lower bound*, $\sup \boldsymbol{x} := \overline{x}$ for the *upper bound*. Let $|x| = \operatorname{abs}(x)$ denote the absolute value of $x \in \mathbb{R}$. Let the *mignitude* of an interval $\boldsymbol{x}$ be the number $\check{\boldsymbol{x}} := \min\{|x| : x \in \boldsymbol{x}\} = \mathbb{1}_{\boldsymbol{x}}(0) \min\{|\underline{x}|, |\overline{x}|\}$ and its *magnitude* be the number $\hat{\boldsymbol{x}} := \max\{|x| : x \in \boldsymbol{x}\} = \max\{|\underline{x}|, |\overline{x}|\}$. If $\mathbb{S}$ is a non-empty bounded subset of $\mathbb{R}$ then $\square\mathbb{S} := [\inf(\mathbb{S}), \sup(\mathbb{S})]$ is the *hull* of $\mathbb{S}$, i.e., the tightest interval containing $\mathbb{S}$. Intervals as sets inherit standard set relations. Let $\underline{x}, \overline{x} \in \mathbb{R}^d$ be real (column) vectors such that $\underline{x}_i \leq \overline{x}_i$, for all $i = 1, 2, \ldots, d$, then $\boldsymbol{x} := [\underline{x}, \overline{x}]$ is an *interval (column) vector* or a *box*. The set of all such boxes is denoted by $\mathbb{IR}^d$. The i-th component of the box $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d)$ is the interval $\boldsymbol{x}_i = [\underline{x}_i, \overline{x}_i]$ and the interval extension of a set $\mathbb{D} \subseteq \mathbb{R}^d$ is $\mathbb{ID} := \{\boldsymbol{x} \in \mathbb{IR}^d : \underline{x}, \overline{x} \in \mathbb{D}\}$. Let the *volume* of a box $\boldsymbol{x} \in \mathbb{IR}^d$ be $\operatorname{vol}(\boldsymbol{x}) := \prod_{i=1}^d \operatorname{wid}(\boldsymbol{x}_i)$. The diameter, radius, mignitude, magnitude, infimum and supremum of a box $\boldsymbol{x} \in \mathbb{IR}^d$ are defined component-wise. Let the maximum norm of a vector $x \in \mathbb{R}^d$ be $\|x\|_\infty := \max_k |x_k|$. Let the vector valued hyper-metric between boxes $\boldsymbol{x}$ and $\boldsymbol{y}$ be

$$\operatorname{dist}(\boldsymbol{x}, \boldsymbol{y}) := \Big(\sup\{|\underline{x}_1 - \underline{y}_1|, |\overline{x}_1 - \overline{y}_1|\}, \cdots, \sup\{|\underline{x}_d - \underline{y}_d|, |\overline{x}_d - \overline{y}_d|\}\Big) \quad .$$

We can turn $\mathbb{IR}^d$ into a metric space with the *Hausdorff distance* $\operatorname{dist}_\infty(\boldsymbol{x}, \boldsymbol{y}) = \|\operatorname{dist}(\boldsymbol{x}, \boldsymbol{y})\|_\infty$.

Our main motivation for the extension to intervals is to enclose the *range*

$$\operatorname{range}(f; \mathbb{S}) := \{f(x) : x \in \mathbb{S}\}$$

of a real-valued function $f : \mathbb{R}^d \to \mathbb{R}$ over a set $\mathbb{S} \subseteq \mathbb{R}^d$. Interval analysis can help us to rigorously enclose the range. First we need to avoid the conventional "black-box" paradigm of pointwise function evaluations with a computer using conventional floating-point arithmetic. The "anti-black-box" paradigm to function evaluations starts with an understanding of how computers encode or express real functions from elementary arithmetic operations and standard functions and extend this expression to interval arguments in a manner that encloses all sources of numerical error.

Elementary binary operations $\star \in \Upsilon := \{+, -, *, /, \hat{}\,\}$ are defined on $\mathbb{IR}$ by putting

$$\boldsymbol{x} \star \boldsymbol{y} := \square\{x \star y : x \in \boldsymbol{x}, y \in \boldsymbol{y}\} = \{x \star y : x \in \boldsymbol{x}, y \in \boldsymbol{y}\}$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{IR}$ such that $x \star y$ is defined for all $x \in \boldsymbol{x}$ and $y \in \boldsymbol{y}$. This restricts the definition of the division $\boldsymbol{x}/\boldsymbol{y}$ to intervals $\boldsymbol{y}$ with $0 \notin \boldsymbol{y}$ and the definition of the exponentiation $\boldsymbol{x}\hat{}\boldsymbol{y}$ to one of the cases (i) $\underline{x} > 0$, (ii) $\underline{x} \geq 0, \underline{y} > 0$, (iii) $0 \notin \boldsymbol{x}$, $\boldsymbol{y}$ is an integer ≤ 0, or (iv) $\boldsymbol{y}$ is a positive integer.

Standard functions are members of a predefined set $\mathfrak{S}$ of univariate real functions that are continuous on every closed interval on which they are defined. We extend any $\varphi \in \mathfrak{S}$ from $\varphi : \mathbb{S}_\varphi \to \mathbb{R}$, $\mathbb{S}_\varphi \subseteq \mathbb{R}$ to take interval arguments by putting

$$\varphi(\boldsymbol{x}) := \square\mathrm{range}(\varphi; \boldsymbol{x}) = \mathrm{range}(\varphi; \boldsymbol{x})$$

for all $\boldsymbol{x} \in \mathbb{IS}$ such that $\varphi(x)$ is defined for all $x \in \boldsymbol{x}$. We let

$$\begin{aligned}\mathfrak{S}_0 := \{&\mathrm{abs}\,(\text{absolute value}), \mathrm{sqr}\,(\text{square}), \mathrm{sqrt}\,(\text{square root}), \exp\ (\text{exponential}),\\ &\log\ (\text{natural logarithm}), \sin\ (\text{sine}), \cos\ (\text{cosine}), \arctan\ (\text{arc tangent})\}\end{aligned}$$

be a fundamental class of such standard functions as in [40]. Due to the monotonicity properties of operations and elementary functions we find that,

$$\begin{array}{lcl}
\boldsymbol{x} \star \boldsymbol{y} & = & \square\{\underline{x} \star \underline{y}, \underline{x} \star \overline{y}, \overline{x} \star \underline{y}, \overline{x} \star \overline{y}\} \ \text{ for } \star \in \{+,-,*,/\} \text{ and } 0 \notin \boldsymbol{y} \text{ if } \star = / \ , \\
|\boldsymbol{x}| & = & \mathrm{abs}(\boldsymbol{x}) = [\check{\boldsymbol{x}}, \hat{\boldsymbol{x}}] \ , \\
\boldsymbol{x}^2 & = & \mathrm{sqr}(\boldsymbol{x}) = [\check{\boldsymbol{x}}^2, \hat{\boldsymbol{x}}^2] \ , \\
e^{\boldsymbol{x}} & = & \exp(\boldsymbol{x}) = [\exp(\underline{x}), \exp(\overline{x})] \ , \\
\arctan(\boldsymbol{x}) & = & [\arctan(\underline{x}), \arctan(\overline{x})] \ , \\
\sqrt{\boldsymbol{x}} & = & \mathrm{sqrt}(\boldsymbol{x}) = [\mathrm{sqrt}(\underline{x}), \mathrm{sqrt}(\overline{x})] \ \text{ if } 0 \le \underline{x} \ , \\
\log(\boldsymbol{x}) & = & [\log(\underline{x}), \log(\overline{x})] \ \text{ if } 0 < \underline{x} \ .
\end{array}$$

The piecewise monotone sin function is extended to $\mathbb{IR}$ by determining whether $\boldsymbol{x}$ intersects the sets $\mathbb{S}^+ := \cup_{k\in\mathbb{Z}}(2k\boldsymbol{p} + \boldsymbol{p}/2)$ and $\mathbb{S}^- := \cup_{k\in\mathbb{Z}}(2k\boldsymbol{p} - \boldsymbol{p}/2)$ using an enclosure of π with the interval $\boldsymbol{p} = [\underline{p}, \overline{p}]$ with enough leading digits as follows:

$$\sin(\boldsymbol{x}) = \begin{cases} [-1, 1] & \text{if } \boldsymbol{x} \cap \mathbb{S}^- \neq \emptyset\,, \boldsymbol{x} \cap \mathbb{S}^+ \neq \emptyset, \\ [-1, \max\{\sin(\underline{x}), \sin(\overline{x})\}] & \text{if } \boldsymbol{x} \cap \mathbb{S}^- \neq \emptyset\,, \boldsymbol{x} \cap \mathbb{S}^+ = \emptyset, \\ [\min\{\sin(\underline{x}), \sin(\overline{x})\}, 1] & \text{if } \boldsymbol{x} \cap \mathbb{S}^- = \emptyset\,, \boldsymbol{x} \cap \mathbb{S}^+ \neq \emptyset, \\ [\min\{\sin(\underline{x}), \sin(\overline{x})\}, \max\{\sin(\underline{x}), \sin(\overline{x})\}] & \text{if } \boldsymbol{x} \cap \mathbb{S}^- = \emptyset\,, \boldsymbol{x} \cap \mathbb{S}^+ = \emptyset. \end{cases}$$

From standard identities, we get $\cos(\boldsymbol{x}) = \sin(\boldsymbol{x} + \pi/2)$ for any $\boldsymbol{x} \in \mathbb{IR}$ and for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{IR}$, the exponentiation $\boldsymbol{x}\hat{\ }\boldsymbol{y} = \exp(\boldsymbol{y} * \log(\boldsymbol{x}))$ provided $\underline{x} > 0$. The operation $\boldsymbol{x} \star \boldsymbol{y}$ for $\star \in \{+,-,*,/\}$ given by $\square\{\underline{x} \star \underline{y}, \underline{x} \star \overline{y}, \overline{x} \star \underline{y}, \overline{x} \star \overline{y}\}$ above can be further simplified as follows:

$$\begin{array}{lcl}
\boldsymbol{x} + \boldsymbol{y} & = & [\underline{x} + \underline{y}, \overline{x} + \overline{y}] \ , \\
\boldsymbol{x} - \boldsymbol{y} & = & [\underline{x} - \overline{y}, \overline{x} - \underline{y}] \ , \\
\boldsymbol{x} * \boldsymbol{y} & = & [\min\{\underline{x} * \underline{y}, \underline{x} * \overline{y}, \overline{x} * \underline{y}, \overline{x} * \overline{y}\}, \max\{\underline{x} * \underline{y}, \underline{x} * \overline{y}, \overline{x} * \underline{y}, \overline{x} * \overline{y}\}] \ , \\
\boldsymbol{x}/\boldsymbol{y} & = & \boldsymbol{x} * [1/\overline{y}, 1/\underline{y}], \text{ provided, } 0 \notin \boldsymbol{y} \ .
\end{array}$$

Interval multiplication is branched into nine cases, on the basis of the signs of the boundaries of the operands, such that only one case entails more than two real multiplications. Thus, every elementary binary operation in Υ and every elementary standard function in $\mathfrak{S}_0$ has been extended to $\mathbb{IR}$.

Other commonly used real functions may be included in the set of standard functions,

$$\mathfrak{S} = \mathfrak{S}_0 \cup \{\, \log_b(x), x^n, x^{p/q}, \tan(x), \sinh(x), \ldots, \arcsin(x), \ldots \,\} \ ,$$

and extended to $\mathbb{IR}$, provided they are continuous on each closed interval on which they are defined either directly or by using standard identities involving functions in

$\mathfrak{S}_0$ and operations in Υ. For example, the interval extension of the integer power function for product likelihood is defined by:

$$\boldsymbol{x}^n = \begin{cases} [\underline{x}^n, \overline{x}^n] & : \text{ if } n \in \{1, 2, \ldots\} \text{ is odd,} \\ [\check{x}^n, \hat{\boldsymbol{x}}^n] & : \text{ if } n \in \{1, 2, \ldots\} \text{ is even,} \\ [1, 1] & : \text{ if } n = 0, \\ [1/\overline{x}, 1/\underline{x}]^{-n} & : \text{ if } n \in \{-1, -2, \ldots\}; 0 \notin \boldsymbol{x}. \end{cases}$$

To work with partially defined real functions $f(x) : \mathbb{S} \to \mathbb{R}$ with $\mathbb{S} \subsetneq \mathbb{R}$, such as $\sqrt{x}$ and $\log(x)$, we introduce the symbol NaN ('not a number') that may be thought of as 'undefined image' and let $\mathbb{R}^* := \mathbb{R} \cup \{\mathsf{NaN}\}$ and $\mathbb{IR}^* := \mathbb{IR} \cup \{\mathsf{NaN}\}$. We can extend the elementary interval operations and standard interval functions from $\mathbb{IR}$ to $\mathbb{IR}^*$ by defining NaN as the value of the expressions $\boldsymbol{x} \star \boldsymbol{y}$ or $\mathsf{f}(\boldsymbol{x})$ that is undefined for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{IR}$, $\star \in \Upsilon$, $\mathsf{f} \in \mathfrak{S}$. Thus, for any $\boldsymbol{x} \in \mathbb{IR}^*$, we get $\mathsf{NaN} \star \boldsymbol{x} = \boldsymbol{x} \star \mathsf{NaN} = \mathsf{f}(\mathsf{NaN}) = \mathsf{NaN}$. The inclusion relations are extended to $\mathbb{IR}^*$ from $\mathbb{IR}$ by defining $x \in \mathsf{NaN}$ for all $x \in \mathbb{R}$, $x \subseteq \mathsf{NaN}$ for all $x \in \mathbb{IR}^*$, $\mathsf{NaN} \subseteq \boldsymbol{x}$ only for $\boldsymbol{x} = \mathsf{NaN}$. We take NaN to be greater than any real number and define $\operatorname{rad}(\mathsf{NaN}) := \mathsf{NaN}$ and $|\mathsf{NaN}| := \mathsf{NaN}$.

An interval function $\boldsymbol{f}(x) : \mathbb{IR}^d \to \mathbb{IR}^*$ is called *inclusion isotone* if, for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{IR}^d$,

$$\boldsymbol{x} \subseteq \boldsymbol{y} \implies \boldsymbol{f}(\boldsymbol{x}) \subseteq \boldsymbol{f}(\boldsymbol{y}) \ . \tag{7}$$

An interval function $\boldsymbol{f} : \mathbb{IR}^d \to \mathbb{IR}^*$ is an *interval extension* of the real function $f : \mathbb{S} \subseteq \mathbb{R}^d \to \mathbb{R}$ if

$$\boldsymbol{f}(x) = f(x) \text{ for } x \in \mathbb{S} \ , \tag{8}$$

$$f(x) \in \boldsymbol{f}(\boldsymbol{x}) \text{ for all } x \in \boldsymbol{x} \in \mathbb{IS} \ , \tag{9}$$

$$\boldsymbol{f}(\boldsymbol{x}) = \mathsf{NaN} \text{ for } \boldsymbol{x} \notin \mathbb{IS} \ . \tag{10}$$

A large class of inclusion isotone interval extensions of real functions can be obtained from arithmetical expressions. An *arithmetical expression* [40, p. 13–14] in the formal variables $\xi_1, \xi_2, \ldots, \xi_d$ is a member of the set $\mathfrak{E} = \mathfrak{E}(\xi_1, \xi_2, \ldots, \xi_d)$ defined by

1. $\mathbb{R} \subseteq \mathfrak{E}$,
2. $\xi_i \in \mathfrak{E}$ for $i = 1, \ldots, d$,
3. $g, h \in \mathfrak{E} \implies (g \star h) \in \mathfrak{E}$ for all $\star \in \Upsilon$,
4. $g \in \mathfrak{E} \implies \varphi(g) \in \mathfrak{E}$ for all $\varphi \in \mathfrak{S}$,
5. among the sets $\mathfrak{E}$ satisfying (1)–(4) above, $\mathfrak{E}(\xi_1, \xi_2, \ldots, \xi_d)$ is minimal with respect to inclusion.

An expression k is a *sub-expression* of f if $f = k$, or $f = g \star h$ and k is a sub-expression of g or h, or $f = \varphi(g)$ and k is a sub-expression of g. When writing down arithmetical expressions we take advantage of standard conventions regarding deletion of brackets, writing $-\xi$ for $(0 - \xi)$ or $(0 - 1) * \xi$, $\xi_1 \xi_2$ for $(\xi_1 * \xi_2)$, $\xi_1^{\xi_2}$ for $(\xi_1 \hat{\ } \xi_2)$, $\xi_1 + \xi_2 \xi_3$ for $(\xi_1 + (\xi_2 * \xi_3))$, $\xi_1 \xi_2 \xi_3$ for $((\xi_1 * \xi_2) * \xi_3)$ or $(\xi_1 * (\xi_2 * \xi_3))$, etc. Since there may be many different precise arithmetical expressions corresponding to an expression written under standard conventions, we make the expression precise if necessary.

Example 3 (Stretched Oscillating Exponential Shape)**.** *The* (a, b, c)*-parametric family of stretched oscillating exponential shape* $f(t)$ *for* $t \in [0, \infty)$ *is*

$$f(t) = \exp(-at^b)(1 + c \sin(at^b \tan(b\pi))), \tag{11}$$

where parameter $a > 0$ determines the scale, parameter $b \in (0, 1/2)$ determines the stretch and frequency of the oscillations, and parameter $c \in (-1, 1)$ determines the magnitude of the oscillations. The density corresponding to this shape is given by Shiryaev [49, p. 294] as a counter example to the claim that knowing all the moments determines the density. The precise arithmetical expression of f in the formal variable ξ_1 is taken to be

$$f(\xi_1) = \exp((0-1) * (a * (\xi_1 \hat{} b))) * (1 + (c * \sin(((a * (\xi_1 \hat{} b)) * \tan(b * \pi))))) \ , \quad (12)$$

with 19 *sub-expressions $f_1, f_2, \ldots, f_{19} = f$ as follows:*

$$\begin{aligned} f_1 = 0,\ f_2 = 1,\ f_3 = a,\ f_4 = b,\ f_5 = \xi_1,\ f_6 = f_1 - f_2,\ f_7 = f_5 \hat{} f_4, \\ f_8 = f_3 * f_7,\ f_9 = f_6 * f_8,\ f_{10} = \exp(f_9),\ f_{11} = c,\ f_{12} = \pi, \\ f_{13} = f_4 * f_{12},\ f_{14} = \tan(f_{13}),\ f_{15} = f_8 * f_{14},\ f_{16} = \sin(f_{15}), \\ f_{17} = f_{11} * f_{16},\ f_{18} = f_2 + f_{17},\ f = f_{19} = f_{10} * f_{18} \quad (13) \end{aligned}$$

For an arithmetical expression $f(\xi) = f(\xi_1, \xi_2, \ldots, \xi_d)$ in d variables we can perform *interval evaluation* to obtain the *value* $\boldsymbol{f}(\boldsymbol{x})$ of f at $\boldsymbol{x} \in \mathbb{IR}^d$ by substituting the intervals $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_d$ for the corresponding formal parameters $\xi_1, \xi_2, \ldots, \xi_d$ in $f(\xi_1, \xi_2, \ldots, \xi_d)$. Formally, $\boldsymbol{f}(\boldsymbol{x})$ is the element of $\mathbb{IR}^*$ that is defined recursively as follows:

$$\begin{array}{lcll} \boldsymbol{f}(\boldsymbol{x}) & = & [c, c] & \text{if } f = c \text{ is a real constant,} \\ \boldsymbol{f}(\boldsymbol{x}) & = & \boldsymbol{x}_i & \text{if } f = \xi_i \text{ is a variable,} \\ \boldsymbol{f}(\boldsymbol{x}) & = & \boldsymbol{g}(\boldsymbol{x}) \star \boldsymbol{h}(\boldsymbol{x}) & \text{if } f = (g \star h), \star \in \Upsilon, \\ \boldsymbol{f}(\boldsymbol{x}) & = & \varphi(\boldsymbol{g}(\boldsymbol{x})) & \text{if } f = \varphi(g), \varphi \in \mathfrak{S}. \end{array}$$

For example, the arithmetical expression $f(\xi_1) = f_{10}(\xi_1) = \exp(-a\xi_1^b) = \exp((0 - 1) * (a * (\xi_1 \hat{} b)))$ of (13) has the sub-expressions: $f_1 = 0$, $f_2 = 1$, $f_3 = a$, $f_4 = b$, $f_5 = \xi_1$, $f_6 = f_1 - f_2$, $f_7 = f_5 \hat{} f_4$, $f_8 = f_3 * f_7$, $f_9 = f_6 * f_8$, and $f_{10} = \exp(f_9)$. Thus for $\boldsymbol{x} = [0.5, 1]$ and constants $a = 0.125$ and $b = 0.45$ its interval evaluation $\boldsymbol{f}(\boldsymbol{x})$ is recursively computed as follows: $\boldsymbol{f}_1(\boldsymbol{x}) = [0, 0]$, $\boldsymbol{f}_2(\boldsymbol{x}) = [1, 1]$, $\boldsymbol{f}_3(\boldsymbol{x}) = [0.125, 0.125]$, $\boldsymbol{f}_4(\boldsymbol{x}) = [0.45, 0.45]$, $\boldsymbol{f}_5(\boldsymbol{x}) = \boldsymbol{x} = [0.5, 1]$, $\boldsymbol{f}_6(\boldsymbol{x}) = [0, 0] - [1, 1] = [-1, -1]$, $\boldsymbol{f}_7(\boldsymbol{x}) = [0.5, 1]\hat{}[0.45, 0.45] = [0.5^{0.45}, 1]$, $\boldsymbol{f}_8(\boldsymbol{x}) = [0.125, 0.125]*[0.5^{0.45}, 1] = [0.125 * 0.5^{0.45}, 0.125]$, $\boldsymbol{f}_9(\boldsymbol{x}) = [-1, -1] * [0.125 * 0.5^{0.45}, 0.125] = [-0.125, -0.125 * 0.5^{0.45}]$. Finally, $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}_{10}(\boldsymbol{x}) = [e^{-0.125}, e^{-0.125*0.5^{0.45}}]$.

On a computer with a finite set $\mathbb{F}$ of machine-representable floating-point numbers, $\mathbb{IF}$ is also finite. Since $\mathbb{F}^* := \mathbb{F} \cup \{\mathsf{NaN}\}$ is not arithmetically closed, when performing arithmetic with intervals in $\mathbb{IF}^*$ we must round the resulting interval *outwards* to guarantee inclusion of the true result [20, 30]. For any $x \in \mathbb{R}^*$ let $\triangledown x := \max\{y \in \mathbb{F}^* : y \leq x\}$ be x *rounded down*, $\vartriangle x := \min\{y \in \mathbb{F}^* : y \geq x\}$ be x *rounded up* and for any $\boldsymbol{x} \in \mathbb{IR}^*$ let $\Diamond \boldsymbol{x} := [\triangledown \underline{x}, \vartriangle \overline{x}]$ be $\boldsymbol{x}$ *rounded outward*. For an arithmetical expression $f(\xi) = f(\xi_1, \xi_2, \ldots, \xi_d)$ in d variables we can perform *outward-rounded interval evaluation* to obtain the *outward-rounded value* $\boldsymbol{f}^{\Diamond}(\boldsymbol{x})$ of f at $\boldsymbol{x} \in \mathbb{IR}^d$ by substituting the intervals $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_d$ for the corresponding formal parameters $\xi_1, \xi_2, \ldots, \xi_d$ in $f(\xi_1, \boldsymbol{x}_2, \ldots, \xi_d)$. Formally, $\boldsymbol{f}^{\Diamond}(\boldsymbol{x})$ is the element of $\mathbb{IF}^*$ that is defined recursively as follows:

$$\begin{array}{lcll} \boldsymbol{f}^{\Diamond}(\boldsymbol{x}) & = & \Diamond c & \text{if } f = c \text{ is a real constant,} \\ \boldsymbol{f}^{\Diamond}(\boldsymbol{x}) & = & \Diamond \boldsymbol{x}_i & \text{if } f = \xi_i \text{ is a variable,} \\ \boldsymbol{f}^{\Diamond}(\boldsymbol{x}) & = & \boldsymbol{g}^{\Diamond}(\boldsymbol{x}) \star^{\Diamond} \boldsymbol{h}^{\Diamond}(\boldsymbol{x}) & \text{if } f = (g \star h), \star \in \Upsilon, \\ \boldsymbol{f}^{\Diamond}(\boldsymbol{x}) & = & \varphi^{\Diamond}(\boldsymbol{g}^{\Diamond}(\boldsymbol{x})) & \text{if } f = \varphi(g), \varphi \in \mathfrak{S}, \end{array}$$

where $\star^{\Diamond}$ and $\varphi^{\Diamond}$ are outward rounded evaluations in $\mathbb{IF}^*$. For example, $\boldsymbol{x} +^{\Diamond} \boldsymbol{y} = [\triangledown(\underline{x} + \underline{y}), \vartriangle(\overline{x} + \overline{y})]$, $\exp^{\Diamond}(\boldsymbol{x}) = [\triangledown e^{\underline{x}}, \vartriangle e^{\overline{x}}]$ and

$$\boldsymbol{f}^{\Diamond}(\boldsymbol{x}) = \boldsymbol{f}^{\Diamond}_{10}(\boldsymbol{x}) = [\triangledown e^{\triangledown(\triangledown(-1)*\vartriangle(0.125))}, \vartriangle e^{\vartriangle\left(\vartriangle(-1)*\triangledown\left(\triangledown(0.125)*\triangledown\left(\triangledown(0.5)^{\triangledown(0.45)}\right)\right)\right)}] \ .$$

Examples 1 and 2 are bivariate arithmetical expressions. Example 3 is a more complicated univariate arithmetical expression.

Theorem 2 (Fundamental theorem of interval analysis, Moore)**.** *The interval functions $\boldsymbol{f}$ and $\boldsymbol{f}^{\Diamond}$ associated with an arithmetical expression $f : \mathbb{S} \subset \mathbb{R}^d \to \mathbb{R}$ are inclusion isotone and we have*

$$\boldsymbol{x} \in \mathbb{IS} \implies \text{range}(f; \boldsymbol{x}) \subseteq \boldsymbol{f}(\boldsymbol{x}) \subseteq \boldsymbol{f}^{\Diamond}(\boldsymbol{x}) \ . \tag{14}$$

Proof. First let us show that

$$\boldsymbol{x}, \boldsymbol{y} \in \mathbb{IS}, \boldsymbol{x} \subseteq \boldsymbol{y} \implies \boldsymbol{f}(\boldsymbol{x}) \subseteq \boldsymbol{f}(\boldsymbol{y}) \ . \tag{15}$$

Obviously, this holds when f is a constant or variable. In light of the recursive definition of the arithmetical expression f it only remains to show that if (15) holds for g and h in place of f then it holds for $f = g \star h$ for $\star \in \Upsilon$ and $f = \varphi(g)$ for $\varphi \in \mathfrak{S}$. But if $\boldsymbol{x} \subseteq \boldsymbol{y}$, $\boldsymbol{g}(\boldsymbol{x}) \subseteq \boldsymbol{g}(\boldsymbol{y})$, $\boldsymbol{h}(\boldsymbol{x}) \subseteq \boldsymbol{h}(\boldsymbol{y})$, then in the first case

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{x}) \star \boldsymbol{h}(\boldsymbol{x}) &= \square\{\tilde{g} \star \tilde{h} : \tilde{g} \in \boldsymbol{g}(\boldsymbol{x}), \tilde{h} \in \boldsymbol{h}(\boldsymbol{x})\} \\ &\subseteq \square\{\tilde{g} \star \tilde{h} : \tilde{g} \in \boldsymbol{g}(\boldsymbol{y}), \tilde{h} \in \boldsymbol{h}(\boldsymbol{y})\} = \boldsymbol{g}(\boldsymbol{y}) \star \boldsymbol{h}(\boldsymbol{y}) = \boldsymbol{f}(\boldsymbol{y}) \ , \end{aligned}$$

and in the second case

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{x}) = \varphi(\boldsymbol{g}(\boldsymbol{x})) &= \square\{\varphi(\tilde{g}) : \tilde{g} \in \boldsymbol{g}(\boldsymbol{x})\} \\ &\subseteq \square\{\varphi(\tilde{g}) : \tilde{g} \in \boldsymbol{g}(\boldsymbol{y})\} = \varphi(\boldsymbol{g}(\boldsymbol{y})) = \boldsymbol{f}(\boldsymbol{y}) \ . \end{aligned}$$

Therefore, we have shown that $\boldsymbol{f}$ is inclusion isotone. To show that $\boldsymbol{f}^{\Diamond}$ is inclusion isotone we have to prove that

$$\boldsymbol{x}, \boldsymbol{y} \in \mathbb{IS}, \boldsymbol{x} \subseteq \boldsymbol{y} \implies \boldsymbol{f}^{\Diamond}(\boldsymbol{x}) \subseteq \boldsymbol{f}^{\Diamond}(\boldsymbol{y}) \ . \tag{16}$$

Since, $\boldsymbol{x} \subseteq \Diamond\boldsymbol{x}$, this holds if f is a constant or variable. Due to the implication $\boldsymbol{x} \subseteq \boldsymbol{y} \implies \Diamond\boldsymbol{x} \subseteq \Diamond\boldsymbol{y}$, (16) follows inductively as before. Finally, a similar induction shows that $\boldsymbol{f}(\boldsymbol{x}) \subseteq \boldsymbol{f}^{\Diamond}(\boldsymbol{x})$, and (14) follows since, by (15), range$(f; \boldsymbol{x}) \subseteq \boldsymbol{f}(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathbb{IS}$. □

We do not explicitly distinguish between the interval functions $\boldsymbol{f}^{\Diamond}$ and $\boldsymbol{f}$ here with the understanding that machine implementation and enclosure of $\boldsymbol{f}$ is done via $\boldsymbol{f}^{\Diamond}$. The above theorem allows us to enclose the range of any function that has an arithmetical expression, i.e., obtain an upper bound for the global maximum and a lower bound for the global minimum over any compact subset of the domain upon which the function is well-defined. We will see that this is the work-horse for rigorously constructing an envelope for rejection sampling even from highly multimodal target distributions. Unlike the interval functions in $\mathfrak{S}$ that produce exact range enclosures, the interval function $\boldsymbol{f}(\boldsymbol{x})$ of an arithmetical expression f often overestimates range$(f; \boldsymbol{x})$, but can be shown under mild conditions to linearly approach the range as the maximal width of the box $\boldsymbol{x}$ goes to zero. This implies that a partition of $\boldsymbol{x}$ into smaller boxes $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ gives better enclosures of range$(f; \boldsymbol{x})$ through the union $\bigcup_{i=1}^{m} \boldsymbol{f}(\boldsymbol{x}^{(i)})$

as illustrated in Figure 1. Next we make the above statements precise after introducing some required definitions.

Let $\mathbb{R}^{1\times d}$ and $\mathbb{IR}^{1\times d}$ denote the set of d-dimensional real and interval row vectors, respectively. Let $\boldsymbol{x}\boldsymbol{y} := \sum_{i=1}^{d} \boldsymbol{x}_i \boldsymbol{y}_i$ denote the inner product of $\boldsymbol{x} \in \mathbb{IR}^{1\times d}$ and $\boldsymbol{y} \in \mathbb{IR}^{d}$. A real function $f : \mathbb{S} \subseteq \mathbb{R}^d \to \mathbb{R}$ is called *Lipschitz continuous* in $\mathbb{Y} \subseteq \mathbb{S} \subseteq \mathbb{R}^d$ if there is a row vector $L^{\mathbb{Y}} \in \mathbb{R}^{1\times d}$ such that:

$$\left|f(x) - f(x')\right| \le L^{\mathbb{Y}} \left|x - x'\right| \quad \text{for all } x, x' \in \mathbb{Y} \ .$$

Similarly, an interval function $\boldsymbol{f} : \mathbb{IR}^d \to \mathbb{IR}^*$ is called *Lipschitz continuous* in $\mathbb{Y} \in \mathbb{IR}^d$ if $\boldsymbol{f}(\mathbb{Y}) \neq \mathsf{NaN}$ and there is a row vector $L^{\mathbb{Y}} \in \mathbb{R}^{1\times d}$ such that:

$$\text{dist}\left(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}')\right) \le L^{\mathbb{Y}} \,\text{dist}\left(\boldsymbol{x} - \boldsymbol{x}'\right) \quad \text{for all } \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{IY} \ .$$

We call an arithmetical expression f in d variables *Lipschitz* at a box $\boldsymbol{y} \in \mathbb{IR}^d$ if $\boldsymbol{f}(\boldsymbol{y}) \neq \mathsf{NaN}$ and if for all sub-expressions g, h of f, the relation $g = h\,\hat{}\,\alpha$, with $0 < \alpha < 1$, implies $\boldsymbol{h}(\boldsymbol{y}) > 0$, and the relation $g = \varphi(h)$, with $\varphi \in \mathfrak{S}$, implies that φ is defined and Lipschitz continuous in a neighbourhood of $\boldsymbol{h}(\boldsymbol{y})$, i.e., in an interval containing $\boldsymbol{h}(\boldsymbol{y})$ in its interior. We call f *locally Lipschitz* in $\boldsymbol{y} \in \mathbb{IR}^d$ if f is Lipschitz at every $y \in \boldsymbol{y}$.

Theorem 3 (Range enclosure tightens linearly with mesh)**.** *Let f be an arithmetical expression in d variables. If f is Lipschitz at $\boldsymbol{y} \in \mathbb{IR}^d$ then the interval evaluation of f is Lipschitz continuous in $\boldsymbol{y}$ and*

$$\text{rad}\,\left(\boldsymbol{f}(\boldsymbol{x})\right) \le L^{\boldsymbol{x}} \text{rad}\,(\boldsymbol{x}) \ . \tag{17}$$

Furthermore, if f is locally Lipschitz in $\boldsymbol{y} \in \mathbb{IR}^d$ then there exists a positive real number $\rho > 0$ such that f is defined and Lipschitz in $\boldsymbol{x}$ for all $\boldsymbol{x} \subseteq \boldsymbol{y}$ with $\text{rad}\,(x_i) \le \rho$*, for $i = 1, 2, \ldots, d$.*

Proof. The proof is given by an induction similar to the proof of Theorem 2. See proofs of [40, 2.1.1,2.1.2, 2.1.3] for details. □

Theorem 3 gives a naive procedure (typically not suited for practical calculations) to obtain an arbitrarily tight enclosure for the range of an expression f over a box $\boldsymbol{y}$ in which f is locally Lipschitz. Suppose we choose positive integers $m_1, m_2, \ldots, m_d$ to naively partition $\boldsymbol{y}$ into a set $\mathbb{B}_0(\boldsymbol{y})$ of $b_0 = m_1 m_2 \cdots m_d$ boxes with sides of length $2\text{rad}\,(\boldsymbol{y}_i)/m_i \le 2\rho$ and subdivide each such box further to get a set $\mathbb{B}_r(\boldsymbol{y})$ of $r^d b_0$ boxes with sides of length $\le 2\rho/r$, then we obtain the enclosure $\bigcup_{\boldsymbol{x}\in\mathbb{B}_r(\boldsymbol{y})} \boldsymbol{f}(\boldsymbol{x})$ for range$(f; \boldsymbol{y})$,

$$0 \le \text{rad}\left(\bigcup_{\boldsymbol{x}\in\mathbb{B}_r(\boldsymbol{y})} \boldsymbol{f}(\boldsymbol{x})\right) - \text{rad}\,\left(\text{range}(f;\boldsymbol{y})\right) \le \max_{\boldsymbol{x}\in\mathbb{B}_r(\boldsymbol{y})} \text{rad}\,\left(\boldsymbol{f}(\boldsymbol{x})\right) \le \gamma\rho/r, \ , \tag{18}$$

where, $\gamma := \max_{\boldsymbol{x}\in\mathbb{B}_r(\boldsymbol{y})} \sum_{i=1}^{d} L_i^{\boldsymbol{x}}$ and in particular,

$$\lim_{r\to\infty} \bigcup_{\boldsymbol{x}\in\mathbb{B}_r(\boldsymbol{y})} \boldsymbol{f}(\boldsymbol{x}) = \text{range}(f;\boldsymbol{y}) \ . \tag{19}$$

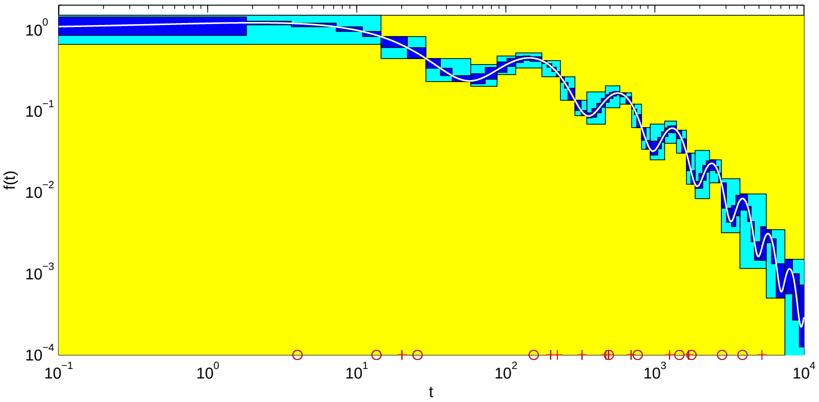

Figure 1: Range enclosure of $f(t) = \exp(-at^b)(1 + c\sin(at^b \tan(b\pi)))$ with $a = 1/8$, $b = 9/20$ and $c = 1/2$ (white line) linearly tightens with the mesh. The range enclosure of the interval extension of f over three adaptive partitions of the domain $[10^{-12}, 10^{12}]$ consisting of 1, 50 and 100 intervals are depicted over $[10^{-1}, 10^4]$ as yellow, cyan and blue rectangles, respectively. See Sections 4.1 and 5.6 for more description.

4 Moore Rejection Sampler

Moore rejection sampler (MRS) is an auto-validating trans-dimensional von Neumann rejection sampler (TRS). MRS is said to be auto-validating because it automatically obtains a proposal ${}^{\mathbb{K}}g$ that is easy to simulate from, and a trans-dimensional envelope $\widehat{{}^{\mathbb{K}}g}$ that is guaranteed to satisfy the envelope condition (5). MRS in a universal algorithm that can produce IID samples from any bounded target density whose shape ${}^{\mathbb{K}}f$ belongs to the class of arithmetical expressions that are locally Lipschitz in ${}^{\mathbb{K}}\mathbb{T}$.

Theorem 4 (MRS)**.** *If the shape of the bounded trans-dimensional density* ${}^{\mathbb{K}}f^{\cdot}({}^{k}t) : {}^{\mathbb{K}}\mathbb{T} \to \mathbb{R}$ *is given by an arithmetical expression* ${}^{\mathbb{K}}f$ *that is locally Lipschitz over* ${}^{\mathbb{K}}\mathbb{T}$*, then the non-empty* ${}^{k}t$ *returned by Algorithm 1 with*

- *envelope function* $\widehat{{}^{\mathbb{K}}g_{\mathfrak{T}}}({}^{k}t) : {}^{\mathbb{K}}\mathbb{T} \to \mathbb{R}$

$$\widehat{{}^{\mathbb{K}}g_{\mathfrak{T}}}({}^{k}t) = \sum_{j=1}^{s} \overline{{}^{k_j}\boldsymbol{f}}({}^{k_j}\boldsymbol{t}_j)\ \mathbb{1}_{{}^{k_j}\boldsymbol{t}_j}({}^{k}t) \enspace , \tag{20}$$

  with a partition $\mathfrak{T} := \{{}^{k_1}\boldsymbol{t}_1, {}^{k_2}\boldsymbol{t}_2, \ldots, {}^{k_s}\boldsymbol{t}_s\}$ *of* ${}^{\mathbb{K}}\mathbb{T}$

- *and proposal density* ${}^{\mathbb{K}}g_{\mathfrak{T}}({}^{k}t) : {}^{\mathbb{K}}\mathbb{T} \to \mathbb{R}$ *as the normalised simple function*

$${}^{\mathbb{K}}g_{\mathfrak{T}}({}^{k}t) = \left(N_{{}^{\mathbb{K}}g_{\mathfrak{T}}}\right)^{-1} \widehat{{}^{\mathbb{K}}g_{\mathfrak{T}}}({}^{k}t), \quad N_{{}^{\mathbb{K}}g_{\mathfrak{T}}} := \sum_{j=1}^{s} \left(\mathrm{vol}\left({}^{k_j}\boldsymbol{t}_j\right) * \overline{{}^{k_j}\boldsymbol{f}}({}^{k_j}\boldsymbol{t}_j)\right) \enspace , \tag{21}$$

is an IID sample from ${}^{K}T \sim {}^{\mathbb{K}}f^{\cdot}$ *and the partition-specific acceptance probability* $\boldsymbol{A}(\widehat{{}^{\mathbb{K}}g_{\mathfrak{T}}}) \to 1$ *as mesh*$(\mathfrak{T}) \to 0$.

Proof. Let $\mathfrak{T} := \{{}^{k_1}\boldsymbol{t}_1, {}^{k_2}\boldsymbol{t}_2, \dots, {}^{k_s}\boldsymbol{t}_s\}$ be a partition of ${}^{\mathbb{K}}\mathbb{T}$ into s labelled boxes. Since ${}^{\mathbb{K}}f$ is locally Lipschitz over ${}^{\mathbb{K}}\mathbb{T}$, ${}^{\mathbb{K}}\boldsymbol{f}({}^{\mathbb{K}}\mathbb{T}) \neq \mathsf{NaN}$ and by Theorem 2 we can enclose $\text{range}({}^{\mathbb{K}}f; {}^{k_j}\boldsymbol{t}_j)$ with the interval function:

$$\begin{aligned} \text{range}({}^{\mathbb{K}}f; {}^{k_j}\boldsymbol{t}_j) &= \text{range}({}^{k_j}\boldsymbol{f}; {}^{k_j}\boldsymbol{t}_j) \\ &\subseteq {}^{k_j}\boldsymbol{f}({}^{k_j}\boldsymbol{t}_j) := [\underline{{}^{k_j}\boldsymbol{f}}({}^{k_j}\boldsymbol{t}_j), \overline{{}^{k_j}\boldsymbol{f}}({}^{k_j}\boldsymbol{t}_j)], \forall\, j \in \{1, 2, ..., s\} \ . \end{aligned} \tag{22}$$

Thus, the partition-specific envelope function given by (20) will satisfy the necessary envelope condition (5) and since the corresponding proposal density ${}^{\mathbb{K}}g_{\mathfrak{T}}$ given by (21) can be sampled from, the theorem follows from Theorem 1. Since ${}^{\mathbb{K}}f$ is locally Lipschitz over ${}^{\mathbb{K}}\mathbb{T}$, the acceptance probability:

$$\boldsymbol{A}\left(\widehat{{}^{\mathbb{K}}g_{\mathfrak{T}}}\right) = \frac{N_{{}^{\mathbb{K}}f}}{N_{\widehat{{}^{\mathbb{K}}g_{\mathfrak{T}}}}} = \frac{N_{{}^{\mathbb{K}}f}}{\sum_{j=1}^{|\mathfrak{T}|}\left(\text{vol}\ \left({}^{k_j}\boldsymbol{t}_j\right) \overline{{}^{k_j}\boldsymbol{f}}({}^{k_j}\boldsymbol{t}_j)\right)} \geq \frac{\sum_{j=1}^{|\mathfrak{T}|}\left(\text{vol}\ \left({}^{k_j}\boldsymbol{t}_j\right) \underline{{}^{k_j}\boldsymbol{f}}({}^{k_j}\boldsymbol{t}_j)\right)}{\sum_{j=1}^{|\mathfrak{T}|}\left(\text{vol}\ \left({}^{k_j}\boldsymbol{t}_j\right) \overline{{}^{k_j}\boldsymbol{f}}({}^{k_j}\boldsymbol{t}_j)\right)} \ , \tag{23}$$

approaches 1 as the mesh of the partition approaches 0 due to Theorem 3 and (19). □

4.1 Geometric Insight

Let us gain geometric insight into the sampler when $\mathbb{K} = \{0\}$ and $d_0 = 1$ from Example 3 and Figure 1. We suppress the only model label appearing unnecessarily as a left super-script for such cases. The range enclosure of the target shape, $f(t) = \exp(-at^b)(1 + c\sin(at^b\tan(b\pi)))$ with $a = 1/8$, $b = 9/20$ and $c = 1/2$, over three adaptive partitions of the domain $[10^{-12}, 10^{12}]$ consisting of 1, 50 and 100 intervals are depicted as yellow, cyan and blue rectangles, respectively, in Figure 1. The upper boundaries of rectangles of a given colour, depicting a simple function is a partition-specific envelope function (20) from the outward rounded interval evaluation of $f(t)$. Normalisation of the envelope function for a given partition gives the corresponding proposal function (21). As the refinement of the domain proceeds through adaptive bisections (described later), the partition size increases. Note how the acceptance probability (ratio of the area below the target shape to that below the envelope) increases with refinement. Twenty exact samples were drawn from this target with $a = 1/8$, $b = 9/20$, $c = 1/2$ using the MRS from a partition of 50 intervals (cyan rectangles). The first ten of these samples are plotted as 'o' and the next ten are plotted as '+' along the horizontal axis of Figure 1. A similar procedure of adaptive bisections is conducted upon the labelled domain boxes of distinct dimensions in order to envelope a trans-dimensional target shape and propose samples for rejection in our MRS.

4.2 Computational Efficiency

We use several standard data-structures and computational statistical techniques to improve the efficiency of our sampler. These ideas are explained in this section and are briefly overviewed below. Priority queues are used to adaptively bisect the labelled boxes that partition the domain. Simple stopping rules are used to stop the adaptive bisections to reach a reasonable acceptance probability. Finally, alias method is used in conjunction with the squeeze principle to construct the proposal.

4.2.1 Prioritised Partitions

We studied the asymptotic behaviour of uniform partitions in (19) for mathematical tractability of any locally Lipschitz arithmetical expression. In practise, we can significantly increase the acceptance probability for a given partition size by adaptively partitioning the domain ${}^{\mathbb{K}}\mathbb{T}$. In our context, adaptive means the possible exploitation of any current information about the target. We can refine the current partition $\mathfrak{T}_\alpha$ and obtain a finer partition $\mathfrak{T}_{\alpha'}$ with an additional labelled box by bisecting a particular labelled box in $\mathfrak{T}_\alpha$ along the midpoint of its first side with the maximal width into two labelled boxes. There are several ways to choose a labelled box ${}^{k_*}\boldsymbol{t}_* \in \mathfrak{T}_\alpha$ for bisection. For instance, the optimal choice based on experiments with three priority functions of (26) in Section 5.1 is

$$ {}^{k_*}\boldsymbol{t}_* = \underset{{}^{k_j}\boldsymbol{t}_j \in \mathfrak{T}_\alpha}{\operatorname{argmax}} \left(\operatorname{vol}\left({}^{k_j}\boldsymbol{t}_j\right) * \operatorname{wid}\left({}^{k_j}\boldsymbol{f}\left({}^{k_j}\boldsymbol{t}_j\right)\right)\right) \ . \tag{24} $$

We employ a priority queue to conduct sequential refinements of ${}^{\mathbb{K}}\mathbb{T}$ under this partitioning scheme. Briefly, a priority queue (PQ) is a container in which the elements may have different user-specified priorities. The priority is based on some sorting criterion that is applicable to the elements in the container. The PQ can be thought of as a collection in which the "next" element is always the one with the highest priority, i.e., the largest with respect to the specified sorting criterion. Since this container sorts using a *heap* which can be thought of as a binary tree, one can add or remove elements in logarithmic time. This is a desirable feature of the PQ.

This approach avoids the exhaustive argmax computations to obtain the ${}^{k_*}\boldsymbol{t}_*$ for bisection at each refinement step. Thus, the current partition is represented by a queue of labelled boxes that are prioritised in descending order by the priority function in (24). Therefore, the labelled box with the largest uncertainty in the enclosure of the integral over it gets bisected first.

4.2.2 Stopping Rules

There are several ways to decide when to stop refining the partition. A simple strategy is to stop when the number of labelled boxes reaches a number that is well within the memory constraints of the computer, say 10^6, or when the lower bound of the acceptance probability given by (23) is above a desired threshold, say 0.1.

4.2.3 Pre-Processed Proposals

Once we have a partition $\mathfrak{T}$ of ${}^{\mathbb{K}}\mathbb{T}$, we can sample ${}^{k}t$ from the proposal density ${}^{\mathbb{K}}g_{\mathfrak{T}}({}^{k}t)$ given by (21) in two steps:

1. Sample a labelled box ${}^{k_j}\boldsymbol{t}_j \in \mathfrak{T}$ according to the discrete distribution:

$$ {}^{\mathbb{K}}\ddot{g}_{\mathfrak{T}}({}^{k_j}\boldsymbol{t}_j) = \frac{\operatorname{vol}\left({}^{k_j}\boldsymbol{t}_j\right)\overline{{}^{k_j}\boldsymbol{f}}({}^{k_j}\boldsymbol{t}_j)}{\sum_{j=1}^{|\mathfrak{T}|}\left(\operatorname{vol}\left({}^{k_j}\boldsymbol{t}_j\right)\overline{{}^{k_j}\boldsymbol{f}}({}^{k_j}\boldsymbol{t}_j)\right)}, \quad {}^{k_j}\boldsymbol{t}_j \in \mathfrak{T} \ , \tag{25} $$

2. Sample a labelled point ${}^{k}t$ uniformly at random from the d_{k_j}-dimensional labelled box ${}^{k_j}\boldsymbol{t}_j$.

Sampling from large discrete distributions (with million states or more) can be made faster by pre-processing the probabilities and saving the result in some convenient look-up table. This basic idea of Marsaglia [34] allows samples to be drawn rapidly. We employ an efficient pre-processing strategy known as the Alias Method [52] that allows samples to be drawn in constant time even for very large discrete distributions as implemented in the GNU Scientific Library [9]. We also minimise the number of evaluations of the target shape ${}^{\mathbb{K}}f$ during the accept/reject step by saving the labelled box-specific computations of $\underline{{}^{k_j}\boldsymbol{f}}({}^{k_j}\boldsymbol{t}_j)$ and $\overline{{}^{k_j}\boldsymbol{f}}({}^{k_j}\boldsymbol{t}_j)$ and exploiting the so-called "squeeze principle", i.e., immediately accepting those labelled points proposed in the labelled box ${}^{k_j}\boldsymbol{t}_j$ that fall below $\underline{{}^{k_j}\boldsymbol{f}}({}^{k_j}\boldsymbol{t}_j)$ when uniformly stretched toward $\overline{{}^{k_j}\boldsymbol{f}}({}^{k_j}\boldsymbol{t}_j)$.

4.3 MRS Software

The concepts of labelled points and labelled boxes that underpin `MRSampler`, our exact trans-dimensional sampler class in `MRS 1.0` [43], build upon the real and interval vector classes in `C-XSC 2.0`, a `C++` class library for extended scientific computing using interval methods [23]. The priority queues and look-up tables that efficiently manage our adaptive partitioning strategy for envelope construction and rapid sampling from the proposal distribution are handled by standard routines in `GSL`, the GNU Scientific Library [9]. `MRS`, `C-XSC` and `GSL` are distributed from source under the terms of GPL, the GNU General Public License. `MRS 1.0` is available from `www.math.canterbury.ac.nz/~r.sainudiin/codes/mrs`.

5 Examples

Having given theoretical and practical considerations to our Moore rejection sampler, we are ready to draw samples from various target densities whose shapes can be given in terms of locally Lipschitz arithmetical expressions. In this Section, we empirically study sampler efficiency by sampling from qualitatively diverse targets since analytical results on efficiency are sharp only for relatively simple target parametrisation. Unless otherwise specified, all computations were done on a 2.8 GHz Pentium IV machine with 1GB RAM. In Section 5.1 we first study the relative efficiencies of MRSs managed by the three PQs (26) by sampling from univariate Gaussian mixture targets. Next, we study the effects of target complexity (number of components, scales and domain size) on sampler efficiency. In Section 5.2 we study the sampler behaviour for a highly multimodal two-dimensional target that is sensitive to a temperature parameter. Using a trivariate mixture target in Section 5.3, we compare MRS to Monte Carlo Markov chain (MCMC) methods that rely on heuristic convergence diagnostics and exploit the connections between RS, importance sampler (IS) and independent Metropolis-Hastings sampler (IMHS) to simultaneously produce samples from all of them. The effect of dimensionality on sampler efficiency is studied in Section 5.4 and 5.5 where we draw samples from multivariate targets, including the multivariate witch's hat. The target densities in Sections 5.1–5.5 are those of random vectors with complex distributions. In Section 5.6 we draw samples from the posterior distribution of the rate parameter for the stretched oscillating exponential model of Example 3. IID trans-dimensional posterior samples are drawn for the first time using a universal sampler from the binomial partition model in Section 5.7 and from the space of phylogenetic triples in Section 5.8.

Table 1: Moore rejection sampling from six different Gaussian mixture target shapes g_n truncated over $\mathbb{T}$, where n is the number of mixture components.

Target	$\mathbb{T}$	Parameters
$g_1(x)$	$[-10^2, 10^2]$	$\mu_1 = -5, \sigma_1 = 1$, and $w_1 = 1.00$
$g_2(x)$	$[-10^2, 10^2]$	$\mu_1 = -5, \sigma_1 = 1$, $w_1 = 0.25$, $\mu_2 = 50$, $\sigma_2 = 0.25$
$g_5(x)$	$[-10^2, 10^2]$	$\mu_1 = -15, \mu_2 = -5, \mu_3 = 3, \mu_4 = 6, \mu_5 = 50$, $\sigma_1 = \sigma_2 = \sigma_4 = 1$, $\sigma_3 = 0.5$, $\sigma_5 = 0.1$, $w_1 = 0.15, w_2 = 0.2, w_3 = 0.05, w_4 = 0.1$
$g'_5(x)$	$[-10^2, 10^2]$	same as $g_5(x)$, except $\sigma_1 = \sigma_2 = \sigma_4 = 0.1$, $\sigma_3 = 0.05, \sigma_5 = 0.01$
$g''_5(x)$	$[-10^2, 10^2]$	same as $g_5(x)$, except $\sigma_1 = \sigma_2 = \sigma_4 = 0.01$, $\sigma_3 = 0.005, \sigma_5 = 0.001$
$\widehat{g}_5(x)$	$[-10^{100}, 10^{100}]$	same as $g_5(x)$

5.1 Univariate Gaussian Mixture

We apply MRS to targets whose shape g_n is obtained from finite mixtures of n univariate Gaussian densities truncated over an interval $\mathbb{T}$. The means (μ_i's), standard deviations (σ_i's), weights (w_i's), and domains ($\mathbb{T}$'s) for each of the six targets studied are shown in Table 1.

We can refine the current partition $\mathfrak{T}_\alpha$ and obtain a finer partition $\mathfrak{T}_{\alpha'}$ with an additional labelled box by bisecting a labelled box ${}^{k_*}\boldsymbol{t}_* \in \mathfrak{T}_\alpha$ along the first side with the maximal width. We explored the following three ways to choose a ${}^{k_*}\boldsymbol{t}_*$ from the current partition $\mathfrak{T}_\alpha$ for bisection:

$$\begin{aligned}
&\text{(a) Volume-based} \qquad {}^{k_*}\boldsymbol{t}_* = \operatorname*{argmax}_{{}^{k_j}\boldsymbol{t}_j \in \mathfrak{T}_\alpha} \left(\mathrm{vol}\left({}^{k_j}\boldsymbol{t}_j\right)\right) \ , \\
&\text{(b) Range-based} \qquad {}^{k_*}\boldsymbol{t}_* = \operatorname*{argmax}_{{}^{k_j}\boldsymbol{t}_j \in \mathfrak{T}_\alpha} \left(\mathrm{wid}\left({}^{k_j}\boldsymbol{f}({}^{k_j}\boldsymbol{t}_j)\right)\right) \ , \qquad (26) \\
&\text{(c) Integral-based} \qquad {}^{k_*}\boldsymbol{t}_* = \operatorname*{argmax}_{{}^{k_j}\boldsymbol{t}_j \in \mathfrak{T}_\alpha} \left(\mathrm{vol}\left({}^{k_j}\boldsymbol{t}_j\right) * \mathrm{wid}\left({}^{k_j}\boldsymbol{f}({}^{k_j}\boldsymbol{t}_j)\right)\right) \ ,
\end{aligned}$$

and implemented each refinement scheme through PQ. The volume-based PQ (a) manages the family of partitions $\mathfrak{U}_W$, the range-based PQ (b) manages the family $\mathfrak{R}_\alpha$ and the integral-based PQ (c) manages the family $\mathfrak{V}_\alpha$.

First, we study the efficiency of the three partitioning schemes (26) by Moore rejection sampling from g_5. Figure 2 shows the empirical acceptance probability of MRS, calculated from up to 10^4 draws from a maximum of 10^5 trials, at each partition size $|\mathfrak{T}_\alpha|$ for each of the three different families of partitions ($\mathfrak{U}_W$, $\mathfrak{R}_\alpha$ and $\mathfrak{V}_\alpha$). Thus, for a given partition size $|\mathfrak{T}_\alpha|$, the domain interval $\mathbb{T}$ gets adaptively partitioned through $|\mathfrak{T}_\alpha| - 1$ bisections by the appropriate PQ. The family of partitions $\mathfrak{V}_\alpha$ managed by the integral-based PQ is the most efficient as it can direct the next refining bisection towards the interval with the most uncertainty in its integral estimate. The efficiency of the integral-based scheme is even more pronounced for multivariate exponential mixtures (results not shown).

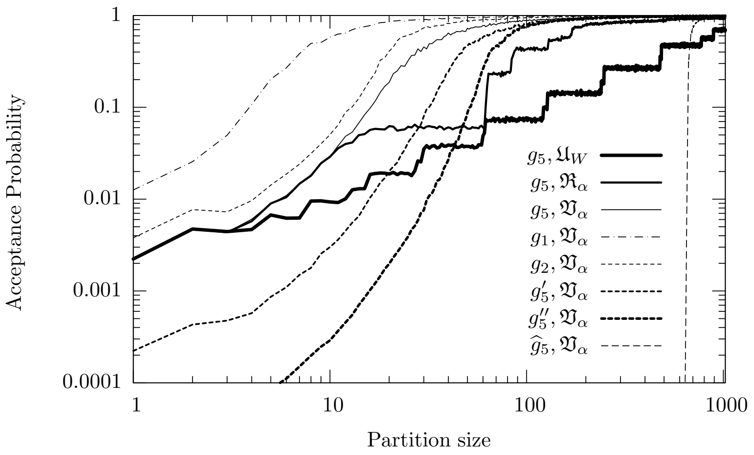

Figure 2: Acceptance probability versus partition size for six target shapes g_1, g_2, g_5, g_5', g_5'' and $\widehat{g}_5$ (Table 1) under different families of partitions: (1) volume-based $\mathfrak{U}_W$, (2) range-based $\mathfrak{R}_\alpha$ and (3) integral-based $\mathfrak{V}_\alpha$ (see text for description).

Note that Theorem 4 guarantees that MRS produces independent draws from any arithmetical expression that is locally Lipschitz on the domain interval $\mathbb{T}$. This includes Gaussian mixture targets with any finite number of components truncated over any compact interval. Furthermore, the locations inside $\mathbb{T}$ are arbitrary and the scales can be highly spiked (i.e., provided $\sigma_i > 0$ and can be enclosed by a machine interval via directed rounding). However, the efficiency of the sampler can depend on (i) number of components, (ii) spikiness of peaks and (iii) domain size. We empirically study these effects by sampling from the six targets (Table 1) using the family of MRSs induced by the most efficient partitions $\mathfrak{V}_\alpha$. The acceptance probability plots (Figure 2) for targets g_1, g_2, and g_5 illustrate decreasing efficiency as the number of components increases, and the plots for targets g_5, g_5' and g_5'', with progressively smaller variances, illustrate a similar effect of spikiness on efficiency at every partition size $|\mathfrak{V}_\alpha|$. Note that in both cases sampler efficiency quickly recovers for larger partition sizes (> 100). Next we study the effect of domain size. In a computer, we cannot represent the real line and are forced to approximate it with the entire number screen, a compact interval. Thus, the domain of any target is necessarily truncated in a machine and we can formally accept it by assuming a uniform prior on the largest machine-representable interval of interest. One can also come to terms with the compact domain due known apriori bounds on possible values of physical quantities. The acceptance probability plot for the target shape $\widehat{g}_5$, that is obtained by extending the domain of g_5 to a large interval of radius 10^{100} centred at 0, shows the effect of domain size. The first 700 bisections or so are spent on homing in on the intervals with relatively higher probability mass. However, by 1000 bisections our acceptance probability is almost 1.

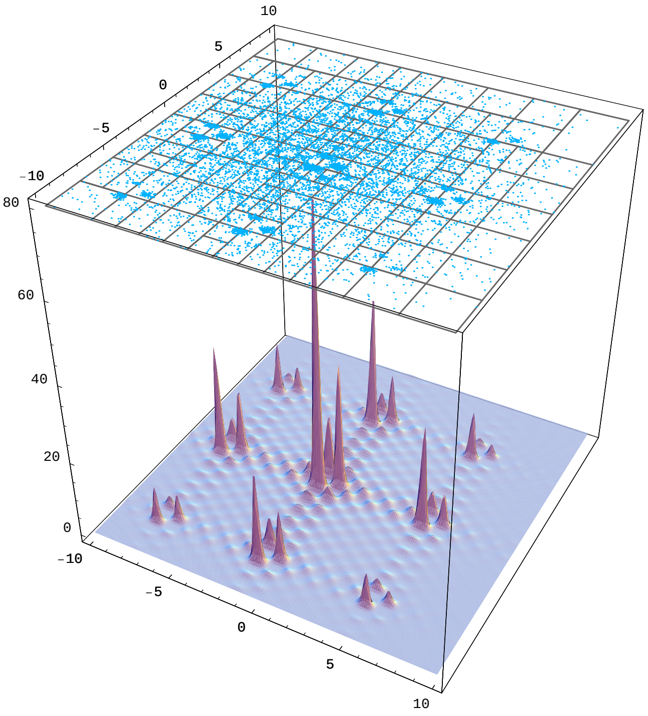

Figure 3: Shape of the Levy density l_{40} with its 700 modes (27). 10^4 samples (points on top) from l_{40} using the MRS induced by an adaptive partitioning of the domain into 150 rectangles (with grey boundaries).

5.2 Bivariate Levy

The bivariate Levy density $l^{\cdot}_r(t_1, t_2)$ over $\mathbb{T} = [-100, 100]^2$ (27) with temperature parameter r and normalising constant N_{l_r} has 700 local maxima and is given by:

$$l^{\cdot}_r(t_1, t_2) = \frac{1}{N_{l_r}} l_r, \text{ where, } l_r = \exp\{-\Lambda(t_1, t_2)/r\}, \tag{27}$$

where,

$$\Lambda(t_1, t_2) = \left(\sum_{i=1}^{5} i \cos\left((i-1)t_1 + i\right) \right) \left(\sum_{j=1}^{5} j \cos\left((j+1)t_2 + j\right) \right) \\ + (t_1 + 1.42513)^2 + (t_2 + 0.80032)^2 .$$

Figure 3 shows l_{40}, i.e., the shape of the Levy density when $r = 40$, and 10^4 samples drawn from l_{40} using the MRS induced by an integral-based adaptive partitioning of the domain into 150 rectangles. This MRS produced 10^4 IID samples in less than 10 CPU seconds at an acceptance probability of about 0.01. Mixtures of bivariate Gaussian shapes yielded comparable results (not shown here).

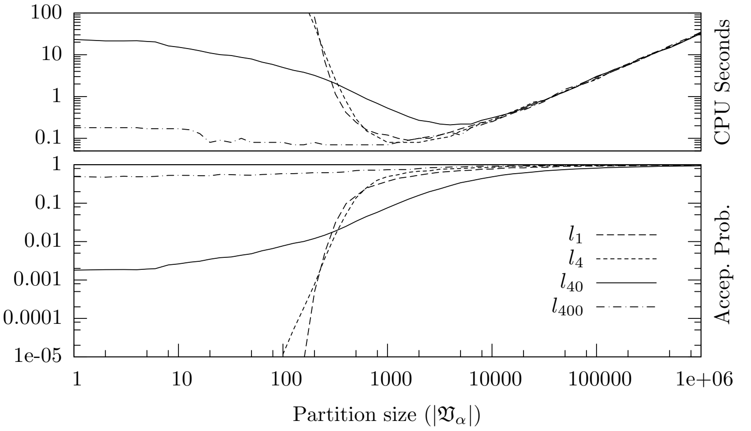

Figure 4: Acceptance probability and CPU seconds versus partition size ($|\mathfrak{V}_\alpha|$) for Levy targets l_r, where r is the temperature (27).

As the temperature parameter r in l_r increases, the density approaches a uniform distribution on $\mathbb{T}$. The density is more peaked at low values of r. Various MCMC methods that use local proposals tend to mix well at higher temperatures and get trapped at local peaks when r is small. To study the effect of temperature on our sampler's efficiency, we plot the empirical acceptance probability as well as the CPU seconds taken to draw 10^4 samples from each of four Levy targets at different temperatures ($r = 1, 4, 40, 400$) as a function of the partition size $|\mathfrak{V}_\alpha|$ (Figure 4). The efficiency decreases as the temperature cools. However, across the range of r we explored, MRS can produce 10^4 independent samples from l_r in a guaranteed manner within 10 CPU seconds with an acceptance probability greater than 1/100. Note that it is difficult to get a Monte Carlo Markov chain to mix properly and even more difficult to rigorously establish convergence for such targets.

5.3 Trivariate Needle in the Haystack

Let $h^{\cdot}(t)$ be an equally weighted mixture of two trivariate Gaussian densities with zero covariance terms and identical variance terms with the following expression for its shape:

$$h(t) = \frac{1}{\sigma_1^3}\exp\{-\frac{1}{2}||(t-\mu_1)/\sigma_1||^2\} + \frac{1}{\sigma_2^3}\exp\{-\frac{1}{2}||(t-\mu_2)/\sigma_2||^2\} \ , \qquad (28)$$

where, $||\cdot||$ is the Euclidean norm. Using this target shape h truncated over $\mathbb{T} = [-10, 10]^3$ with location parameters μ_1 and μ_2 that are distinct in $\mathbb{T}$ and scale parameters σ_1 and σ_2 that differ by at least an order of magnitude in $(0, \infty)$, we compare MRS to a popular MCMC sampler that relies on heuristics for convergence diagnosis and exploit the connection between three Monte Carlo methods.

5.3.1 Metropolis-Hastings Sampler (MHS)

Given $q_Y(t,\cdot)$, a possibly dependent proposal distribution for the base Markov chain Y, one can produce a Monte Carlo Markov chain known as the Metropolis-Hastings (MH) chain on $\mathbb{T}$ merely from the knowledge of ratios of the form $h(t)/h(t')$ for any $(t,t') \in \mathbb{T} \times \mathbb{T}$ such that the stationary distribution of the MH chain is $h^{\cdot}$ [38, 22]. We run a MH chain with local proposal specified by a uniform cube of side $6\sigma_1$ centred at the current state. Using this local Metropolis-Hastings sampler (LMHS) we try to draw samples from the following needle in the haystack, i.e., h in (28) with the following parameters:

$$\mu_1 = (0,0,0)', \mu_2 = (1,1,1)', \sigma_1 = 1, \sigma_2 = 0.006 \ . \tag{29}$$

We run multiple MH chains with randomly dispersed initial conditions and monitor the between-chain variation (B) and within-chain variation (W) of the samples to diagnose convergence heuristically. To diagnose convergence of the LMHS we calculate B/W for each component of t and assume that the chain's burn-in time (the time when the samples may be affected by the initial condition) has ended when $B/W \leq 0.05$ for all three components. The post burn-in run length, i.e., the number of samples kept after the burn-in, is set to be 100 times the burn-in time (typical run lengths ranged in $[1,5] \times 10^4$ for target h specified by (29)). The above convergence diagnostics are more conservative than the standard recommendations [12, 11, 29].

Figure 5 shows the results (along the t_1 axis) of the above LMHS that relies on the B/W statistic from four randomly initialised chains. The running mean for each of the four chains has converged to the haystack mean of $(0,0,0)$ and completely missed the needle at $(1,1,1)$. Thus, if we relied on our convergence diagnostic B/W, which appears to be consistently vanishing and thus suggestive of convergence to our target h, we would have entirely missed the needle. Tuning the diagnostic parameters, including the number of chains, burn-in time, and run length, does not help diagnose true convergence for much sharper needles ($\sigma_2 < 10^{-5}$) that are naturally amenable to our MRS.

Next we compare the samples obtained from the B/W diagnosed LMHS described above with 10^4 samples from MRS induced by an integral-based adaptive partitioning of $\mathbb{T}$ into 10^3 boxes. We compare the two samplers on two targets: (1) a blunt needle with $\sigma_2 = 0.10$ and (2) a sharp needle with $\sigma_2 = 0.01$. The other parameters of the two targets are the same as before (29). The results are summarised in Figure 6. The diagnostic B/W works better in diagnosing convergence to the blunt target. The bias is severe for the sharp needle in all 100 replicates. MRS clearly outperforms LMHS, both in terms of producing the true samples and in terms of CPU time (Figure 6). Moreover, the sharpness of the needle only has a minor effect on the efficiency of MRS. For example, for a much sharper needle with $\sigma_2 = 10^{-10}$, the MRS induced by an integral-based adaptive partitioning of $\mathbb{T}$ into just 120 cuboids, achieves an acceptance probability of 0.40.

5.3.2 Rejection, Importance and Independent MHS

Suppose we are interested in estimating an expectation $E_{f^{\cdot}}(s(T))$, say the mean $E_{f^{\cdot}}(T)$. Importance sampler (IS) [28, 35] is an efficient Monte Carlo method to estimate the desired expectation using importance-weighted sample mean of samples drawn from a density that is close to $\mathrm{abs}(s(t))f^{\cdot}(t)/\left(\int \mathrm{abs}(s(t))f^{\cdot}(t)dt\right)$. The same proposal density used in RS may be used as the proposal in importance sampler (IS)

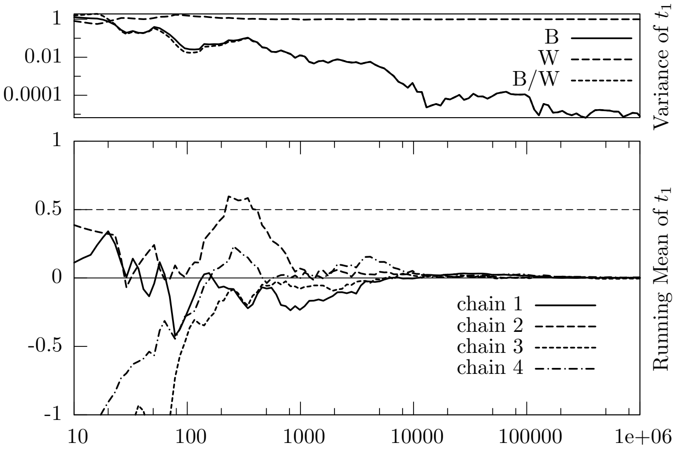

Figure 5: The running mean for four MH chains, as well as B, W, and B/W for t_1 as a function of run length. The true mean for t_1 is at 0.5.

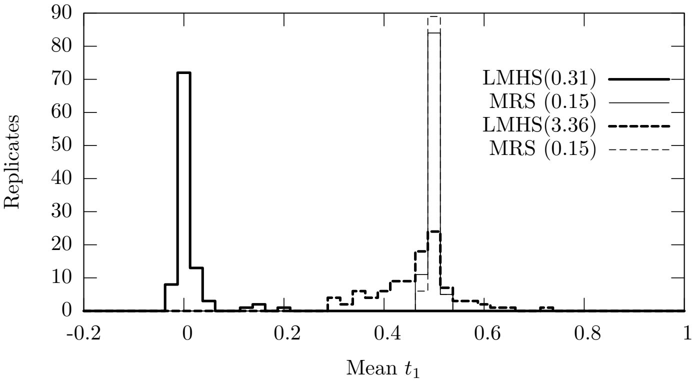

Figure 6: Histograms of the mean t_1 from 100 replicates of the LMHS and MRS. The broken lines and solid lines represent targets with a blunt needle ($\sigma_2 = 0.10$) and a sharp needle ($\sigma_2 = 0.01$), respectively. The CPU time in seconds for each sampler is given in parenthesis.

or as the proposal of the independent base chain in IMHS. The latter two samplers are typically more efficient than RS, although in some cases the efficiency of IMHS can be as low as half that of RS [33]. The disadvantage of IMHS and IS (or RS) compared to MRS is in terms of diagnosing convergence and finding the right proposal(s), respectively. However, if one shares the proposal obtained through interval methods in MRS with IS and IMHS, then we get their Moore versions which circumvent the disadvantages that arise from non-rigorously constructed proposals. Indeed all three samples may be generated simultaneously from the same sequence of proposed values [2]; each proposed value would be output with its importance weight, with some subset of the proposed values marked as IMHS-accepted, and with some further subset of those additionally marked as MRS-accepted and thereby constituting our collection of independent samples.

Let us consider the problem of estimating $E_{h^{\cdot}}(T)$ for the target shape h (28) to illustrate simultaneous sampling from Rejection, Importance and Independent MH samplers. Figure 7 shows the mean squared error MSE for the sampler trio as a function of the size of the partition that is invoking their common proposal. The sample trio is constructed for our target shape h (28) with the sharp needle ($\sigma_2 = 0.01$). The objective is to estimate $E_{h^{\cdot}}(T)$. To obtain the mean squared error (MSE) for each sampler with target $h^{\cdot}$ and proposal g, we drew $t_i \sim g$, $i = 1, \ldots, N$ using MRS, where N is the number of samples needed to obtain 100 Moore rejection samples. For IS, each of the t_i's were assigned the importance sampling weight $w_i = h^{\cdot}(t_i)/g(t_i)$ and the estimated mean $\widehat{\mu} = \left(\sum_{i=1}^{N} (w_i t_i)\right) / \left(\sum_{i=1}^{N} w_i\right)$. The MRS estimated mean is $\widehat{\mu} = \left(\sum_{i=1}^{100} t_{r_i}\right)/100$, where t_{r_i} is the i^{th} MRS sample. For IMHS the mean is estimated by $\widehat{\mu} = \left(\sum_{i=r_1}^{N} t_i\right) / (N - r_1 + 1)$, where r_1 is the index of the first MRS sample; the early samples t_i, $i < r_1$ are excluded as burn-in. This mean estimation was repeated 500 times to obtain $\widehat{\mu}_j$, $j = 1, \ldots, 500$ for each sampler. Finally, the MSE was computed with the known mean $\mu = (0.5, 0.5, 0.5)$ under the Euclidean norm $||\widehat{\mu}_j - \mu||$ as $\left(\sum_j ||\widehat{\mu}_j - \mu||^2\right)/500$.

Figure 7 compares the three samplers and shows a typical pattern: at low acceptance probability, IS has lowest MSE, and MRS the highest, while at high acceptance probability all three samplers approach the same MSE. The lower MSE of IS is due to the large number of (MRS-discarded) samples being appropriately weighted. Observe that such an auto-validating Moore importance sampler can be efficient and rigorous in estimating some expectation of interest. As the acceptance probability of MRS increases with refinement of the domain and the number of samples from each sampler approaches equality, the MSE of all three samplers converge as expected. For some target shapes, e.g., the witches hat (31), we have observed the MSE of IMHS to be greater than that of MRS, but by less than a factor of 2, in agreement with Liu [33] (results not shown).

5.4 Multivariate Rosenbrock

Next we examine the effect of dimensionality on efficiency of MRS through the challenging Rosenbrock function from the optimisation literature. We obtain $r_d^{\cdot}(t)$, the Rosenbrock density in d dimensions over some box $\mathbb{T} \in \mathbb{IR}^d$, by appropriately normal-

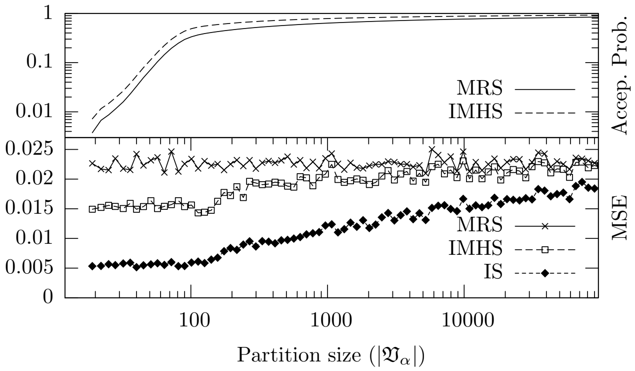

Figure 7: MSE of the three samplers, namely, MRS, IMHS and IS, as well as the acceptance probability of MRS and IMHS as a function of partition size (see text for description).

ising the Rosenbrock shape given by:

$$r_d(t) = \exp\left(-\sum_{i=2}^{d}(100(t_i - t_{i-1}^2)^2 + (1 - t_{i-1})^2)\right) \quad . \tag{30}$$

Figure 8 summarises the efficiency for various Rosenbrock densities. For the more demanding nine-dimensional Rosenbrock target r_9, we were able to draw 10^4 samples in about 650 CPU seconds at an acceptance probability of 10^{-4}. The acceptance probability can be improved and/or d can be increased naively if we allowed the partition size to be greater than a million. Thus, the extent of RAM (random access memory) at our disposal ultimately determines the complexity and dimensionality of the target that can be rigorously sampled with MRS. However, the manner in which the natural interval extension is constructed will greatly affect the sampler's efficiency as discussed later. The acceptance probability for the relatively less complicated multivariate exponential mixture density truncated over $\mathbb{T} = [-100, 100]^{10}$ is higher at 1/1000 compared to that for the Rosenbrock target r_9 even when there were 10 modes inside a 10-dimensional $\mathbb{T}$ (results not shown). Thus, the complexity of the arithmetical expression of the target shape determines sampler efficiency by affecting the sharpness of the range enclosures.

5.5 Multivariate Witch's Hat

Using MRS we can even sample from the infamous witch's hat density which is considered to be a pathological target for most samplers [29]. The density is often thought of in two dimensions as an $m : (1 - m)$ mixture of a cone with centre C and basal radius R and a uniform distribution on a rectangle $\mathbb{T} \in \mathbb{IR}^2$. It can be easily generalised to

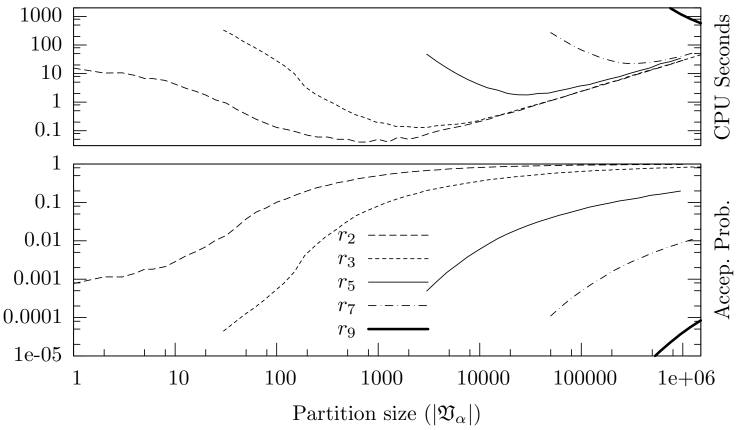

Figure 8: Acceptance probability and CPU time to generate 10^4 samples, as a function of partition size ($|\mathfrak{V}_\alpha|$), for Rosenbrock targets r_d over $\mathbb{T} = [-10, 10]^d$, where d is the dimension.

a d-dimensional box $\mathbb{T} \in \mathbb{IR}^d$ as follows:

$$\begin{aligned} w_r^d(t) =& m\, \mathbb{1}_{\{||t-C|| \leq R\}} \left(1 - \frac{||t - C||}{R}\right) H + (1 - m)\, \frac{1}{\mathrm{vol}\,(\mathbb{T})}, \text{ where} \\ H =& \frac{\Gamma(d/2) d(d+1)}{2\pi^{d/2} R^d}, \quad R = 10^{-r}. \end{aligned} \tag{31}$$

Our formulation of the witch's hat is even more challenging than the differentiable formulation suggested in [29], as the gradient is 0 over the entire brim. MRS is amenable to any target with a well-defined interval extension over the domain including w_r^d. Mixtures of several sharply-peaked bivariate normals with a uniform distribution, a further generalisation of the other formulation [29], pose no sampling problems to MRS. Figure 9 shows that one can efficiently sample from witch's hat targets by rigorously constructing envelopes through the interval evaluation of (31), an arithmetical expression that is locally Lipschitz in $\mathbb{T}$. We can even sample from the hat of an eleven-dimensional witch (w_0^{10}). We can also make the brim of the hat as large as $[-10^{100}, +10^{100}]^2$ without much trouble ($\widehat{w}_0^2$). Note that decreasing the radius has a similar effect as widening the brim, in terms of lowering the acceptance probability as a function of partition size. Thus we are able to sample rigorously from a range of multivariate witch's hat targets with reasonable partition sizes and CPU seconds.

5.6 Posterior of the Rate Parameter in the Stretched Oscillating Exponential Model

Recall the (a, b, c)-parametric family of stretched oscillating exponential shape $f(t)$ (11) with arithmetical expression (12) from Example 3. In this model, parameter $a > 0$ determines the scale, parameter $b \in (0, 1/2)$ determines the stretch and frequency of the oscillations, and parameter $c \in (-1, 1)$ determines the magnitude of the

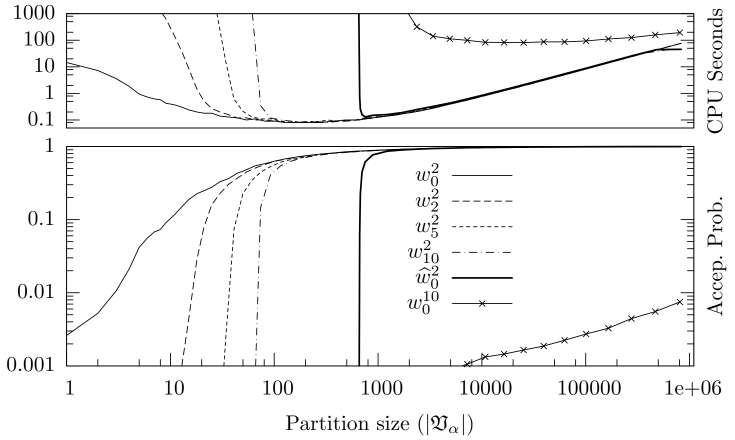

Figure 9: Acceptance probability and CPU time to generate 10^4 samples, versus partition size for witch's hat targets w_r^d, where d is the dimension of the domain and $R = 10^{-r}$ is the hat's radius (31). The hats of all targets were centred at the two vector $(2, \ldots, 2)$. The domain $\mathbb{T}$ for $\widehat{w}_0^2$ was $[-10^{100}, +10^{100}]^2$, but all other targets had $\mathbb{T} = [-10, 10]^d$.

oscillations. We drew twenty samples from this model with $a = 1/8$, $b = 9/20$, $c = 1/2$ and over the uniform prior-specified support $\mathbb{T} = [10^{-12}, 10^{12}]$ using MRS. The first ten of these samples are plotted as 'o' and the next ten are plotted as '+' along with the associated MRS partition and envelope function in Figure 1.

We use MRS to produce samples from the posterior distribution of the parameter a. A uniform prior was assumed for a over the interval $[10^{-3}, 10^0]$ and an adaptive partition of 100 intervals were used. The posterior mean from the first ten samples was 0.1118 while that from the all twenty samples was 0.1173. The CPU time per posterior sample was less than 0.002 seconds for a sampler that produced 10^4 posterior samples. The more informative multimodal histograms from 10^4 IID posterior samples based on 10 and 20 data points are shown in blue and cyan respectively (in bottom panel of Figure 10).

5.7 Trans-Dimensional Posterior Samples Over Binomial Partitions

Let us generalise Example 2 of deciding if the two coins have the same bias. Suppose there are m sets:

$$X_1 := (X_{1,1}, X_{1,2}, \ldots, X_{1,n_1}), \ldots, X_m := (X_{m,1}, X_{m,2}, \ldots, X_{m,n_m}) \ ,$$

of Bernoulli trials that are assumed to be definitely identical and independent within each set and possibly identical but definitely independent across the sets. Only the sum of the j-th set of Bernoulli trials, namely, $Y_j := \sum_{\ell=1}^{n_j} X_{j,\ell}$, is known to the experimenter for each $j = 1, 2, \ldots, m$. So effectively, we have m independent binomial trials $Y_1, Y_2, \ldots, Y_m$ of known sizes $n_1, n_2, \ldots, n_m$, respectively, with possibly distinct

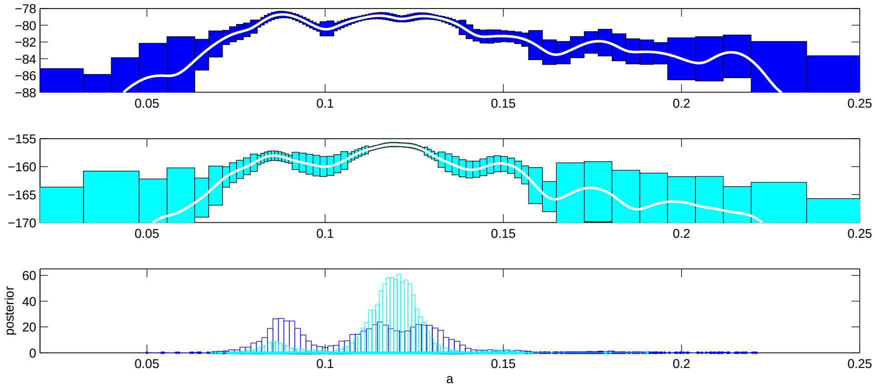

Figure 10: Enclosure of the log-likelihood function (white line) of the parameter a in the stretched oscillating exponential model of Example 3 based on 10 samples (blue rectangles in top panel) and 20 samples (cyan rectangles in middle panel) that were drawn from the model (see text for details) and an adaptive partition of 100 intervals were used. The histograms from 10^4 exact posterior samples based on 10 and 20 data points are shown in blue and cyan respectively (in bottom panel).

success probabilities. Our task is to determine the nature and extent of this distinctness.

We define a *binomial partition model* as follows. Let $\mathfrak{L} := \{1, 2, \ldots, m\}$ be the set of binomial trial labels. Let $\mathbb{C}_m$ be the set of set partitions of $\mathfrak{L}$ and $\mathbb{C}_m^{(\kappa)}$ be the set of set partitions of $\mathfrak{L}$ with κ many blocks, where $\kappa \in \{1, 2, \ldots, m\}$. Thus, $\mathbb{C}_m = \bigcup_{\kappa=1}^{m} \mathbb{C}_m^{(\kappa)}$, $|\mathbb{C}_m| = B_m$ and $|\mathbb{C}_m^{(\kappa)}| = S_m^{(\kappa)}$, where B_m is the m-th Bell number, i.e., the number of partitions of a set with m elements, and $S_m^{(\kappa)}$ is the Stirling number of the second kind, i.e., the number of ways to partition a set of m elements into κ nonempty subsets. For each given *canonically ordered set partition* $k := (k_1, k_2, \ldots, k_\kappa) \in \mathbb{C}_m^{(\kappa)}$, let us define its partition-specific model such that the success probability is identical between the canonically ordered trial labels within each of its κ many blocks $k_1, k_2, \ldots, k_\kappa$ of possibly distinct sizes $|k_1|, |k_2|, \ldots, |k_\kappa|$ and independent both within and between the blocks. For example, when $m = 2$ with $\mathfrak{L} = \{1, 2\}$ there are only two partitions, say $k = (k_1) = ((1, 2))$ and $k' = (k'_1, k'_2) = ((1), (2))$, then $k_{1,1} = 1, k_{1,2} = 2$ and $k'_{1,1} = 1, k'_{2,1} = 2$. For each given partition k, the vector of parameters ${}^{k}t := ({}^{k}t_{k_1}, {}^{k}t_{k_2}, \ldots, {}^{k}t_{k_{|k|}})$ that specify the block-specific success probabilities belongs to the $|k|$-dimensional vector of unit intervals or unit hyper-cube. Thus, the parameter space of the trans-dimensional binomial probability model over all B_m partitions is:

$$ {}^{\mathbb{C}_m}\mathbb{T} := \left\{ {}^{k}\mathbb{T} : k \in \mathbb{C}_m \right\}, \quad {}^{k}\mathbb{T} := \left({}^{k}\mathbb{T}_{k_1}, {}^{k}\mathbb{T}_{k_2}, \ldots, {}^{k}\mathbb{T}_{k_{|k|}} \right) = [0, 1]^{|k|} \ . $$

Suppose the realised number of successes in the m binomial trials of size $n := (n_1, n_2, \ldots, n_m)$ is $y := (y_1, y_2, \ldots, y_m)$. Our goal is to sample exactly from the trans-dimensional posterior distribution ${}^{\mathbb{C}_m}f^{\cdot}({}^{k}t|y)$ over the parameter space ${}^{\mathbb{C}_m}\mathbb{T}$. Let the

Table 2: Posterior Results from Binomial Partition Model for Mortality of Pine Seedlings ($\hat{P}(k|y)$ is given by (33)).

no.	k	$\hat{P}(k\|y)$	posterior sample mean in ${}^k\mathbb{T}$
0	$((1,2,3,4))$	0.0000000	none
1	$((1),(2,3,4))$	0.5548453	$(0.5882, 0.9039)$
2	$((2),(1,3,4))$	0.0000000	none
3	$((3),(1,2,4))$	0.0000000	none
4	$((4),(1,2,3))$	0.0000000	none
5	$((1,2),(3,4))$	0.0000030	$(0.7444, 0.9053)$
6	$((1,3),(2,4))$	0.0000106	$(0.7256, 0.9187)$
7	$((1,4),(2,3))$	0.0000000	none
8	$((1),(2),(3,4))$	0.0647222	$(0.5882, 0.8823, 0.9109)$
9	$((1),(3),(2,4))$	0.0946800	$(0.5881, 0.8725, 0.9158)$
10	$((1),(4),(2,3))$	0.2562380	$(0.5882, 0.9411, 0.8811)$
11	$((2),(3),(1,4))$	0.0000000	none
12	$((2),(4),(1,3))$	0.0000035	$(0.8839, 0.9439, 0.7250)$
13	$((3),(4),(1,2))$	0.0000011	$(0.8586, 0.9452, 0.7445)$
14	$((1),(2),(3),(4))$	0.0294963	$(0.5884, 0.8823, 0.8724, 0.9411)$

prior density over the success probability vector ${}^k t \in {}^k\mathbb{T} = [0,1]^{|k|}$ be the product of Beta densities

$$^k q(^k t) = \mathbb{1}_{^k\mathbb{T}}(^k t) \prod_{b=1}^{|k|} \frac{\Gamma\left(^k\alpha_b + {}^k\beta_b\right)}{\Gamma\left(^k\alpha_b\right)\Gamma\left(^k\beta_b\right)} (^k t_{k_b})^{^k\alpha_b - 1}(1 - {}^k t_{k_b})^{^k\beta_b - 1}$$

that are specified by the shape parameter vectors

$$(^k\alpha, {}^k\beta) := \left(\left(^k\alpha_1, {}^k\alpha_2, \ldots, {}^k\alpha_{|k|}\right), \left(^k\beta_1, {}^k\beta_2, \ldots, {}^k\beta_{|k|}\right)\right) \in (0,\infty)^{|k|\times 2} \quad .$$

For ease of interpretation, we set all the values of each $\left(^k\alpha, {}^k\beta\right)$ to be 1 and thus let the uniform density that assigns Lebesgue measure over each unit hyper-cube ${}^k\mathbb{T}$ be our prior density ${}^k q(^k t) = \mathbb{1}_{^k\mathbb{T}}(^k t) \ll \lambda^{|k|}$ and let $(p_k : k \in \mathbb{C}_m)$ be the model prior probabilities. We let $p_k = 1/|\mathbb{C}_m|$ in this study. Finally, the posterior density is proportional to

$$^{\mathbb{C}_m}f(^k t) = \sum_{k \in \mathbb{C}_m} p_k \prod_{j=1}^{m} \binom{n_j}{y_j} \prod_{b=1}^{|k|} (^k t_{k_b})^{\sum_{\ell=1}^{|k_b|} y_{k_b,\ell}} (1 - {}^k t_{k_b})^{\sum_{\ell=1}^{|k_b|} n_{k_b,\ell} - y_{k_b,\ell}} \mathbb{1}_{^k\mathbb{T}}(^k t) \quad . \tag{32}$$

Let us apply our sampler to produce IID samples from the pine seedling mortality data 59,89,88,95 in [3] based on 100 trials each. Let the partition of trial labels $\mathfrak{L} = \{1,2,3,4\}$ corresponding to $\{\mathsf{LH}, \mathsf{LD}, \mathsf{SH}, \mathsf{SD}\}$ for the variety of the pine seedling (L, longleaf; S, slash) and the planting depth (H, planting too high; D, planting too deep). We produce the first exact trans-dimensional samples for such a data set over the fifteen trans-dimensional models indexed by the partitions of the trial label set

$\{1, 2, 3, 4\}$ and summarise the results in Table 2. The model label or partition k is given in the second column and the asymptotically normal point estimate for the posterior model probability $P(k|y)$ is given in the third column of Table 2 and computed from 10^7 IID samples from ${}^{\mathbb{K}}f^{\cdot}$ as follows:

$$\hat{P}(k|y) = 10^{-7} \sum_{{}^k t} \mathbb{1}_{{}^k\mathbb{T}}({}^k t), \quad {}^k t \sim {}^{\mathbb{K}}f^{\cdot} \quad . \tag{33}$$

One can easily obtain 95% confidence interval for $P(k|y)$ from its point estimate (when its point estimate is 0, we can obtain an enclosure of the 95% confidence interval) as well as the sample variance-covariance matrix from the posterior samples (results not shown). What is interesting is that more than half of the posterior mass is in the two-dimensional model with partition $((1), (2, 3, 4))$, i.e., $((\mathsf{LH}), (\mathsf{LD}, \mathsf{SH}, \mathsf{SD}))$. The three-dimensional model with the next highest posterior mass of 0.256 corresponds to the partition $((1), (4), (2, 3))$, i.e., $((\mathsf{LH}), (\mathsf{SD}), (\mathsf{LD}, \mathsf{SH}))$. The four-dimensional model with the finest partition $((1), (2), (3), (4))$, i.e., $((\mathsf{LH}), (\mathsf{LD}), (\mathsf{SH}), (\mathsf{SD}))$, only has 0.029 of the posterior probability mass. Our results, despite the differences in model prior and formulation, are in general agreement with those drawn from the dependent samples from the reversible jump MCMC sampler of [18, Table 1] and with those obtained from an approximation and quadrature in [3, Table 3 and Table 4].

The computations to produce 10^7 IID samples from the posterior were done on a MacBook Pro laptop with OS X 10.5.8, 2.4 GHz Intel processor and 2 GB RAM. The total number of interval and real function calls were 1999985 and 19165849, respectively. The total number of seconds taken for constructing the partition and producing all 10^7 samples were 35.4702 and 3564.56, respectively. Thus the total amount of time was about an hour.Our approach is flexible and allows one to draw exact trans-dimensional posterior samples on any subset of partition models as specified by the model priors. This is particularly helpful for cases with $5 \leq m \leq 10$ and too many partitions. We expect the sampler to return no samples when the dimension of the parameter space gets large due to extremely large over-enclosures of the range. Currently MRS is the only available exact trans-dimensional sampler for binomial partition models.

5.8 Trans-Dimensional Phylogenetic Posterior Samples

In this section we briefly review phylogenetic estimation and solve an open problem in trans-dimensional phylogenetic sampling. Introduction to phylogenetics can be found in [48, 8, 53]. Inferring the ancestral relationship among a set of extant (presently surviving) species based on their DNA sequences is a basic problem in phylogenetic estimation. A phylogenetic tree relates the extant species represented by its leaf nodes with ancestral species represented by its internal nodes. The length of an edge (branch length) connecting two nodes (species) in the phylogenetic tree represents the amount of evolutionary time (divergence) between the two species as measured by the differences in their DNA sequence due to mutation. One can obtain the likelihood of a particular phylogenetic tree that relates the extant species of interest at its leaves by superimposing a continuous time Markov chain model of DNA mutation along the lengths of the branches on that tree. During the likelihood computation, one needs to sum over all possible states of the DNA sequence at the unobserved ancestral nodes. In [45] MRS was used to draw IID posterior samples from small phylogenetic tree spaces of the same dimension (number of branches) based on primate DNA sequence data. Here we generalise this to the trans-dimensional setting where the number of branch length parameters are allowed to vary between models of phylogenetic trees.

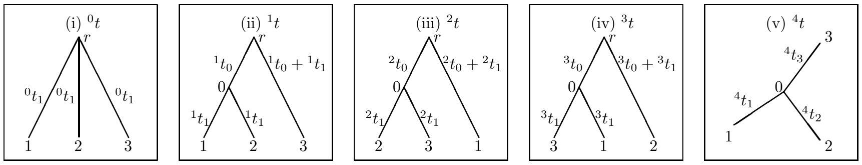

Figure 11: Space of phylogenetic trees with three labelled leaves $\{1, 2, 3\}$. See text for description.

A phylogenetic tree can be rooted or unrooted. In a rooted phylogenetic tree, exactly one internal node is identified as the root node. This root node (labelled r) usually denotes the most recent common ancestor of the species represented by the leaf nodes. The rooted tree is conventionally depicted with the root node r at the top. Let the three leaf nodes that represent extant species be labelled 1, 2 and 3. The topology or shape of the tree specifies the order of speciation or branching events. For example, the rooted trifurcating star-tree $^0t := (^0t_1)$ has topology (model) label 0 and common branch-length parameter $^0t_1$ as shown in Figure 11(i). The topology of such star-trees specifies an instant speciation of the three extant species from their common ancestor at the only internal node r. In contrast, the set of rooted bifurcating trees corresponding to (ii),(iii) and (iv) of Figure 11 have topology labels 1, 2 and 3, respectively. For example, the topology (labelled 1) of binary trees in Figure 11(ii) specifies that species 1 and 2 share a common ancestor at internal node 0 while the topology (labelled 2) of binary trees in Figure 11(iii) specifies that species 2 and 3 share a common ancestor at internal node 0. The unrooted tree with topology label 4 or the unrooted triplet ^4t is shown in Figure 11(v). As opposed to the star-tree ^0t, the unrooted triplet $^4t := (^4t_1, {}^4t_2, {}^4t_3)$, can have distinct branch length parameters $^4t_1$, $^4t_2$ and $^4t_3$. For each tree, the terminal branch lengths, i.e., the branch lengths leading to the leaf nodes, have to be strictly positive and the internal branch lengths have to be non-negative. The trees in Figure 11(ii)–(iv) are said to satisfy the *molecular clock*, since the branch lengths of each ^kt, where $k \in \{1, 2, 3\}$, satisfy the constraint that the distance from the root node r to each of the leaf nodes is equal to $^kt_0 + {}^kt_1$ with $^kt_1 > 0$ and $^kt_0 \geq 0$. The star-tree also satisfies the molecular clock since the distance from the only internal node r to each leaf node is identically $^0t_1$.

The simplest model for the evolution of binary sequences under a symmetric transition matrix over all branches of a tree is referred to as the Cavender-Farris-Neyman (CFN) model. Under the CFN mutation model, only pyrimidines and purines, denoted respectively by $\mathtt{Y} := \{\mathtt{C}, \mathtt{T}\}$ and $\mathtt{R} := \{\mathtt{A}, \mathtt{G}\}$, are distinguished as evolutionary states among the four nucleotides $\{\mathtt{A}, \mathtt{G}, \mathtt{C}, \mathtt{T}\}$. Time t is measured by the expected number of substitutions in this continuous time Markov chain with rate matrix Q and transition probability matrix $P(t) = e^{Qt}$:

$$Q = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} , \qquad P(t) = \begin{pmatrix} 1 - (1 - e^{-2t})/2 & (1 - e^{-2t})/2 \\ (1 - e^{-2t})/2 & 1 - (1 - e^{-2t})/2 \end{pmatrix} .$$

Thus, the probability that $\mathtt{Y}$ mutates to $\mathtt{R}$, or vice versa, in time t is $a(t) := (1 - e^{-2t})/2$. The stationary distribution is uniform on $\{\mathtt{Y}, \mathtt{R}\}$, i.e., $\pi(\mathtt{R}) = \pi(\mathtt{Y}) = 1/2$. For closely related species with at most two distinct nucleotides per site one can apply the CFN model directly to site patterns of nucleotides as opposed to pyrimidines and purines.

Consider the unrooted tree-space with a single topology labelled 4 and three non-negative terminal branch-lengths ${}^4t = ({}^4t_1, {}^4t_2, {}^4t_3) \in \mathbb{R}_+^3$ as shown in Figure 11(v). The well-known product-sum algorithm in [7] gives the likelihoods of four minimally sufficient site pattern classes, namely, `xxx`, `xxy`, `yxx` and `xyx`, where `x` and `y` simply denote distinct characters in $\{\mathtt{R}, \mathtt{Y}\}$. The corresponding likelihoods are:

$$\begin{aligned}
l_{\mathtt{xxx}}({}^4t) := l_0({}^4t) = l_1({}^4t) &= \frac{1}{8}\left(1 + e^{-2({}^4t_1 + {}^4t_2)} + e^{-2({}^4t_2 + {}^4t_3)} + e^{-2({}^4t_1 + {}^4t_3)}\right) \\
l_{\mathtt{xxy}}({}^4t) := l_2({}^4t) = l_3({}^4t) &= \frac{1}{8}\left(1 + e^{-2({}^4t_1 + {}^4t_2)} - e^{-2({}^4t_2 + {}^4t_3)} - e^{-2({}^4t_1 + {}^4t_3)}\right) \\
l_{\mathtt{yxx}}({}^4t) := l_4({}^4t) = l_5({}^4t) &= \frac{1}{8}\left(1 - e^{-2({}^4t_1 + {}^4t_2)} + e^{-2({}^4t_2 + {}^4t_3)} - e^{-2({}^4t_1 + {}^4t_3)}\right) \\
l_{\mathtt{xyx}}({}^4t) := l_6({}^4t) = l_7({}^4t) &= \frac{1}{8}\left(1 - e^{-2({}^4t_1 + {}^4t_2)} - e^{-2({}^4t_2 + {}^4t_3)} + e^{-2({}^4t_1 + {}^4t_3)}\right) \quad (34)
\end{aligned}$$

Therefore, given a multiple sequence alignment data d from three taxa at v homologous sites, i.e., $d \in \{\mathtt{Y}, \mathtt{R}\}^{3 \times v}$, the product likelihood function across sites over the tree space ${}^k\mathbb{T}$ can be obtained from the minimal sufficient site pattern counts $c := (c_{\mathtt{xxx}}, c_{\mathtt{xxy}}, c_{\mathtt{yxx}}, c_{\mathtt{xyx}})$ as follows:

$$l_d({}^kt) = \prod_{q=1}^{v} l_{d_{\bullet,q}}({}^kt) = \prod_{\mathtt{s}=\mathtt{xxx},\mathtt{xxy},\mathtt{yxx},\mathtt{xyx}} \left(l_{\mathtt{s}}({}^kt)\right)^{c_{\mathtt{s}}} \quad . \tag{35}$$

We compute the topology-specific likelihood functions, i.e., $l({}^kt)$ for $k \in \{0, 1, 2, 3\}$ (Figure 11) by substituting the constraints imposed by the molecular clock upon branch-lengths in ${}^4\mathbb{T} = \mathbb{R}_+^3$, the space of unrooted triplets. Thus, the parameter space for the bifurcating trees with model or topology label $k \in \{1, 2, 3\}$ is ${}^k\mathbb{T} = \mathbb{R}_+^2$ and that for the star tree is ${}^0\mathbb{T} = \mathbb{R}_+^1$. Finally, the posterior distribution is obtained by normalising the likelihood with a uniform prior over the biologically meaningful compact domain $[10^{-10}, 10]$ for each branch-length parameter. The prior model probabilities are taken to be discrete uniform over the five models in $\mathbb{K} = \{0, 1, 2, 3, 4\}$.

Table 3: Posterior Results from Trans-dimensional Phylogenetic Tree Spaces of Three Apes.

$\{1,2,3\}$, c	$\hat{P}({}^0\mathbb{T}\|d)$	$\hat{P}({}^1\mathbb{T}\|d)$	$\hat{P}({}^2\mathbb{T}\|d)$	$\hat{P}({}^3\mathbb{T}\|d)$	$\hat{P}({}^4\mathbb{T}\|d)$	${}^{\hat{k}}\hat{t}$
$\{H,C,G\}$, $(884,6,2,3)$	0.999764 (4.86e-6)	0.000191 (4.37e-6)	0.0000179 (1.33e-6)	0.0000271 (1.65e-6)	0.0 (1.0e-7)	${}^0\hat{t} = (0.004509)$
$\{H,C,O\}$, $(858,32,3,2)$	0.0000015 (3.87e-7)	0.9999176 (2.87e-6)	0.0 (1.0e-7)	0.0 (1.0e-7)	0.0000809 (2.84e-6)	${}^1\hat{t} = (0.003378,$ $0.035186)$
$\{H,C,B\}$, $(848,42,3,2)$	0.0 (1.0e-7)	0.9999221 (2.79e-6)	0.0 (1.0e-7)	0.0 (1.0e-7)	0.0000779 (2.79e-6)	${}^1\hat{t} = (0.003377,$ $0.047488)$
$\{H,G,O\}$, $(857,30,5,3)$	0.000531 (7.29e-6)	0.9993588 (8.01e-6)	0.0000001 (1.0e-7)	0.0 (1.0e-7)	0.0001101 (3.32e-6)	${}^1\hat{t} = (0.005085,$ $0.031166)$
$\{H,G,B\}$, $(846,41,4,4)$	0.0 (1.0e-7)	0.9999075 (3.04e-6)	0.0 (1.0e-7)	0.0 (1.0e-7)	0.0000925 (3.04e-6)	${}^1\hat{t} = (0.005085,$ $0.044700)$
$\{H,O,B\}$, $(829,31,14,21)$	0.9973552 (1.62e-5)	0.0025385 (1.60e-5)	0.0000295 (1.72e-6)	0.000076 (2.76e-6)	0.0000008 (2.83e-7)	${}^0\hat{t} = (0.026290)$

Table 3 summarises the results of a trans-dimensional Bayesian analysis based on sets of three mitochondrial DNA sequences [1] from five apes: humans (H), chimpanzee

(C), gorilla (G), orangutan (O) and gibbon (B). Column 1 gives the species label set $\{1, 2, 3\}$ and c, the site pattern counts of pyrimidines and purines from the multiple sequence alignment data d. Columns 2–6 give $\hat{P}({}^k\mathbb{T}|d)$, the point estimate (standard error) of the posterior probability of model $k \in \{0, 1, 2, 3, 4\}$. When this point estimate is 0.0 the upper bound for the standard error is reported from the 10^7 exact trans-dimensional posterior samples. The last column gives ${}^{\hat{k}}\hat{t}$, the posterior mean branch-lengths for the model with the highest posterior probability. One can obtain other statistics such as credible sets from the exact posterior samples (not shown).

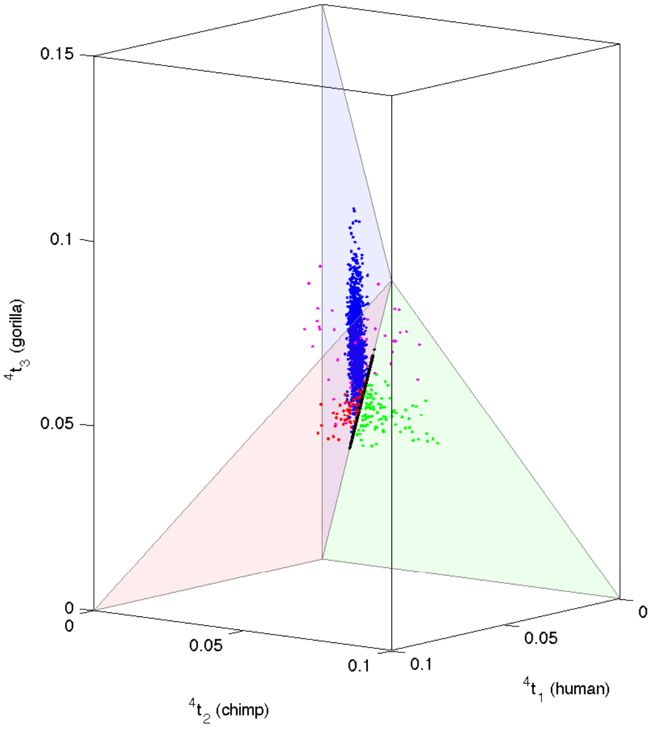

Figure 12: Exact trans-dimensional posterior samples from the phylogenetic tree space of human, chimpanzee and gorilla on the basis of $(c_{\mathrm{xxx}}, c_{\mathrm{xxy}}, c_{\mathrm{yxx}}, c_{\mathrm{xyx}}) = (762, 54, 38, 41)$ with topology label set $\{0, 1, 2, 3, 4\}$ (see Figure 11(i),(ii),(iii),(iv),(v)) (black, blue, red, green and magenta dots, respectively).

Figure 12 shows ten thousand exact trans-dimensional posterior samples from the phylogenetic tree space of human, chimpanzee and gorilla with topology or model label set $\{0, 1, 2, 3, 4\}$ (see Figure 11(i),(ii),(iii),(iv),(v)) on the basis of mitochondrial data [1] summarised by $(c_{\mathrm{xxx}}, c_{\mathrm{xxy}}, c_{\mathrm{yxx}}, c_{\mathrm{xyx}}) = (762, 54, 38, 41)$ under the Cavender-Farris-Neyman model that distinguishes dimorphic nucleotides (black, blue, red, green, magenta dots, respectively) are depicted in Figure 12. The posterior probability estimates (standard errors) on the basis of 10^7 exact samples for the five models in $\mathbb{K} = \{0, 1, 2, 3, 4\}$ are 0.8679336(0.0001071), 0.1136644(0.0001004), 0.0061397(0.0000247), 0.0083094(0.0000287), 0.0039529(0.0000198), respectively. The CPU time to produce 10^7 IID samples for the seven sets of three ape sequences ranged between 15 and 50 seconds on a MacBook Pro laptop with OS X 10.5.8, 2.4 GHz Intel processor and 2 GB RAM.

MRS has produced the first exact trans-dimensional posterior samples over three-taxa phylogenetic tree spaces. The significantly higher posterior probability for the star-tree with topology 0 for $\{H, C, G\}$ and $\{H, O, B\}$ is a persistent pattern in a recent genomic dataset of multiply aligned ape genomic. This biologically interesting phenomenon deserves a systematic investigation across the sample space of $(c_{\mathrm{xxx}}, c_{\mathrm{xxy}}, c_{\mathrm{yxx}}, c_{\mathrm{xyx}})$ using set-valued methods.

6 Conclusion

In this paper, we make the first formal trans-dimensional extension of von Neumann's rejection sampler. We use interval methods to automatically and rigorously construct envelope functions for universal trans-dimensional rejection sampling from possibly non-normalised target densities whose arithmetical expressions are locally Lipschitz over their support. In particular the method allows the envelope to be drawn from a large, flexible family of functions (simple functions over a family of adaptively refined partitions), and to be constructed in a manner that rigorously maintains the envelope property as the envelope function is adaptively refined. Refining the partition decreases the rejection probability at a rate that is no slower than linear with the mesh. The corresponding proposal density is easily constructed in $\mathcal{O}$(partition size) time into a data structure that allows samples from it to be drawn in constant time. When one substitutes conventional floating-point arithmetic for real arithmetic in a computer and uses discrete lattices to construct the envelope and/or proposal, it is generally not possible to guarantee the envelope property and thereby ensure that samples are drawn from the desired target density, except in special cases.

Unfortunately, the efficiency of MRS is not immune to the curse of dimensionality and the complexity of the target arithmetical expression. When the arithmetical expression gets large, its interval extension can have terrible over-enclosures of the true range, which in turn forces the adaptive refinement of the domain to be extremely fine for efficient envelope construction. Thus, a naive application of interval methods to large targets can be terribly inefficient. In such cases, sampler efficiency rather than rigour is the issue. Thus, one will not obtain samples in a reasonable time rather than produce samples from some unknown and undesired target. There are several ways in which efficiency can be improved for such cases. First, the particular structure of the target expression should be exploited to avoid any redundant computations. For example, algebraic statistical methods can be used to find sufficient statistics to dissolve symmetries [46] or other enclosure techniques such as affine arithmetic can be used [6]. Second, we can further improve efficiency by limiting ourselves to differentiable targets in C^n. Tighter enclosures of the range can come from the enclosures of Taylor expansions around the midpoint through interval-extended automatic differentiation [20, 30] that can then yield tighter estimates of the integral enclosures. Third, we can employ pre-processing to improve efficiency. For example, we can pre-enclose the range over a partition of the domain and then obtain the enclosure through a combination of hash access and hull operations on the pre-enclosures. These tighter range enclosures can be represented efficiently by a multi-dimensional metric data-structure called a regular sub-paving [26]. Such a pre-enclosing technique reduces not only the overestimation of target shapes with large expressions but also the computational cost incurred while performing interval operations with processors that are optimised for floating-point arithmetic. Fourth, interval constraint propagation is a powerful technique to obtain tighter range enclosures [47] and can dramatically increase sampler efficiency. Fifth, efficiency at the possible cost of rigour can also be gained (up to 30%) by foregoing directed rounding during envelope construction. It would be interesting to study set-valued extensions of other Monte Carlo methods. Nonetheless, we have demonstrated that for a diverse set of examples in up to 10 dimensions our sampler is currently the only existing exact sampler.

Acknowledgements

This work was partly supported by a joint NSF-NIGMS grant DMS-02-01037, a Research Fellowship of the Royal Commission for the Exhibition of 1851 and a Sabbatical grant from the University of Canterbury. Many thanks to Mohan Delampady for discussions on Bayesian model selection, to Richard Everitt, Peter Green and Gareth Roberts for discussions on Section 2 and the interplay of Monte Carlo methods and interval analysis, to James Booth, Rob Strawderman and Warwick Tucker for comments on an earlier draft, to Ziheng Yang for guidance on the data analysis in Section 5.8, to Nicholas Duncan for comments on a later draft and to two anonymous referees for their careful review.

References

[1] W. M. Brown, E. M. Prager, A. Wang, and A. C. Wilson. Mitochondrial DNA sequences of primates, tempo and mode of evolution. *J. Mol. Evol.*, 18:225–239, 1982.

[2] H. Cai. Exact sampling using auxiliary variables. *Statistical Computing Section, ASA Proceedings*, 1999.

[3] G. Consonni and P. Veronese. A Bayesian method for combining results from several binomial experiments. *Journal of the American Statistical Association*, 90(431):pp. 935–944, 1995.

[4] L. Devroye. *Non-Uniform Random Variate Generation.* Springer-Verlag, New York, 1986.

[5] L. Devroye. Random variate generation for multivariate unimodal densities. *ACM Trans. Model. Comput. Simul.*, 7(4):447–477, 1997.

[6] R. G. Everitt. *Using the autocorrelation time and auto-validating methods to improve the performance of Monte Carlo algorithms.* PhD thesis, University of Bristol, Bristol, UK, 2008.

[7] J. Felsenstein. Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol.*, 17:368–376, 1981.

[8] J. Felsenstein. *Inferring phylogenies.* Sinauer Associates, Sunderland, MA, 2003.

[9] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, M. Booth, and F. Rossi. *GNU Scientific Library Reference Manual - 2nd Ed.* Network Theory Ltd., 2003.

[10] A. E. Gelfand and A. F. M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409, 1990.

[11] A. Gelman. Inference and monitoring convergence. In W.R. Gilks, S. Richardson, and D.J. Spiegelhalter, editors, *Markov Chain Monte Carlo in Practice*, page 137. Chapman and Hall, 1996.

[12] A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences (with duscussion). *Statistical Science*, 7:457–511, 1992.

[13] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Trans. Pattn. Anal. Mach. Intell.*, 6:721–741, 1984.

[14] W. R. Gilks. Derivative-free adaptive rejection sampling for Gibbs sampling. In J Bernardo, J Berger, AP Dawid, and AFM Smith, editors, *Bayesian Statistics 4*. Oxford Univ. Press, 1992.

[15] W. R. Gilks, N. G. Best, and K. C. Tan. Adaptive rejection Metropolis sampling. *Applied Statistics*, 44:455–472, 1995.

[16] W. R. Gilks and P. Wild. Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*, 41:337–348, 1992.

[17] D. Görür and Y. W. Teh. Concave-convex adaptive rejection sampling. *Journal of Computational and Graphical Statistics*, 20(3):670–691, 2011.

[18] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[19] T. C. Hales. A proof of the Kepler conjecture. *Annals of Mathematics*, 162:1065–1185, 2005.

[20] R. Hammer, M. Hocks, U. Kulisch, and D. Ratz. *C++ Toolbox for Verified Computing: Basic Numerical Problems*. Springer-Verlag, 1995.

[21] J. Hass and R. Schlafly. Double bubbles minimize. *Annals of Mathematics*, 151(2):459–515, 2000.

[22] W. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[23] W. Hofschuster and W. Krämer. C-XSC 2.0: A C++ library for extended scientific computing. In R. Alt, A. Frommer, R.B. Kearfott, and W. Luther, editors, *Numerical Software with Result Verification*, volume 2991 of *Lecture Notes in Computer Science*, pages 15–35. Springer-Verlag, 2004.

[24] W. Hörmann. A rejection technique for sampling from T-concave distributions. *ACM Trans. Math. Softw.*, 21(2):182–193, 1995.

[25] W. Hörmann. Algorithm 802: an automatic generator for bivariate log-concave distributions. *ACM Trans. Math. Softw.*, 26(1):201–219, 2000.

[26] L. Jaulin, M. Kieffer, O. Didrit, and É. Walter. *Applied Interval Analysis: With Examples in Parameter and State Estimation, Robust Control and Robotics.* Springer-Verlag, 2001.

[27] G. Jones and J. Hobert. Honest exploration of intractable probability distributions via Markov chain Monte Carlo. *Statistical Science*, 16(4):312–334, 2001.

[28] H. Kahn and A. W. Marshall. Methods of reducing sample size in Monte Carlo computations. *Journal of the Operational Research Society of America*, 1:263–271, 1953.

[29] R. E. Kass, B. P. Carlin, A. Gelman, and R. M. Neal. Markov chain Monte Carlo in practice: a round table discussion. *The American Statistician*, 52:93–100, 1998.

[30] U. Kulisch. Advanced arithmetic for the digital computer, interval arithmetic revisited. In U. Kulisch, R. Lohner, and A. Facius, editors, *Perspectives on encolsure methods*, pages 50–70. Springer-Verlag, 2001.

[31] O. E. Lanford. A computer-assisted proof of the Feigenbaum conjectures. *Bull. Amer. Math. Soc. (N.S.)*, 6(3):427–434, 1982.

[32] J. Leydold. A rejection technique for sampling from log-concave multivariate distributions. *ACM Trans. Model. Comput. Simul.*, 8(3):254–280, 1998.

[33] J. Liu. Metropolised independent sampling and comparisons to rejection sampling and importance sampling. *Statist. and Comput.* , 6:113–119, 1995.

[34] G. Marsaglia. Generating discrete random numbers in a computer. *Comm ACM*, 6:37–38, 1963.

[35] A. W. Marshall. The use of multi-stage sampling schemes in Monte Carlo computation. In M Meyer, editor, *Symposium on Monte Carlo methods*, pages 123–140. Wiley, 1956.

[36] L. Martino and J. Míguez. Generalized rejection sampling schemes and applications in signal processing. *Signal Process.*, 90(11):2981–2995, 2010.

[37] L. Martino and J. Míguez. A generalization of the adaptive rejection sampling algorithm. *Statistics and Computing*, 21:633–647, 2011.

[38] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys.* , 21:1087–1092, 1953.

[39] R. E. Moore. *Interval analysis.* Prentice-Hall, 1967.

[40] A. Neumaier. *Interval Methods for Systems of Equations.* Cambridge University Press, 1990.

[41] J. P. Propp and D. B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. In *Proceedings of the Seventh International Conference on Random Structures and Algorithms (Atlanta, GA, 1995)*, volume 9, pages 223–252, 1996.

[42] R. Sainudiin. *Machine interval experiments.* PhD thesis, Cornell University, Ithaca, New York, 2005.

[43] R. Sainudiin and J. Harlow. A C++ class library for statistical set-processing. In R. Bhatia, editor, *Short Communication in Mathematical Software, International Congress of Mathematicians, Hyderabad, August 19-27*, page 670. Hindustan Book Agency, 2010.

[44] R. Sainudiin and T. York. An auto-validating rejection sampler. BSCB Dept. Technical Report BU-1661-M, Cornell University, Ithaca, New York, 2005.

[45] R. Sainudiin and T. York. Auto-validating von Neumann rejection sampling from small phylogenetic tree spaces. *Algorithms for Molecular Biology*, 4:1, 2009.

[46] R. Sainudiin and R. Yoshida. Applications of interval methods to phylogenetic trees. In L. Pachter and B. Sturmfels, editors, *Algebraic statistics for computational biology*, pages 359–374. Cambridge University Press, 2005.

[47] H. Schichl and A. Neumaier. Interval analysis on directed acyclic graphs for global optimization. *Journal of Global Optimization*, 33(4):541–562, 2005.

[48] C. Semple and M. Steel. *Phylogenetics*. Oxford University Press, 2003.

[49] A. N. Shiryaev. *Probability*. Springer-Verlag, 1989.

[50] W. Tucker. A rigorous ODE solver and Smale's 14th problem. *Foundations of Computational Mathematics*, 2(1):53–117, 2002.

[51] J. von Neumann. Various techniques used in connection with random digits. In *John Von Neumann, Collected Works*, volume V. Oxford University Press, 1963.

[52] A. J. Walker. An efficient method for generating discrete random variables with general distributions. *ACM Trans. Math. Softw.*, 3:253–256, 1977.

[53] Z. Yang. *Computational Molecular Evolution*. Oxford University Press, UK, 2006.