2 | APPROXIMATE INFERENCE

In the previous chapter we argued that the key computational challenge in performing inference in probabilistic models is being able to evaluate integrals with respect to probability distributions defined on high-dimensional spaces. Generally these integrals will not have analytic solutions and for models with even moderate numbers of unobserved variables, numerical quadrature approaches to evaluating integrals are computationally infeasible.

Truth is much too complicated to allow anything but approximations.
—John von Neumann

In this chapter we will review some of the key algorithms proposed for computing *approximate* solutions to inference problems. A unifying aspect to all of these methods is trading off some loss of the accuracy of the answers provided to inferential queries, for a potentially significant increase in computational tractability. The literature on *approximate inference* methods is vast and so necessarily this chapter will only be a partial review of the methods directly relevant to this thesis.

Approximate inference methods can be roughly divided into two groups: methods in which integrals with respect to the target distribution are estimated by averaging over points sampled from a distribution over the target space and those in which a more tractable approximation to the target distribution is found by optimising the approximation to be 'close' to the target distribution. In this chapter we will concentrate on the sampling-based approaches to approximate inference.

In particular we will focus on *Markov chain Monte Carlo* (MCMC) methods, as these form the key basis for the contributions discussed in later chapters. We will review the key theory underlying Monte Carlo integration and MCMC methods and some of the standard algorithms for implementing these approaches. We will conclude with a discussion of auxiliary variable MCMC methods which are central to the methods discussed in the remainder of this thesis.

Although they are not the main focus of this thesis we will make use of several optimisation-based approximate inference methods within the algorithms discussed in the following chapters. We review the ideas underlying these methods in Appendix C.

2.1 MONTE CARLO METHODS

Inference at both the level of computing conditional expectations of unobserved variables in a model and in evaluating evidence terms to allow model comparison involves integrating functions against a probability distribution. Typically the distribution of interest will be defined by a probability density with respect to a base measure. Therefore we wish to be able to compute integrals of the form

$$\int_X f(\boldsymbol{x})\,\mathrm{d}P(\boldsymbol{x}) = \int_X f(\boldsymbol{x})\,p(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x}) \tag{2.1}$$

where p is the density of a target distribution P on a space X with respect to a base measure μ and f is a measurable function. For instance in the case of computing the *posterior mean* in a Bayesian inference problem with observed variables $\mathbf{y}$ and latent variables $\mathbf{x}$ where the posterior density $\mathsf{p}_{\mathbf{x}|\mathbf{y}}$ is defined with respect to the D-dimensional Lebesgue measure, we would have $p(\boldsymbol{x}) = \mathsf{p}_{\mathbf{x}|\mathbf{y}}(\boldsymbol{x} \mid \boldsymbol{y})$ for an observed $\boldsymbol{y}$, $\mu(\boldsymbol{x}) = \lambda^D(\boldsymbol{x})$ and $f(\boldsymbol{x}) = \boldsymbol{x}$. Often we will only be able to evaluate p up to an unknown unnormalising constant i.e. $p(\boldsymbol{x}) = \frac{1}{Z}\tilde{p}(\boldsymbol{x})$ with we able to evaluate $\tilde{p}$ pointwise but Z intractable to compute. For example in a Bayesian inference setting $\tilde{p}(\boldsymbol{x})$ would be the joint density $\mathsf{p}_{\mathbf{x},\mathbf{y}}(\boldsymbol{x}, \boldsymbol{y})$ and Z the model evidence $\mathsf{p}_{\mathbf{y}}(\boldsymbol{y})$. When peforming inference in undirected models, we would instead have that $\tilde{p}$ is the product of unnormalised factors and Z the corresponding normaliser.

2.1.1 Monte Carlo integration

The eponym of the Monte Carlo method is a Monocan casino, favoured haunt of the uncle of Stanisław Ulam, one of the method's inventors.

The framework that unifies all of the methods we will discuss in this section is the *Monte Carlo* method for integration [82]. Let $\mathbf{x}$ be a random vector distributed according to the target distribution i.e. $\mathsf{P}_{\mathbf{x}} = P$. Given an arbitrary measurable function $f : X \to \mathbb{R}$ we define a random variable $\mathsf{f} = f(\mathbf{x})$. Our task is to compute expectations $\mathbb{E}[\mathsf{f}] = \bar{f}$ corresponding to the integral (2.1). We assume that $\mathbb{E}[\mathsf{f}]$ exists and both $\mathbb{E}[\mathsf{f}]$ and $\mathbb{V}[\mathsf{f}]$ are finite. For now we assume we have a way of generating values of N random variables $\{\mathbf{x}_n\}_{n=1}^N$, each marginally distributed according to the target distribution i.e. $\mathsf{P}_{\mathbf{x}_n} = P \;\forall n \in 1 \dots N$ but potentially not independent of each other. We define random variables

$$\mathsf{f}_n = f(\mathbf{x}_n) \quad \forall n \in \{1 \dots N\} \quad \text{and} \quad \hat{\mathsf{f}}_N = \frac{1}{N}\sum_{n=1}^N \mathsf{f}_n. \tag{2.2}$$

Due to linearity of the expectation operator, we have that

$$\mathbb{E}\left[\hat{\mathsf{f}}_N\right] = \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}[\mathsf{f}_n] = \frac{1}{N}\sum_{n=1}^{N}\bar{f} = \bar{f} \tag{2.3}$$

and so that in expectation $\hat{\mathsf{f}}_N$ is equal to $\bar{f}$, i.e. realisations of $\hat{\mathsf{f}}_N$ are *unbiased estimators* of $\bar{f}$. Note that this result does not require any independence assumptions about the generated random variables. Now considering the variance of $\hat{\mathsf{f}}_N$ we can show that

$$\mathbb{V}\left[\hat{\mathsf{f}}_N\right] = \frac{\mathbb{V}[\mathsf{f}]}{N}\left(1 + \frac{2}{N}\sum_{n=1}^{N-1}\sum_{m=1}^{n-1}\frac{\mathbb{C}[\mathsf{f}_n,\mathsf{f}_m]}{\mathbb{V}[\mathsf{f}]}\right). \tag{2.4}$$

If the generated random variables $\{\mathbf{x}_n\}_{n=1}^{N}$ and so $\{\mathsf{f}_n\}_{n=1}^{N}$ are independent, then $\mathbb{C}[\mathsf{f}_n,\mathsf{f}_m] = 0\ \forall m \neq n$. In this case (2.4) reduces to $\mathbb{V}\left[\hat{\mathsf{f}}_N\right] = \mathbb{V}[\mathsf{f}]/N$, i.e. the variance of the *Monte Carlo estimate* $\hat{\mathsf{f}}_N$ for $\bar{f}$ is inversely proportional to the number of samples N. Importantly this scaling does not depend on the dimension of $\mathbf{x}$.

Therefore if we can generate a set of independent random variables from the target distribution, we can estimate expectations that asymptotically tend to the true value as N increases, with a typical deviation from the true value (as measured by the standard deviation, i.e. the square root of variance) that is $\mathcal{O}\left(N^{-\frac{1}{2}}\right)$. In comparison a fourth-order quadrature method such as *Simpson's rule* has an error that is $\mathcal{O}\left(N^{-\frac{4}{D}}\right)$ for a grid of N points uniformly spaced across a D dimensional space. Asymptotically for $D > 8$, Monte Carlo integration will therefore give better convergence than Simpson's rule, and even for smaller dimensions large constant factors in the Simpson's rule dependence can mean Monte Carlo performs better for practical N.

Note that computing Monte Carlo estimates from independent random variables is not optimal in terms of minimising $\mathbb{V}\left[\hat{\mathsf{f}}_N\right]$ for a given f; the covariance terms in (2.4) can be negative which can reduce the overall variance. A wide range of *variance reduction* methods have been proposed to exploit this and produce lower variance of Monte Carlo estimates for a given f [46]. Although these methods can be important in practice for achieving an estimator with a practical variance for a specific f of interest, we will generally concentrate on the case where we do not necessarily know f in advance.

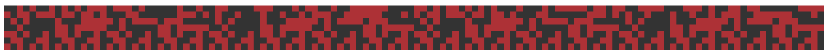


Figure 2.1: Binary representation of linear congruential generator sequence $s_{n+1} = 37s_n + 61 \mod 128$. Columns left to right represents successive integer states in sequence. From least significant (bottom) to most significant (top), the bits in each column have patterns repeating with periods 2, 4, 8, 16, 32, 64, 128.

2.1.2 Pseudo-random number generation

The generation of random numbers is too important to be left to chance.
—Robert R. Coveyou

Virtually all statistical computations involving random numbers in practice make use of *pseudo-random number generators* (PRNGs). Rather than generating samples via a truly random process[1], PRNGs produce deterministic sequences of integers in a fixed range that nonetheless maintain many of the properties of a random sequence. In particular through careful choice of the updates, sequences with a very long period (number of iterations before the sequence begins repeating), a uniform distribution across the numbers in the sequence range and low correlation between successive states can be constructed.

A very simple example of a class of PRNGs is the *linear congruential generator* [47] which obeys the recurrent update

$$s_{n+1} = (as_n + c) \mod m \quad \text{with} \quad 0 < a < m,\ 0 \le c < m, \tag{2.5}$$

with a, c and m integer parameters. If a, c and m are chosen appropriately, iterating the update (2.5) from an initial seed $0 \le s_0 < m$, will produce a sequence of states which visits all the integers in $[0, m)$ before repeating. An example state sequence with $m = 128$ is shown in Figure 2.1. In practice, linear congruential generators produce sequences with poor statistical properties, particularly when used to generate random points in high dimensional spaces [51], hence most modern numerical computing libraries use more robust PRNGs such as the *Mersenne-Twister* [52], which is used in all experiments in this thesis.

The raw output of a PRNG is an integer sequence, with typically the sequence elements uniformly distributed over all integers in a range $[0, 2^n)$ for some $n \in \mathbb{N}$. All real values are represented at a finite precision on computers, typically using a floating point representation [4] of *single* (24-bit mantissa) or *double* (53-bit mantissa) precision. Through

1 We consider a true random process as one in which it is impossible to precisely predict the next value in the sequence given the previous values.

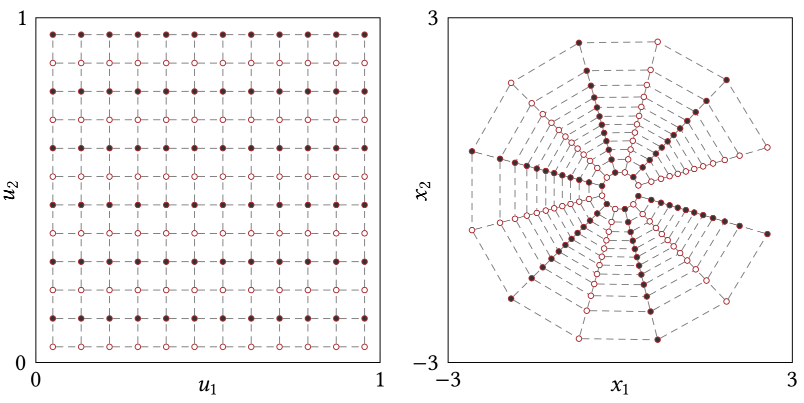


Figure 2.2: Visualisation of Box–Muller transform. Left axis shows uniform grid on $U = [0,1]^2$ and right-axis shows grid points after mapping through $\boldsymbol{g}$ in transformed space $X = \mathbb{R}^2$.

an approriate linear transformation, the integer outputs of a PRNG can be converted to floating-point values uniformly distributed across a finite interval. PRNG implementations typically provide a primitive to generate floating-point values uniformly distributed on $[0,1)$. Given the ability to generate sequences of (effectively) independent samples from a uniform distribution $\mathcal{U}(0,1)$, the question is then how to use these values to produce random samples from arbitrary densities.

2.1.3 Transform sampling

Samples from many standard distributions can be generated by exploiting the transformation of random variables relationships discussed in 1.1.4. Let $\mathbf{u}$ be a D-dimensional vector of independent random variables marginally distributed according to $\mathcal{U}(0,1)$ and $\boldsymbol{g} : [0,1)^D \to X$ be a bijective map to a vector space $X \subseteq \mathbb{R}^D$. If we define $\mathbf{x} = \boldsymbol{g}(\mathbf{u})$, then by the vector change of variables formula (1.22) we have that

$$\mathrm{p}_{\mathbf{x}}(\boldsymbol{x}) = \left|\frac{\partial \boldsymbol{g}^{-1}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|. \tag{2.6}$$

For example for $D = 2$, $X = \mathbb{R}^2$ and a bijective map $\boldsymbol{g}$ defined by

$$\boldsymbol{g}\begin{pmatrix}u_1\\u_2\end{pmatrix} = \begin{bmatrix}\sqrt{-2\log u_1}\cos(2\pi u_1)\\\sqrt{-2\log u_1}\sin(2\pi u_2)\end{bmatrix},\ \boldsymbol{g}^{-1}\begin{pmatrix}x_1\\x_2\end{pmatrix} = \begin{bmatrix}\exp\left(-\frac{1}{2}(x_1^2+x_2^2)\right)\\\frac{1}{2\pi}\arctan\left(\frac{x_1}{x_2}\right)\end{bmatrix},$$

then we have that the density of the transformed $\mathbf{x} = \boldsymbol{g}(\mathbf{u})$ is

$$\mathrm{p}_{\mathbf{x}}(\boldsymbol{x}) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x_1^2}{2}\right)\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x_2^2}{2}\right), \tag{2.7}$$

i.e. x_1 and x_2 are independent random variables with standard normal distributions $\mathcal{N}(0,1)$. This is the *Box–Muller transform* [16], and allows generation of independent standard normal variables given a PRNG primitive for sampling from $\mathcal{U}(0,1)$. A visualisation of the transformation of space applied by the method is shown in Figure 2.2.

A general method for sampling from univariate distributions is to use an inverse *cumulative distribution function* (CDF) transform. For a probability density p on a scalar random variable, the corresponding CDF $r : \mathbb{R} \to [0,1]$ is defined as

$$r(x) = \int_{-\infty}^{x} p(v)\,\mathrm{d}v \implies \frac{\partial r(x)}{\partial x} = p(x). \tag{2.8}$$

If u is a standard uniform random variable and $\mathrm{x} = r^{-1}(\mathrm{u})$ then

$$\mathrm{p}_{\mathrm{x}}(x) = \left|\frac{\partial r(x)}{\partial x}\right| = p(x). \tag{2.9}$$

To be able to use the inverse CDF transform method we need to be able to evaluate r^{-1}, sometimes termed the *quantile function*. Often neither the CDF or quantile function of a univariate distribution will have closed form solutions however we can use polynomial approximation methods and iterative solvers to evaluate both to arbitary precision [64]. For some distributions such as the standard normal $\mathcal{N}(0,1)$ even though the CDF and quantile function do not have analytic forms in terms of elementary functions it is common for numerical computing libraries to provide numerical approximations to both which are accurate to within small multiples of machine precision. Although the inverse CDF transform method gives a general recipe for sampling from univariate densities, it is not easy to generalise to multivariate densities and alternatives can be simpler to implement and more numerically stable.

2.1.4 Rejection sampling

An important class of generic sampling methods, particularly due their use as a building block in other algorithms, is rejection sampling [61]. Rejection sampling uses the observation that to sample from a distribution with density $p : X \to [0,\infty)$ it is sufficient to uniformly sample from the volume under the graph of $(\boldsymbol{x}, p(\boldsymbol{x}))$.

The key requirement in rejection sampling is to identify a *proposal distribution* Q which we can generate independent samples from and has

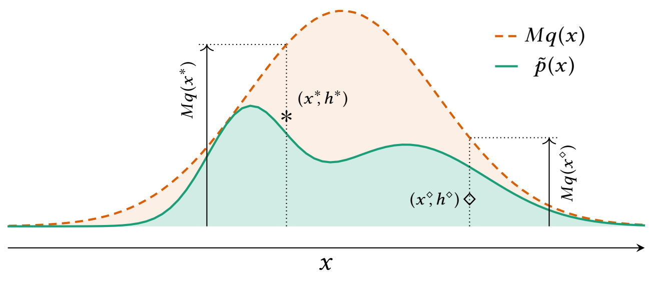


Figure 2.3: Visualisation of rejection sampling. The green curve shows the (unnormalised) target density $\tilde{p}$, the green region underneath representing the area we wish to sample points uniformly from. The dashed orange curve shows the scaled proposal density Mq, with the orange (plus green) region representing the area we uniformly propose values from. Two example proposals are shown: $\diamond$ is under the target density and so accepted; $*$ is outside of the green region and so would be rejected.

Algorithm 1 Rejection sampling.

Input: $\tilde{p}$: unnormalised target density, q : normalised density of proposal distribution Q, M : constant such that $\tilde{p}(\boldsymbol{x}) \leq Mq(\boldsymbol{x})\ \forall \boldsymbol{x} \in X$.
Output: Independent sample from distribution with density $p(\boldsymbol{x}) = \frac{1}{Z}\tilde{p}(\boldsymbol{x})$.

```
1: repeat
2:     $\boldsymbol{x} \sim q(\cdot)$
3:     $h \sim \mathcal{U}(\cdot \mid 0, Mq(\boldsymbol{x}))$
4: until $h \leq \tilde{p}(\boldsymbol{x})$
5: return $\boldsymbol{x}$
```

a density $q = \frac{\partial Q}{\partial \mu}$ that upper bounds the potentially unnormalised target density $\tilde{p}$ across its full support X when multiplied by a known constant M, i.e. $\tilde{p}(x) \leq Mq(x)\ \forall x \in X$. The requirement to be able to generate independent samples from Q can be met for example by distributions amenable to transform sampling, e.g. the standard normal. The second requirement is generally more challenging and as we will see the efficiency of rejection sampling methods is very dependent on how tight the bound can be made.

Algorithm 1 describes the rejection sampling method to produce a single independent sample from a target distribution. A visualisation of how the algorithm works for a univariate target distribution is shown in Figure 2.3. The overall aim is to generate points uniformly distributed across the green area under the (unnormalised) target density curve. This is achieved by generating points uniformly under the dashed orange curve corresponding to the scaled proposal density and then accepting only those which are below the green curve. To generate a point

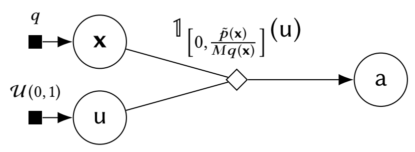

q
$\mathbf{x}$
$\mathbb{1}_{\left[0, \frac{\tilde{p}(\mathbf{x})}{Mq(\mathbf{x})}\right]}(\mathrm{u})$
$\mathcal{U}(0,1)$
u
a

Figure 2.4: Factor graph of rejection sampling process.

under the dashed orange curve we first generate an x from the proposal distribution and then generate an auxilliary 'height' variable by sampling uniformly from $[0, Mq(x)]$. If the sampled height is below the green curve we accept (as in the $\diamond$ example in Figure 2.3) else we reject the sample (as in the $*$ example in Figure 2.3).

Figure 2.4 shows the rejection sampling generative process as a directed factor graph, with $\mathbf{x}$ be a random variable representing the proposal, u the uniform auxiliary variable drawn to sample the 'height' and a a binary variable that indicates whether the proposal is accepted (a = 1) or not (a = 0). By marginalising out u we have that that

$$\mathrm{p}_{\mathbf{x},\mathrm{a}}(\boldsymbol{x}, a) = q(\boldsymbol{x})\left(\frac{\tilde{p}(\boldsymbol{x})}{Mq(\boldsymbol{x})}\right)^{a}\left(1 - \frac{\tilde{p}(\boldsymbol{x})}{Mq(\boldsymbol{x})}\right)^{1-a}, \tag{2.10}$$

and further marginalising over the proposal $\mathbf{x}$

$$\mathrm{p}_{\mathrm{a}}(a) = \left(\frac{Z}{M}\right)^{a}\left(1 - \frac{Z}{M}\right)^{1-a}. \tag{2.11}$$

Conditioning on the proposal being accepted we therefore have that

$$\mathrm{p}_{\mathbf{x}|\mathrm{a}}(\boldsymbol{x} \mid 1) = \frac{q(\boldsymbol{x})\frac{\tilde{p}(\boldsymbol{x})}{Mq(\boldsymbol{x})}}{\frac{Z}{M}} = \frac{\tilde{p}(\boldsymbol{x})}{Z} = p(\boldsymbol{x}). \tag{2.12}$$

Therefore the accepted proposals are distributed according to the target density as required. Further from (2.11) we have that the $\mathrm{p}_{\mathrm{a}}(1) = \frac{Z}{M}$. This suggests we can use the accept rate to estimate Z but also hints at the difficulty in finding a M which guarantees the upper bound requirement as for $\frac{Z}{M}$ to be a valid probability $M \geq Z$ i.e. M needs to be an upper bound on the unknown normalising constant Z. This relationship also suggests it is desirable to set M as small as possible to maximise the acceptance rate.

Although rejection sampling can be an efficient method of sampling from univariate target distributions (particularly for distributions with log-concave densities where adaptive variants are available [33]), it generally scales very poorly with the dimensionality of the target distribution. This is as the ratio of the volume under the target density to the volume under the scaled proposal density (in terms of Figure 2.3 the ratio of the green area to the green plus orange regions), and so the probability of accepting a proposal, will tend become exponentially smaller as the dimensionality increases. This is an example of the so-called *curse of dimensionality*. Therefore although rejection sampling can be a useful subroutine for generating random variables from low-dimensional distributions, in general it is not a viable option for generating samples directly for high-dimensional Monte Carlo integration.

2.1.5 Importance sampling

So far we have considered methods for generating samples directly from a target distribution. Although samples can be of value in themselves for giving a representative set of plausible values from the target distribution (e.g. for visualisation purposes), usually the end goal is to estimate integrals of the form in (2.1).

Importance sampling [44] is a Monte Carlo method which allows arbitrary integrals to be estimated. If Q is a distribution, with density $q = \frac{\partial Q}{\partial \mu}$, which is absolutely continuous with respect to the target distribution (which requires that $p(\boldsymbol{x}) > 0 \Rightarrow q(\boldsymbol{x}) > 0$), then importance sampling is based on the identity

$$\bar{f} = \frac{\int_X f(\boldsymbol{x})\,\tilde{p}(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x})}{\int_X \tilde{p}(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x})} = \frac{\int_X f(\boldsymbol{x})\,\frac{\tilde{p}(\boldsymbol{x})}{q(\boldsymbol{x})}\,q(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x})}{\int_X \frac{\tilde{p}(\boldsymbol{x})}{q(\boldsymbol{x})}\,q(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x})}. \tag{2.13}$$

Each of the numerator and denominator in (2.13) take the form of an expectation of a measurable function of a random variable $\mathbf{x}$ with distribution Q. Further the denominator is exactly equal to $Z = \int_X \tilde{p}(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x})$. We therefore have that

$$Z\bar{f} = \mathbb{E}[w(\mathbf{x})f(\mathbf{x})] \;\text{ and }\; Z = \mathbb{E}[w(\mathbf{x})] \;\text{ with }\; w(\boldsymbol{x}) = \frac{\tilde{p}(\boldsymbol{x})}{q(\boldsymbol{x})}. \tag{2.14}$$

If we can generate random variables $\{\mathbf{x}_n\}_{n=1}^N$ each marginally distributed according to Q we can therefore form Monte Carlo estimates of both the numerator and denominator. We define $\hat{\mathrm{Z}}_N$ and $\hat{\mathrm{g}}_N$ as

$$\hat{\mathrm{Z}}_N = \frac{1}{N}\sum_{n=1}^{N} w(\mathbf{x}_n) \;\text{ and }\; \hat{\mathrm{g}}_N = \frac{1}{\hat{\mathrm{Z}}}\sum_{n=1}^{N} w(\mathbf{x}_n) f(\mathbf{x}_n). \tag{2.15}$$

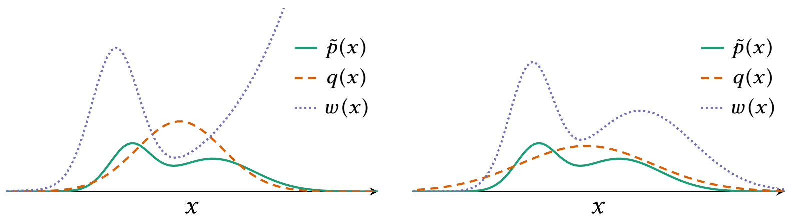


Figure 2.5: Visualisation of importance sampling. On both axes the green curve shows the unnormalised target density $\tilde{p}$, the dashed orange curve the density q values are sampled from and the dotted violet curve the importance weighting function $w(x) = \frac{\tilde{p}(x)}{q(x)}$ to estimate expectations with respect to the target density using samples from q. In the left axis the q chosen is underdispersed compared to $\tilde{p}$ leading to very large w values in the right tail. In constrast in the right axis, the broader q leads to less extreme variation in w.

By the same argument as Section 2.1.1, $\mathbb{E}\left[\hat{\mathrm{Z}}_N\right] = Z$ and $\mathbb{E}[\hat{\mathrm{g}}_N] = Z\bar{f}$. We can therefore use importance sampling to form an unbiased estimate of the unknown normalising constant Z.

If we define $\hat{\mathrm{f}}_N = \hat{\mathrm{g}}_N / \hat{\mathrm{Z}}_N$, then this is a biased[2] estimator for $\bar{f}$ as in general the expectation of the ratio of two random variables is not equal to the ratios of their expectations. However if both the numerator and denominator have finite variance, i.e. $\mathbb{V}\left[\hat{\mathrm{Z}}_N\right] < \infty$ and $\mathbb{V}[\hat{\mathrm{g}}_N] < \infty$, then $\hat{\mathrm{f}}_N$ is a *consistent* estimator for $\bar{f}$ i.e. $\lim_{N\to\infty} \mathbb{E}\left[\hat{\mathrm{f}}_N\right] = \bar{f}$.

The $w(\mathrm{x}_n)$ values are typically termed the *importance weights*. If a few of the weights are very large, the weighted sums in (2.15) will be dominated by those few values, reducing the effective number of samples in the Monte Carlo estimates. This can particularly be a problem if the are regions of X with low probability under q where $p(\boldsymbol{x}) \gg q(\boldsymbol{x})$. As sampling points in these regions will be a rare event, a large number of samples may be needed to diagnose the issue adding further difficulty. A general recommendation is to use densities q with tails as least as heavy of those of p, and in general the closer the match between q and p the better [49, 65]. Figure 2.5 shows a visualisation of the effect of the choice of q on the importance weights.

When previously discussing rejection sampling, we introduced an auxiliary binary accept indicator variable, a, associated with each proposed

2 In cases where the normalising constant Z is known, we can instead use $w(\boldsymbol{x}) = \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}$ in which case the ratio estimator is not required and an unbiased estimates can be calculated. As the problems we are interested in will generally have unknown Z we do not consider this case further

sample $\mathbf{x}$ (see Figure 2.4). If we generate N independent proposal – indicator pairs $\{\mathbf{x}_n, \mathsf{a}_n\}_{n=1}^N$ then the number of accepted proposals is $\mathsf{N}_{\text{acc}} = \sum_{n=1}^N \mathsf{a}_n$. Conditioned on N_{acc} being a value more than one, the generated rejection sampling variables $\{\mathbf{x}_n, \mathsf{a}_n\}_{n=1}^N$ can be used to form an *unbiased* Monte Carlo estimate of $\bar{f}$ using the estimator

$$\hat{\mathsf{f}}_N^{\text{RS}} = \frac{\sum_{n=1}^N \mathsf{a}_n f(\mathbf{x}_n)}{\sum_{m=1}^N \mathsf{a}_m}, \tag{2.16}$$

which just correponds to computing the empirical mean of the accepted proposals i.e. the standard Monte Carlo estimator. In comparison importance sampling forms a biased but consistent estimator for $\bar{f}$ from N samples $\{\mathbf{x}_n\}_{n=1}^N$ from a distribution Q using the estimator

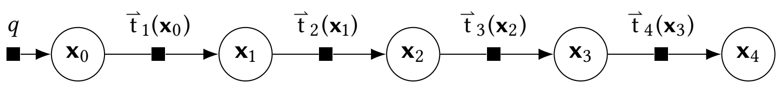
$$\hat{\mathsf{f}}_N^{\text{IS}} = \frac{\sum_{n=1}^N w(\mathbf{x}_n) f(\mathbf{x}_n)}{\sum_{m=1}^N w(\mathbf{x}_m)}. \tag{2.17}$$

From this perspective the accept indicators a_n in rejection sampling can be seen to act like binary importance weights, in contrast importance sampling using 'soft' weights which mean all sampled $\mathbf{x}_n$ make a contribution to the estimator (assuming $w(\boldsymbol{x}) \neq 0 \ \forall \boldsymbol{x} \in X$). However this correspondence is only loose. The rejection sampling estimator $\hat{\mathsf{f}}_N^{\text{RS}}$ is unbiased unlike $\hat{\mathsf{f}}_N^{\text{IS}}$, but this unbiasedness relies on conditioning on a non-zero value for N_{acc} (i.e. the number of accepted samples to generate) and continuing to propose points until this condition is met. In contrast importance sampling generates a fixed number of samples from Q and does not use any auxiliary random variables.

Unlike rejection sampling, there is no need in importance sampling for q to upper-bound the target density (when multiplied by a constant). This allows more freedom in the choice of q however it is still important to choose q to be as close as possible to the target while remaining tractable to generate samples from. In general for target densities defined on high-dimensional spaces, it can be difficult to find an appropriate q such that the variation in importance weights is not too extreme [49].

2.2 MARKOV CHAIN MONTE CARLO

The sampling methods considered in the previous section used independent random variables to form Monte Carlo estimates. When in-



q
$\vec{t}_1(\mathbf{x}_0)$
$\vec{t}_2(\mathbf{x}_1)$
$\vec{t}_3(\mathbf{x}_2)$
$\vec{t}_4(\mathbf{x}_3)$
$\mathbf{x}_0$
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_3$
$\mathbf{x}_4$

Figure 2.6: Markov chain factor graph. The initial state $\mathbf{x}_0$ is sampled according to a density q and each subsequent state $\mathbf{x}_n$ is then generated from a transition density $\vec{t}_n$ conditioned on the previous state $\mathbf{x}_{n-1}$.

troducing the Monte Carlo method we saw that is was not necessary for the random variables used in a Monte Carlo estimator to be independent. While it can be impractically computationally expensive to generate independent samples from complex high-dimensional target distributions, simulating a stochastic process which converges in distribution to the target and produces a sequence of *dependent* random variables is often a more tractable task. This is the idea exploited by MCMC methods.

A *Markov chain* is an ordered sequence of random variables $\{\mathbf{x}_n\}_{n=0}^{N}$ which have the *Markov property* — for all $n \in \{1 \dots N\}$, $\mathbf{x}_n$ is conditionally independent of $\{\mathbf{x}_n\}_{m<n-1}$ given $\mathbf{x}_{n-1}$. This conditional independence structure is visualised as a factor graph in Figure 2.6.

For a Markov chain defined on a general measurable state space $(X, \mathcal{F})$, the probability distribution of a state $\mathbf{x}_n$ given the state $\mathbf{x}_{n-1}$ is specified for each $n \in \{1 \dots N\}$ by a *transition operator*, $\vec{\mathsf{T}}_n : \mathcal{F} \times X \to [0,1]$. In particular the transition operators define a series of regular conditional distributions for each $n \in \{1 \dots N\}$

$$\mathsf{P}_{\mathbf{x}_n | \mathbf{x}_{n-1}}(A \,|\, \boldsymbol{x}) = \vec{\mathsf{T}}_n(A \,|\, \boldsymbol{x}) \quad \forall A \in \mathcal{F},\ \boldsymbol{x} \in X. \tag{2.18}$$

We will often assume that the chain is *homogeneous*, i.e. that the same transition operator is used for all steps $\vec{\mathsf{T}}_n = \vec{\mathsf{T}}\ \forall n \in \{1 \dots N\}$.

The key property required of a transition operator for use in MCMC methods is that the target distribution P is *invariant* under the transition, that is it satisfies

$$P(A) = \int_X \vec{\mathsf{T}}(A \,|\, \boldsymbol{x})\, \mathrm{d}P(\boldsymbol{x}) \quad \forall A \in \mathcal{F}, \tag{2.19}$$

The invariance property means that if a chain state $\mathbf{x}_n$ is distributed according to the target P, all subsequent chain states $\mathbf{x}_{n+1}$, $\mathbf{x}_{n+2} \dots$ will also be marginally distributed according to the target. Therefore given a single random sample $\mathbf{x}_0$ from the target distribution, a series of dependent states marginally distributed according to the target could be generated and used to form Monte Carlo estimates of expectations.

Being able to generate even one exact sample from a complex high-dimensional target distribution is generally infeasible. Importantly however the marginal distribution on the chain state $\mathsf{P}_{\mathbf{x}_n}$ of a Markov chain with a transition operator which leaves the target distribution invariant will converge to the target distribution irrespective of the distribution of the intial chain state if the target distribution is the *unique* invariant distribution of the chain.

To have a unique invariant distribution, a chain must be *irreducible* and *aperiodic* [80]. For a chain on a measurable space $(X, \mathcal{F})$, irreducibility is defined with respect to a measure ν, which could but does not necessarily need to be the target distribution P. A chain is ν-irreducible if starting at any point in X there is a non-zero probability of moving to any set with positive ν-measure in a finite number of steps, i.e.

$$\forall \boldsymbol{x} \in X,\ A \in \mathcal{F} : \nu(A) > 0\ \ \exists\, m \in \mathbb{Z}^+ : \mathsf{P}_{\mathbf{x}_m | \mathbf{x}_0}(A \,|\, \boldsymbol{x}) > 0. \qquad (2.20)$$

A chain is periodic (and aperiodic otherwise) if disjoint regions of X are visited cyclically, i.e. there exists an integer $r > 1$ and an ordered set of r disjoint P-positive subsets of X, $\{A_i\}_{i=1}^{r}$ such that $\vec{\mathsf{T}}(A_j \,|\, \boldsymbol{x}) = 1\ \forall \boldsymbol{x} \in A_i,\ i \in \{1 \dots r\},\ j = (i+1) \mod r$.

If we can construct a ν-irreducible and aperiodic Markov chain $\{\mathbf{x}_n\}_{n=0}^{N}$ which has the target distribution P as its invariant distribution, then a MCMC estimator $\hat{\mathrm{f}}_N = \frac{1}{N}\sum_{n=1}^{N} f(\mathbf{x}_n)$ converges almost surely as $N \to \infty$ to $\bar{f} = \int_X f \,\mathrm{d}P$ for all starting states except for a ν-null set[3][55]. This convergence of *time-averages* (i.e. over states at different steps of the Markov chain) to *space-averages* (i.e. with respect to the stationary distribution across the state space), is termed *ergodicity* and is a consequence of the *Birkhoff–Khinchin ergodic theorem* [15].

Although irreducibility and aperiodicity of a Markov chain which leaves the target distribution invariant are sufficient for convergence of MCMC estimators, this does not tell us anything about the rate of that convergence and so how to quantify the error introduced by computing estimates with a Markov chain simulated for only a finite number of

3 The 'except for a ν-null set' caveat can be removed by requiring the stronger property of *Harris recurrence* [38].

steps. Stronger notions of ergodicity can be used to help quantify convergence; we will concentrate on *geometric ergodicity* here. We first define a notion of distance between two measures μ and ν on a measurable space $(X, \mathcal{F})$, the *total variation distance*, as

$$\|\mu - \nu\|_{\mathrm{TV}} = \sup_{A \in \mathcal{F}} |\mu(A) - \nu(A)|. \tag{2.21}$$

For a ν-irreducible and aperiodic chain with invariant distribution P our earlier statement that the distribution on the chain state converges to P can now be restated more precisely as that for ν-almost all initial states $\mathbf{x}_0 = \boldsymbol{x}$, $\lim_{n\to\infty} \|\mathsf{P}_{\mathbf{x}_n \mid \mathbf{x}_0}(\cdot \mid \boldsymbol{x}) - P\|_{\mathrm{TV}} = 0$. Geometric ergodicity makes a stronger statement that the convergence in total variation distance conditioned is geometric in n, i.e. that

$$\|\mathsf{P}_{\mathbf{x}_n \mid \mathbf{x}_0}(\cdot \mid \boldsymbol{x}) - P\|_{\mathrm{TV}} \leq m(\boldsymbol{x}) r^n \tag{2.22}$$

for a positive measurable function m which depends on the initial chain state $\boldsymbol{x}$ and rate constant $r \in [0, 1)$. For chains which are geometrically ergodic, we can derive an expression for the *asymptoptic variance* of an MCMC estimator $\hat{\mathsf{f}}_N$ related to the variance of a simple Monte Carlo estimator previously considered in Section 2.1.1.

A stochastic process is stationary if the joint distribution of the states at any set of time points does not change if all those times are shifted by a constant.

As in Section 2.1.1 we define $\mathsf{f}_n = f(\mathbf{x}_n)$ and $\hat{\mathsf{f}}_N = \frac{1}{N}\sum_{n=1}^{N} \mathsf{f}_n$, with here the $\{\mathbf{x}_n\}_{n=1}^{N}$ the states of a Markov chain. For a homogeneous Markov chain with a unique invariant distribution P which is *stationary*, the marginal distribution on the states $\mathsf{P}_{\mathbf{x}_n}$ is equal to P for all n and we can use the expression for the variance of a general Monte Carlo estimator (which did not assume independence of the random variables) stated earlier in (2.4). Further the stationarity of the chain means that the covariance $\mathbb{C}[\mathsf{f}_n, \mathsf{f}_m]$ depends only on the difference $n - m$, and so the variance of the estimator simplifies to

$$\mathbb{V}\left[\hat{\mathsf{f}}_N\right] = \frac{\mathbb{V}[\mathsf{f}]}{N}\left(1 + 2\sum_{n=1}^{N-1}\left(\frac{N-n}{N}\frac{\mathbb{C}[\mathsf{f}_0, \mathsf{f}_n]}{\mathbb{V}[\mathsf{f}]}\right)\right). \tag{2.23}$$

If we multiply both sides of (2.23) by N and define $\rho_n = \frac{\mathbb{C}[\mathsf{f}_0, \mathsf{f}_n]}{\mathbb{V}[\mathsf{f}]}$ (the lag n autocorrelations of $\{\mathsf{f}_n\}$), under the assumption that $\sum_{n=1}^{\infty} |\rho_n| < \infty$ in the limit of $N \to \infty$ we have that

$$\lim_{N\to\infty}\left(N\,\mathbb{V}\left[\hat{\mathsf{f}}_N\right]\right) = \mathbb{V}[\mathsf{f}]\left(1 + 2\sum_{n=1}^{\infty}\rho_n\right). \tag{2.24}$$

Now considering a chain which is geometrically ergodic from its initial state, if $\mathbb{E}\left[|\mathsf{f}|^{2+\delta}\right]$ is finite for some $\delta > 0$ then it can be shown [22, 30, 71] that (2.24) is also the asymptoptic variance for a MCMC estimator calculated using the chain states.

This motivates a definition of the *effective sample size* (ESS) for an MCMC estimator $\hat{\mathsf{f}}_N$ computed using a geometrically ergodic chain as

$$N_{\text{eff}} = \frac{N}{1 + 2\sum_{n=1}^{\infty} \rho_n}. \tag{2.25}$$

The ESS quantifies the number of independent samples that would be required in a Monte Carlo estimator to give an equivalent variance to the MCMC estimator $\hat{\mathsf{f}}_N$ in the asymptoptic limit $N \to \infty$. In practice we cannot evaluate the exact autocorrelations and so we can only compute an estimated ESS, $\hat{N}_{\text{eff}}$, from one or more simulated chains with the estimation method needing to be carefully chosen to ensure reasonable values [79]. Although the assumption of geometric ergodicity can often be hard to verify in practice and ESS estimates can give misleading results in chains far from convergence, when used appropriately estimated ESSs can still be a useful heuristic for evaluating and comparing the effiency of Markov chain estimators and are often available as a standard diagnostic in MCMC software packages [19, 68, 75].

So far we have not discussed how to construct a transition operator giving a chain with the required invariant distribution. As a notational convenience we will consider the transition operator as being specified by a conditional density we term the *transition density* $\vec{\mathsf{t}} : X \times X \to [0, \infty)$ which is defined with respect to a base measure μ (which we assume to be the same as that which the target density we wish to integrate against is defined with respect to, hence the reuse of notation). The transition operator is then

$$\vec{\mathsf{T}}(A \,|\, \boldsymbol{x}) = \int_A \vec{\mathsf{t}}(\boldsymbol{x}' \,|\, \boldsymbol{x}) \,\mathrm{d}\mu(\boldsymbol{x}') \quad \forall A \in \mathcal{F},\ \boldsymbol{x} \in X. \tag{2.26}$$

In practice the probability measure defined by a transition operator will often have a singular component, for example corresponding to a non-zero probability of the chain remaining in the current state. In this case $\vec{\mathsf{T}}$ is not absolutely continuous with respect to μ and a transition density is not strictly well defined. As we did in the previous chapter however we will informally use Dirac deltas to represent a 'density' of singular measures, and so still consider a transition density as existing. The requirement that the transtion operator leaves the target distribution invariant, can then be expressed in terms of the target density p and transition density $\vec{\mathrm{t}}$ as

$$p(\boldsymbol{x}') = \int_X \vec{\mathrm{t}}\,(\boldsymbol{x}' \,|\, \boldsymbol{x})\, p(\boldsymbol{x})\, \mathrm{d}\mu(\boldsymbol{x}) \quad \forall \boldsymbol{x}' \in X. \tag{2.27}$$

Finding a transition density which leaves the target density invariant by satisfying (2.27) seems difficult in general as it involves evaluating an integral against the target density - precisely the computational task which we have been forced to seek approximate solutions to. We can make progress by considering the joint density of a pair of successive states for a chain with invariant distribution P that has converged to stationarity. Then we have that

$$\mathrm{p}_{\mathbf{x}_n, \boldsymbol{x}_{n-1}}(\boldsymbol{x}', \boldsymbol{x}) = \mathrm{p}_{\mathbf{x}_n \,|\, \boldsymbol{x}_{n-1}}(\boldsymbol{x}' \,|\, \boldsymbol{x})\, \mathrm{p}_{\mathbf{x}_{n-1}}(\boldsymbol{x}) = \vec{\mathrm{t}}\,(\boldsymbol{x}' \,|\, \boldsymbol{x})\, p(\boldsymbol{x}). \tag{2.28}$$

We can also consider factorising this joint density into the product of the marginal density of the current state $\mathrm{p}_{\mathbf{x}_n}$ and the conditional density of the previous state given the current state $\mathrm{p}_{\mathbf{x}_{n-1} | \boldsymbol{x}_n}$. Due to stationarity $\mathrm{p}_{\mathbf{x}_n}$ is also equal to p and so we have that $\mathrm{p}_{\mathbf{x}_{n-1} | \boldsymbol{x}_n}$ must be the density of a transition operator which also leaves P invariant, corresponding to a time reversed version of the original (stationary) Markov chain[4]. If we therefore denote $\overleftarrow{\mathrm{t}} = \mathrm{p}_{\mathbf{x}_{n-1} | \boldsymbol{x}_n}$ (and which we will term the *backward transition density* in contrast to $\vec{\mathrm{t}}$ which in this context we will qualify as the *forward transition density*), we have that

$$\vec{\mathrm{t}}\,(\boldsymbol{x}' \,|\, \boldsymbol{x})\, p(\boldsymbol{x}) = \overleftarrow{\mathrm{t}}\,(\boldsymbol{x} \,|\, \boldsymbol{x}')\, p(\boldsymbol{x}') \;\; \forall \boldsymbol{x} \in X,\, \boldsymbol{x}' \in X. \tag{2.29}$$

Integrating both sides with respect to $\boldsymbol{x}$, we have that $\forall \boldsymbol{x}' \in X$

$$\int_X \vec{\mathrm{t}}\,(\boldsymbol{x}' \,|\, \boldsymbol{x})\, p(\boldsymbol{x})\, \mathrm{d}\mu(\boldsymbol{x}) = \int_X \overleftarrow{\mathrm{t}}\,(\boldsymbol{x} \,|\, \boldsymbol{x}')\, \mathrm{d}\mu(\boldsymbol{x})\, p(\boldsymbol{x}') = p(\boldsymbol{x}'), \tag{2.30}$$

and so that (2.27) is satisfied, with the last inequality arising due to $\overleftarrow{\mathrm{t}}$ being a normalised density on its first argument. Therefore if we can find a pair of transition densities, $\vec{\mathrm{t}}$ and $\overleftarrow{\mathrm{t}}$, satisfying (2.29), then the transition operator specified by $\vec{\mathrm{t}}$ will leave the target distribution P

4 The time reversal of a Markov chain is always itself a a Markov chain irrespective of stationarity (as the defining conditional independence structure is symmetric with respect to the direction of time), however the reverse of a homogeneous Markov chain which is not stationary will not in general itself be homogeneous.

invariant (and by an equivalent argument so will the transition operator specified by $\overleftarrow{\mathrm{t}}$). We can further simplify (2.29) by requiring that $\overrightarrow{\mathrm{t}} = \overleftarrow{\mathrm{t}} = \mathrm{t}$, i.e. that both forward and backward transition densities (and corresponding operators) take the same form and so that the chain at stationarity is *reversible*, in which case have that

$$\mathrm{t}(\boldsymbol{x}' \mid \boldsymbol{x})\, p(\boldsymbol{x}) = \mathrm{t}(\boldsymbol{x} \mid \boldsymbol{x}')\, p(\boldsymbol{x}') \quad \forall \boldsymbol{x} \in X,\ \boldsymbol{x}' \in X. \tag{2.31}$$

This is often termed the *detailed balance* condition. Importantly both the detailed balance (2.31) and *generalised balance* (2.29) conditions can also be written in terms of the unnormalised density $\tilde{p}$ by multiplying both sides by Z, and so can be checked even when Z is unknown.

The restriction to reversible transition operators in detailed balance, while sufficient for (2.27) to hold is not necessary. Markov chains which satisfy the generalised balance condition but not detailed balance are termed *non-reversible*, and there are theoretical results suggesting that non-reversible Markov chains can sometimes achieve significantly improved convergence compared to related reversible chains [25, 43, 59]. While there are several general purpose frameworks for specifying reversible transition operators which leave a target distribution invariant, developing methods for constructing irreversible transition operators with a desired invariant distribution has proven more challenging. The approaches proposed to date are generally limited in practice to special cases such as finite state spaces [76, 77, 81] or chains with tractable invariant distributions such as the multivariate normal [14].

Nonetheless non-reversible Markov chains are still commonly used in MCMC applications. Given a set of transition operators which each individually leave a target distribution invariant, the sequential composition of the transition operators will by induction necessarily also leave the target distribution invariant. Even if the individual transition operators are all reversible, the overall sequential composition will generally not be (instead having an adjoint 'backward' operator corresponding to applying the individual transitions in the reversed order). Sequentially combining several reversible transition operators is common in MCMC implementations, though this is more often the result of each individual operator not meaning the requirements for ergodicity in isolation and so needing to be combined with other operators, rather than due to a specific aim of introducing irreversibility.

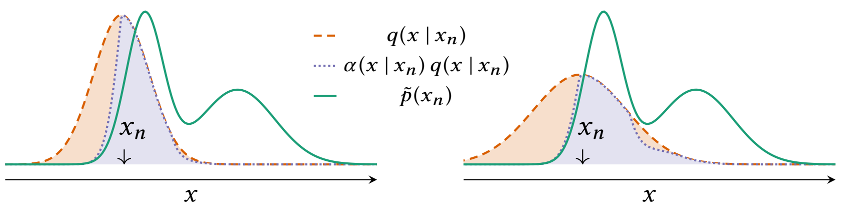


Figure 2.7: Visualisation of Metropolis–Hastings algorithm in a univariate target distribution. The green curves shows the unnormalised target density. The arrows indicate the current chain state. The orange curves show the density of proposed moves from this state, with the left axis using a narrower proposal than the right. The violet curves show the proposal density scaled by the acceptance probability of the proposed move, this reducing the probability of transitions to states with lower density than the current state. The orange region between the violet and orange curves represents the probability mass reallocated to rejections by the downscaling by the acceptance function. The broader proposal in the right axis has an increased probability of making a move to the other mode in the target density but at a cost of an increased rejection probability.

Having now introduced the key theory underlying MCMC methods, we will now discuss practical implementations of the approach. In the following sub-sections we review two of the most popular frameworks for constructing reversible transition operators which leave a target distribution invariant: the *Metropolis–Hastings* algorithm and *Gibbs sampling*.

2.2.1 Metropolis–Hastings

Although the algorithm has come to be commonly known by Edward Metropolis' name as first author on the 1953 paper [54], it is believed that Arianna and Marshall Rosenbluth, two of the other co-authors, were the main contributors to the development of the algorithm [36].

The seminal *Metropolis–Hastings* algorithm provides a general framework for constructing Markov chains with a desired invariant distribution and is ubiquitous in MCMC methodology. The original Rosenbluth–Teller–Metropolis variant of the algorithm [54] dates to the very beginnings of the Monte Carlo method, having being first implemented on Los Alamos' MANIAC[5] one of the earliest programmable computers. The method was generalised in a key paper by Hastings [39], and the optimality among several competing alternatives of the form now used demonstrated by Peskun [66]. An extension to Markov chains on trans-dimensional spaces was proposed by Green [35].

An outline of the method is given in Algorithm 2 and a visualisation of its application to a univariate target distribution shown in Figure 2.7. The key idea is to propose updates to the state using an arbitrary trans-

5 *Mathematical Analyzer, Numerical Integrator and Computer.*

Algorithm 2 Metropolis–Hastings transition.

```
Input: $x_n$ : current chain state, $\tilde{p}$ : unnormalised target density,
   $q$ : normalised proposal density which we can sample according to.
Output: $x_{n+1}$ : next chain state with $x_n \sim p(\cdot) \implies x_{n+1} \sim p(\cdot)$.
1: $x^* \sim q(\cdot \,|\, x_n)$                                        ▷ Generate proposed new state
2: $u \sim \mathcal{U}(\cdot \,|\, 0, 1)$
3: if $u < \frac{\tilde{p}(x^*)\, q(x_n \,|\, x^*)}{\tilde{p}(x)\, q(x^* \,|\, x_n)}$ then
4:     $x_{n+1} \leftarrow x^*$                                         ▷ Proposed move accepted
5: else
6:     $x_{n+1} \leftarrow x_n$                                         ▷ Proposed move rejected
```

ition operator and then correct for this transition operator not necessarily leaving the target distribution invariant by stochastically accepting or rejecting the proposal. If a proposal is rejected the chain remains at the current state, otherwise the chain state takes on the proposed value. The transition density corresponding to Algorithm 2 is

$$\begin{aligned}\mathrm{t}(\boldsymbol{x}' \,|\, \boldsymbol{x}) = \alpha(\boldsymbol{x}' \,|\, \boldsymbol{x})\, q(\boldsymbol{x}' \,|\, \boldsymbol{x}) +& \\ \left(1 - \int_X \alpha(\boldsymbol{x}^* \,|\, \boldsymbol{x})\, q(\boldsymbol{x}^* \,|\, \boldsymbol{x})\, \mathrm{d}\mu(\boldsymbol{x}^*)\right)&\delta(\boldsymbol{x}' - \boldsymbol{x}),\end{aligned} \tag{2.32}$$

with the *acceptance probability* $\alpha : X \times X \to [0, 1]$ defined as

$$\alpha(\boldsymbol{x}' \,|\, \boldsymbol{x}) = \min\left\{1, \frac{q(\boldsymbol{x} \,|\, \boldsymbol{x}')\, p(\boldsymbol{x}')}{q(\boldsymbol{x}' \,|\, \boldsymbol{x})\, p(\boldsymbol{x})}\right\} = \min\left\{1, \frac{q(\boldsymbol{x} \,|\, \boldsymbol{x}')\, \tilde{p}(\boldsymbol{x}')}{q(\boldsymbol{x}' \,|\, \boldsymbol{x})\, \tilde{p}(\boldsymbol{x})}\right\}, \tag{2.33}$$

and $q : X \times X \to [0, \infty)$ the *proposal density*.

The original Rosenbluth–Teller–Metropolis algorithm used a symmetric proposal density $q(\boldsymbol{x}' \,|\, \boldsymbol{x}) = q(\boldsymbol{x} \,|\, \boldsymbol{x}')\ \forall \boldsymbol{x} \in X,\, \boldsymbol{x}' \in X$ (with the extension to the non-symmetric case being due to Hastings [39]), in which case the acceptance probability definition simplifies to

$$\alpha(\boldsymbol{x}' \,|\, \boldsymbol{x}) = \min\left\{1, \frac{p(\boldsymbol{x}')}{p(\boldsymbol{x})}\right\} = \min\left\{1, \frac{\tilde{p}(\boldsymbol{x}')}{\tilde{p}(\boldsymbol{x})}\right\}. \tag{2.34}$$

Note that in both (2.33) and (2.34) the target density only appears as a ratio and so only need be known up to a constant.

For the purposes of verifying the detailed balance condition (2.31), the density of *self-transitions*, i.e. a transition to the same state, can be ignored as (2.31) is trivially satisfied for $\boldsymbol{x}' = \boldsymbol{x}$. Considering therefore the cases $\boldsymbol{x} \neq \boldsymbol{x}'$ where the Dirac delta term representing the singular component corresponding to rejected proposals can be neglected, we have $\forall \boldsymbol{x} \in X,\, \boldsymbol{x}' \in X : \boldsymbol{x} \neq \boldsymbol{x}$

$$\begin{aligned}
\mathrm{t}(\boldsymbol{x}' \,|\, \boldsymbol{x})\, p(\boldsymbol{x}) &= \min\left\{1,\, \frac{q(\boldsymbol{x} \,|\, \boldsymbol{x}')\, p(\boldsymbol{x}')}{q(\boldsymbol{x}' \,|\, \boldsymbol{x})\, p(\boldsymbol{x})}\right\} q(\boldsymbol{x}' \,|\, \boldsymbol{x})\, p(\boldsymbol{x}) && (2.35)\\
&= \min\{q(\boldsymbol{x}' \,|\, \boldsymbol{x})\, p(\boldsymbol{x}),\, q(\boldsymbol{x} \,|\, \boldsymbol{x}')\, p(\boldsymbol{x}')\} && (2.36)\\
&= \min\left\{\frac{q(\boldsymbol{x}' \,|\, \boldsymbol{x})\, p(\boldsymbol{x})}{q(\boldsymbol{x} \,|\, \boldsymbol{x}')\, p(\boldsymbol{x}')},\, 1\right\} q(\boldsymbol{x} \,|\, \boldsymbol{x}')\, p(\boldsymbol{x}') && (2.37)\\
&= \mathrm{t}(\boldsymbol{x} \,|\, \boldsymbol{x}')\, p(\boldsymbol{x}'). && (2.38)
\end{aligned}$$

Therefore the detailed balance condition is satisfied, and the Metropolis–Hastings transition operator leaves the target distribution P invariant.

A special case for chains on a Euclidean state space $X = \mathbb{R}^D$, is when the proposal transition operator is deterministic and corresponds to a differentiable involution of the current state. Let $\boldsymbol{\phi} : X \to X$ be an involution, i.e. $\boldsymbol{\phi} \circ \boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{x}\ \forall X$ with Jacobian determinant $D_{\boldsymbol{\phi}}(\boldsymbol{x}) = \left|\frac{\partial \boldsymbol{\phi}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|$ which is defined and non-zero P-almost everywhere. Then if we define a transition operator via the transition density

$$\begin{aligned}
\mathrm{t}(\boldsymbol{x}' \,|\, \boldsymbol{x}) &= \delta(\boldsymbol{x}' - \boldsymbol{\phi}(\boldsymbol{x}))\alpha(\boldsymbol{x}) + \delta(\boldsymbol{x}' - \boldsymbol{x})(1 - \alpha(\boldsymbol{x})),\\
\alpha(\boldsymbol{x}) &= \min\left\{1,\, \frac{p \circ \boldsymbol{\phi}(\boldsymbol{x})}{p(\boldsymbol{x})} D_{\boldsymbol{\phi}}(\boldsymbol{x})\right\},
\end{aligned} \qquad (2.39)$$

then this transition operator will leave the target distribution P invariant. This deterministic transition operator variant is as a special case of the trans-dimensional Metropolis–Hastings extension introduced by Green [31, 35]. To generate from this transition operator from a current state $\boldsymbol{x}$ we compute the proposed move $\boldsymbol{\phi}(\boldsymbol{x})$ and accept the move with probablity $\alpha(\boldsymbol{x})$. We can demonstrate that this transition operator leaves P invariant by directly verifying (2.27)

$$\begin{aligned}
&\int_X \mathrm{t}(\boldsymbol{x}' \,|\, \boldsymbol{x})\, p(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} && (2.40)\\
&= \int_X \delta(\boldsymbol{x}' - \boldsymbol{\phi}(\boldsymbol{x}))\, \alpha(\boldsymbol{x})\, p(\boldsymbol{x}) + \delta(\boldsymbol{x}' - \boldsymbol{x})(1 - \alpha(\boldsymbol{x}))\, p(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} && (2.41)\\
&= \int_X \delta(\boldsymbol{x}' - \boldsymbol{y})\, \alpha \circ \boldsymbol{\phi}(\boldsymbol{y})\, p \circ \boldsymbol{\phi}(\boldsymbol{y})\, D_{\boldsymbol{\phi}}(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y} + (1 - \alpha(\boldsymbol{x}'))\, p(\boldsymbol{x}') && (2.42)\\
&= p(\boldsymbol{x}') + \alpha \circ \boldsymbol{\phi}(\boldsymbol{x}')\, p \circ \boldsymbol{\phi}(\boldsymbol{x}')\, D_{\boldsymbol{\phi}}(\boldsymbol{x}') - \alpha(\boldsymbol{x}')\, p(\boldsymbol{x}'). && (2.43)
\end{aligned}$$

In going from (2.42) to (2.43) we use a change of variables $\boldsymbol{y} = \boldsymbol{\phi}(\boldsymbol{x})$ in the integral. As $\boldsymbol{\phi}$ is an involution we have that $\boldsymbol{\phi} \circ \boldsymbol{\phi}(\boldsymbol{x}') = \boldsymbol{x}'$ and $D_{\boldsymbol{\phi}} \circ \boldsymbol{\phi}(\boldsymbol{x}') = D_{\boldsymbol{\phi}}(\boldsymbol{x}')^{-1}$ and so

$$\alpha \circ \boldsymbol{\phi}(\boldsymbol{x}') p \circ \boldsymbol{\phi}(\boldsymbol{x}') D_{\boldsymbol{\phi}}(\boldsymbol{x}') = \min\Big\{p \circ \boldsymbol{\phi}(\boldsymbol{x}') D_{\boldsymbol{\phi}}(\boldsymbol{x}'), p(\boldsymbol{x}')\Big\} = \alpha(\boldsymbol{x}') p(\boldsymbol{x}').$$

The last two terms in (2.43) therefore cancel and so (2.27) is satisfied by the transition operator defined by (2.39).

Although this transition operator leaves the target distribution P invariant, it is clear that it will not generate an ergodic Markov chain. Starting from a point $\boldsymbol{x}$ the next chain state will be either $\boldsymbol{\phi}(\boldsymbol{x})$ if the proposed move is accepted or $\boldsymbol{x}$ if rejected. In the former case the next proposed move will be to $\boldsymbol{\phi} \circ \boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{x}$ i.e. back to the original state. Therefore the chain will visit a maximum of two states. However as noted previously we can sequentially compose individual transition operators which all leave a target distribution invariant. Therefore a deterministic proposal Metropolis–Hastings transition can be combined with other transition operators to ensure the chain is irreducible and aperiodic.

In general for a Metropolis–Hastings transition operator to be irreducible, it is necessary that the proposal operator is irreducible [80], however this is not sufficient. For a target density which is positive everywhere on $X = \mathbb{R}^D$, then a sufficient but not necessary condition for irreducibility is that the proposal density is positive everywhere [71]. If the set of points with a non-zero probability of rejection has non-zero P-measure, then the transition operator is aperiodic [80].

A common choice of proposal density when the target distribution is defined on $\mathbb{R}^D$ is a multivariate normal density centred at the current state i.e. $q(\boldsymbol{x}' \,|\, \boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}' \,|\, \boldsymbol{x}, \Sigma)$ which satisfies the positivity condition for irreducibility. In general we would achieve optimal performance with a proposal density covariance Σ which is proportional to the covariance of the target distribution [74]. In practice we do not have access to the true covariance and so typically an isotropic proposal density is used with covariance $\Sigma = \sigma^2\mathbf{I}$ controlled by a single scale parameter σ, often termed the *step size* or *proposal width*. This proposal density is symmetric so the simplified acceptance rule (2.34) can be used, further the proposal density depends only on the difference $\boldsymbol{x}' - \boldsymbol{x}$ with Metropolis–Hastings methods having these properties often termed *random-walk Metropolis*.

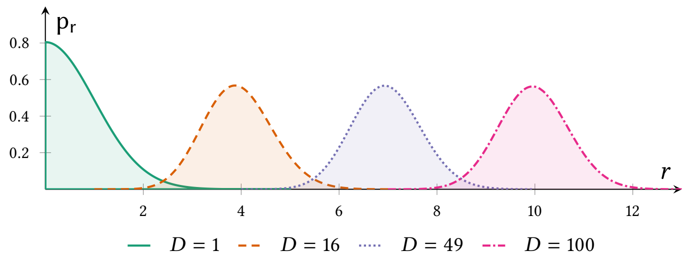


Figure 2.8: Illustration of concentration of measure in a multivariate normal distribution. The plots shows the probability density of the distance from the origin $\mathrm{r} = \|\mathbf{x}\|_2$ of a D-dimensional multivariate normal random vector $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for different dimensionalities D. As the dimension increases most of the mass concentrates away from the origin around a spherical shell of radius $\sqrt{D}$. For a multivariate normal random vector with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ this generalises to the mass being mainly in an ellipsoidal shell aligned with the eigenvectors of $\boldsymbol{\Sigma}$ and centred at $\boldsymbol{\mu}$.

Random walk Metropolis methods have been extensively theoretically studied, with sufficient conditions known in some cases to ensure geometric ergodicity of a chain [53, 73] though these can be hard to verify in practical problems. There has also been much work on practical guidelines and methods for tuning the free parameters in the algorithm, including approaches for tuning the step-size using acceptance rates [28, 70] and adaptive variants which automatically estimate a non-isotropic proposal covariance [37, 74].

In general the choice of proposal density will be key in determining the efficiency of Metropolis–Hastings MCMC methods. Ideally we want to be able to propose large moves in the state space to reduce the depencies between successive chain states and so increase the number of effective samples, however this needs to be balanced with maintaining a reasonable acceptance probability with large proposed moves often having a low acceptance probability. Figure 2.7 gives an illustration of this tradeoff in a one-dimensional example.

In high-dimensional spaces this issue is much more severe due to the phenomenon of *concentration of measure*: in probability distributions defined on high-dimensional spaces most of the probability mass will tend to be concentrated into a 'small' subset of the space [5, 49]. An illustration of this phenomenon for the multivariate normal distribution is shown in Figure 2.8, where the mass in high dimensions is mostly located in a thin ellipsoidal shell. The region where most of the mass concentrates, termed the *typical set* of the distribution, will for the tar-

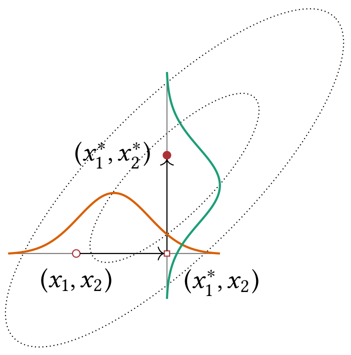


Figure 2.9: Schematic of Gibbs sampling transition in a bivariate normal target distribution (ellipses indicate constant density contours). Given an initial state $\mathbf{x} = (x_1, x_2)$, the x_1 (horizontal) co-ordinate is first updated by independently sampling from the normal conditional $\mathrm{p}_{\mathrm{x}_1|\mathrm{x}_2}(\cdot \,|\, x_2)$, represented by the orange curve. The new partially updated state is then $\mathbf{x} = (x_1^*, x_2)$. The second x_2 (vertical) co-ordinate is then independently resampled from the normal conditional $\mathrm{p}_{\mathrm{x}_2|\mathrm{x}_1}(\cdot \,|\, x_1^*)$, shown by the green curve. The final updated state is then $\mathbf{x} = (x_1^*, x_2^*)$.

Algorithm 3 Sequential scan Gibbs transition.

Input: $\boldsymbol{x}_n$: current chain state, I : ordered set of indices of all individual variables in chain state, $\{p_i\}_{i \in I}$: set of complete conditionals of target density p which can all be sampled from.
Output: $\boldsymbol{x}_{n+1}$: next chain state with $\boldsymbol{x}_n \sim p(\cdot) \implies \boldsymbol{x}_{n+1} \sim p(\cdot)$.

```
1: $\boldsymbol{x} \leftarrow \boldsymbol{x}_n$
2: for $i \in I$ do
3:     $x_i \sim p_i(\cdot \,|\, \boldsymbol{x}_{\backslash i})$          ▷ Resample $x_i$ from $p_i$ given current $\boldsymbol{x}_{\backslash i}$.
4: $\boldsymbol{x}_{n+1} \leftarrow \boldsymbol{x}$
```

get distributions of interest generally have a significantly more complex geometry. Finding proposals which can make large moves in such settings is challenging: moves in most directions will have a probability of acceptance which exponentially drops to zero as the distance away from the current state is increased and so simple proposal densities which ignore the geometry the typical set such as those used in random-walk Metropolis will need to make very small moves to have a reasonable probability of acceptance [10].

2.2.2 Gibbs sampling

Gibbs sampling [27, 29], originally proposed by Geman and Geman for image restoration using a Markov random field image model, is based on the observation that a valid transition operator for a joint target distribution across many variables, is one which updates only a subset of the variables and leaves the conditional distribution on that subset given the rest invariant. Although if used in isolation a transition operator which only updates some components of the state will not give an ergodic chain, as discussed previously multiple transition operators can be combined together to achieve ergodicity.

More specifically the original formulation of Gibbs sampling defines a Markov chain by sequentially independently resampling each individual variable in the model from its conditional distribution given the current values of the remaining variables. If I is an index set over the individual variables in the vector target state $\mathbf{x}$, then for each $i \in I$ we partition the state $\mathbf{x}$ into the i^{th} variable x_i and a vector containing all the remaining variables $\mathbf{x}_{\setminus i}$. For each $i \in I$ the target density can be factorised in to the marginal density $p_{\setminus i}$ on $\mathbf{x}_{\setminus i}$ and conditional density p_i on x_i given $\mathbf{x}_{\setminus i}$, i.e.

$$p(\boldsymbol{x}) = p_i(x_i \mid \boldsymbol{x}_{\setminus i})\, p_{\setminus i}(\boldsymbol{x}_{\setminus i}), \tag{2.44}$$

with the conditional densities $\{p_i\}_{i \in I}$ termed the *complete conditionals* of the target density. If each of these complete conditionals corresponds to a distribution we can generate samples from (for example using a transform method or rejection sampling) then we can apply the sequential Gibbs sampling transition operator defined in Algorithm 3 and visualised for a bivariate example in Figure 2.9.

The sequential Gibbs transition is irreducible and aperiodic under mild conditions [21, 72]. Rather than using a deterministic sequential scan through the variables, an alternative is to randomly sample without replacement the variable to update on each iteration; unlike the sequential scan version this defines a reversible transition operator. The random update variant is more amenable to theoretical analysis, however in practice the ease of implementation of the sequential scan variant and computational benefits in terms of memory access locality mean it seems to be more often used in practice [40]. A compromise between the completely random updates and a sequential scan is to randomly permute the update order after each complete scan.

A apparent advantage of Gibbs sampling over Metropolis–Hastings is the lack of a proposal density which needs to be tuned. This has helped popularise 'black-box' implementations of Gibbs sampling such as the probabilistic modelling packages BUGS [32] and JAGS [67]. A well-known issue with Gibbs sampling however is that its performance is highly dependent on the parameterisation used for the target density [69], with strong correlations between variables leading to large dependencies between successive states and slow convergence to stationarity. This can be alleviated in some cases by using a suitable reparameterisation to reduce dependencies between variables, however this restores the difficulty of tuning free parameters.

Gibbs sampling updates do not necessarily need to be performed by sampling from complete conditionals of single variables - in some cases the complete conditional of a vector of variables has a tractable form which can be sampled from as a 'block'; this motivates the name *block Gibbs samling* for such variants. By accounting for the dependencies between the variables in a block this can help alleviate some of the issues with highly correlated targets where applicable.

Compound terms such as *Metropolis-within-Gibbs* are sometimes used to refer to methods which sequentially apply Metropolis transition operators which each update only a subset of variables in the target distribution. We will however consider the defining feature of Gibbs sampling as being exact sampling from one or more conditionals rather than sequentially applying transition operators which update only subsets of variables and so will only refer to 'Gibbs sampling' in that context.

2.3 AUXILIARY VARIABLE METHODS

Although Gibbs sampling and random-walk Metropolis are commonly used in practice, as discussed above both have drawbacks when applied to complex high-dimensional target distributions. One approach which has proven particularly successful for constructing alternative Markov transition operators which can overcome some of these shortcomings is the introduction of *auxiliary variables* in to the chain state. For concreteness of notation in the following discussion we let the variables of interest, which we term the target variables, be represented by the random vector $\mathbf{x} \in X$ and the introduced auxiliary variables by the random vector $\mathbf{a} \in A$. We assume for generality here multiple auxiliary variables are introduced, however methods using a single scalar auxiliary variable are a common special case.

One way of defining a joint distribution across the target and auxiliary variables is to specify an arbitrary conditional distribution $\mathsf{P}_{\mathbf{a}|\mathbf{x}}$

and choose the marginal distribution $\mathsf{P}_{\mathbf{x}}$ to be equal to the target distribution P. Given samples from a joint distribution we can estimate expectations with respect to the marginal distribution on a subset of the variables by simply ignoring the dimensions of the sampled state we wish to marginalise over. Therefore if we can construct a Markov chain with the resulting joint distribution $\mathsf{P}_{\mathbf{x},\mathbf{a}}$ as its unique invariant distribution, then we can use the target variable components of the sampled states to estimate expectations with respect to the target distribution. We will consider two MCMC methods using this approach, *slice sampling* and *Hamiltonian Monte Carlo* in Sections 2.3.1 and 2.3.2.

An alternative approach is to instead construct a joint distribution on the target and auxiliary variables such that the regular conditional distribution $\mathsf{P}_{\mathbf{x}|\mathbf{a}}$ is equal to the target distribution P across some set of values $A^* \subset A$ of the auxiliary variables. In terms of the density $\mathsf{p}_{\mathbf{x}|\mathbf{a}}$ this requires that

$$\mathsf{p}_{\mathbf{x}|\mathbf{a}}(\boldsymbol{x} \mid \boldsymbol{a}) = p(\boldsymbol{x}) \qquad \forall \boldsymbol{x} \in X, \boldsymbol{a} \in A^*. \tag{2.45}$$

If the resulting marginal probability $\mathsf{P}_{\mathbf{a}}(A^*)$ is non-zero, then the sampled states $\{\boldsymbol{x}^{(n)}, \boldsymbol{a}^{(n)}\}_{n=1}^{N}$ of a Markov chain which has $\mathsf{P}_{\mathbf{x},\mathbf{a}}$ as its unique invariant distribution can be used to estimate expectations with respect to the target distribution by computing averages over only the sampled target variable values $\boldsymbol{x}^{(n)}$ for which the corresponding auxiliary variables $\boldsymbol{a}^{(n)}$ take values in A^* (these being at convergence samples from $\mathsf{P}_{\boldsymbol{x}|\boldsymbol{a}}(\cdot \mid \boldsymbol{a})$ and so the target distribution), i.e.

$$\int_X f(\boldsymbol{x}) \,\mathrm{d}P(\boldsymbol{x}) = \lim_{N\to\infty} \frac{\sum_{n=1}^{N} \mathbb{1}_{A^*}(\boldsymbol{a}^{(n)}) f(\boldsymbol{x}^{(n)})}{\sum_{n=1}^{N} \mathbb{1}_{A^*}(\boldsymbol{a}^{(n)})}. \tag{2.46}$$

We will discuss *simulated tempering*, an MCMC method which introduces an auxiliary variable in this manner in Section 2.3.3.

An issue with this approach is that if $\mathsf{P}_{\mathbf{a}}(A^*)$ is small, the number of sampled states with $\boldsymbol{a} \in A^*$ may be very small or even zero. This can require a large number of samples N for the sufficient samples with auxiliary variables in the required set A^* to be generated to allow the MCMC estimates computed using (2.46) to be reliable.

The estimator in (2.46) has a close resemblance to the formulation of the Monte Carlo estimator corresponding to rejection sampling given in (2.16), with averages computed over the subset of samples meeting

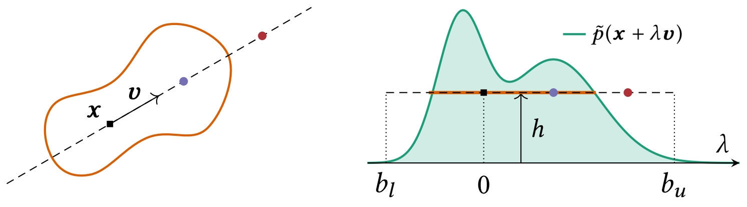


Figure 2.10: Schematic of linear slice sampling, showing 'plan' (left) and 'cross-sectional' (right) views of a bivariate target density. Orange curve (left) and line (right) indicates a constant density slice S_h. The black square indicates current target state value $\boldsymbol{x}$ and the dashed line is *slice line*, the one-dimensional linear sub-space aligned with the vector $\boldsymbol{v}$ which a new value from the state will be sampled on. The extents of the dashed line segment represent the initial bracket new proposed states will be drawn from. Points are proposed on the slice line by drawing a value uniformly from the current bracket. The red circle represents an initial proposed point which is not in the slice and so the right bracket edge is shrunk to this point. The violet circle shows a second sampled point from the new reduced bracket, this point within the slice and so returned as the updated target state.

an 'acceptance' criteria. As noted previously rejection sampling can in fact be considered as an auxiliary variable method, with binary accept indicator variables $\mathsf{a} \in \{0, 1\}$ introduced such that the conditional density $\mathsf{p}_{\mathbf{x}|\mathsf{a}}(\boldsymbol{x} \,|\, 1)$ is equal to the target density $p(\boldsymbol{x})$ as shown in (2.12), i.e. exactly corresponding to the property in (2.45). Rejection sampling can therefore be seen to be an example of this construct, though in this case each pair of target – auxiliary variable samples are generated independently rather than by constructing a Markov chain.

When discussing the rejection sampling estimator (2.16) we saw there was a close link to importance sampling estimator (2.17), with the importance sampling estimator having the potential advantage however of using all of the generated samples in computing estimates. In Chapter 5 we will discuss a related alternative approach to constructing estimators for auxiliary variable methods based on conditioning like simulated tempering, which unlike the estimator in (2.46) allows using all of the samples in a Markov chain to compute estimates.

2.3.1 Slice sampling

Slice sampling is a family of auxiliary variable MCMC methods which exploit the same observation as used to motivate rejection sampling - to sample from a target distribution it is sufficient to uniformly sample from the volume beneath a graph of the target density function. Rather

Algorithm 4 Linear slice sampling transition.

Input: $\boldsymbol{x}_n$: current chain state, $\tilde{p}$: unnormalised target density,
q : slice vector density, M : maximum number of step out iterations.
Output: $\boldsymbol{x}_{n+1}$: next chain state with $\boldsymbol{x}_n \sim p(\cdot) \implies \boldsymbol{x}_{n+1} \sim p(\cdot)$.

```
1:  $h \sim \mathcal{U}(\cdot \mid 0, \tilde{p}(\boldsymbol{x}_n))$                              ▷ Sample slice height
2:  $\boldsymbol{v} \sim q(\cdot)$              ▷ Sample vector setting slice line and initial bracket width
3:  $b_u \sim \mathcal{U}(\cdot \mid 0, 1)$               ▷ Uniformly sample bracket around current state
4:  $b_l \leftarrow b_u - 1$
5:  if $M > 0$ then $(b_l, b_u) \leftarrow$ LinearStepOut$(\boldsymbol{x}_n, b_l, b_u, M)$
6:  $\lambda \sim \mathcal{U}(\cdot \mid b_l, b_u)$
7:  while True do
8:      $\boldsymbol{x}^* \leftarrow \boldsymbol{x}_n + \lambda \boldsymbol{v}$                            ▷ Update proposed state
9:      if $\tilde{p}(\boldsymbol{x}^*) \leq h$ then                     ▷ Proposed point not on slice
10:         if $\lambda < 0$ then $b_l \leftarrow \lambda$ else $b_u \leftarrow \lambda$              ▷ Shrink slice bracket
11:         $\lambda \sim \mathcal{U}(\cdot \mid b_l, b_u)$                ▷ Sample uniformly from new bracket
12:     else                                                    ▷ Proposed state on slice
13:         return $\boldsymbol{x}^*$
14: function LinearStepOut$(\boldsymbol{x}_n, b_l, b_u, M)$
15:     $L \sim \mathrm{UniformInt}(\cdot \mid 0, M)$          ▷ Sample integer uniformly from $[0, M]$
16:     $U \leftarrow M - L$
17:     while $L > 0$ and $\tilde{p}(\boldsymbol{x}_n + b_l \boldsymbol{v}) > h$ do   ▷ Step out lower bracket edge
18:         $b_l \leftarrow b_l - 1$
19:         $L \leftarrow L - 1$
20:     while $U > 0$ and $\tilde{p}(\boldsymbol{x}_n + b_u \boldsymbol{v}) > h$ do  ▷ Step out upper bracket edge
21:         $b_u \leftarrow b_u + 1$
22:         $U \leftarrow U - 1$
23:     return $b_l, b_u$
```

than generate independent points from this volume as in rejection sampling, slice sampling instead constructs a transition operator which leaves the uniform distribution on this volume invariant.

The method we will concentrate on here was proposed by Neal [57, 58]. A related algorithm which uses per data-point auxiliary variables in Bayesian inference problems developed by Damien, Wakefield and Walker [24]. Murray, Adams and Mackay later proposed *elliptical slice sampling* [56], an extension of Neal's slice sampling method which is particularly effective for target distributions which are well approximated by a multivariate normal distribution.

Slice sampling defines a Markov chain on an augmented state space by introducing an auxiliary *height* variable $\mathsf{h} \in [0, \infty)$ in addition to the target variables $\mathbf{x} \in X$. The conditional density on h is

$$\mathsf{p}_{\mathsf{h}|\mathbf{x}}(h \mid \boldsymbol{x}) = \mathcal{U}(h \mid 0, \tilde{p}(\boldsymbol{x})) = \frac{1}{\tilde{p}(\boldsymbol{x})} \mathbb{1}_{[0, \tilde{p}(\boldsymbol{x}))}(h), \tag{2.47}$$

i.e. uniform over the interval between zero and the unnormalised target density value. The joint density on the augmented space is then

$$\mathsf{p}_{\mathbf{x},\mathsf{h}}(\boldsymbol{x}, h) = \frac{1}{\tilde{p}(\boldsymbol{x})} \mathbb{1}_{[0,\tilde{p}(\boldsymbol{x}))}(h) \frac{\tilde{p}(\boldsymbol{x})}{Z} = \frac{1}{Z} \mathbb{1}_{[0,\tilde{p}(\boldsymbol{x}))}(h). \tag{2.48}$$

Marginalising (2.48) over $\mathbf{x}$ recovers the target density i.e. $\mathsf{p}_{\mathbf{x}} = p$.

The overall slice sampling transition is formed of the sequential composition of a transition operator which updates h given $\mathbf{x}$ and a second operator which updates $\mathbf{x}$ given h, each leaving the distributions corresponding to the conditional densities $\mathsf{p}_{\mathsf{h}|\mathbf{x}}$ and $\mathsf{p}_{\mathbf{x}|\mathsf{h}}$ respectively invariant, and so by the same argument as for Gibbs sampling the overall transition leaving the target distribution invariant. By construction the conditional density $\mathsf{p}_{\mathsf{h}|\mathbf{x}}$ is a simple uniform density and so the first transition operator is a Gibbs sampling update in which the height variable is independently resampled from $\mathcal{U}(0, \tilde{p}(\boldsymbol{x}))$, where $\boldsymbol{x}$ is the current value of the target state $\mathbf{x}$.

The conditional density $\mathsf{p}_{\mathbf{x}|\mathsf{h}}(\boldsymbol{x} \mid h)$ is also locally uniform, equal to a positive constant whenever $\tilde{p}(\boldsymbol{x}) > h$ and zero elsewhere. However we can usually only evaluate the density up to an unknown constant as we cannot compute the measure of the set $S_h = \{\boldsymbol{x} \in X : \tilde{p}(\boldsymbol{x}) > h\}$ that the density is non-zero over. In general S_h, which is the eponymous *slice* of slice sampling (so called as it represents a slice through the volume under the density curve at a fixed height h), will have a complex geometry including potentially consisting of several disconnected components in the case of multimodal densities. The complexity of the slices generally prevents us therefore from being able to independently sample a new value for $\mathbf{x}$ uniformly from S_h and so we cannot use a full Gibbs sampling scheme corresponding to sequentially independently sampling from $\mathsf{P}_{\mathsf{h}|\mathbf{x}}$ and $\mathsf{P}_{\mathbf{x}|\mathsf{h}}$.

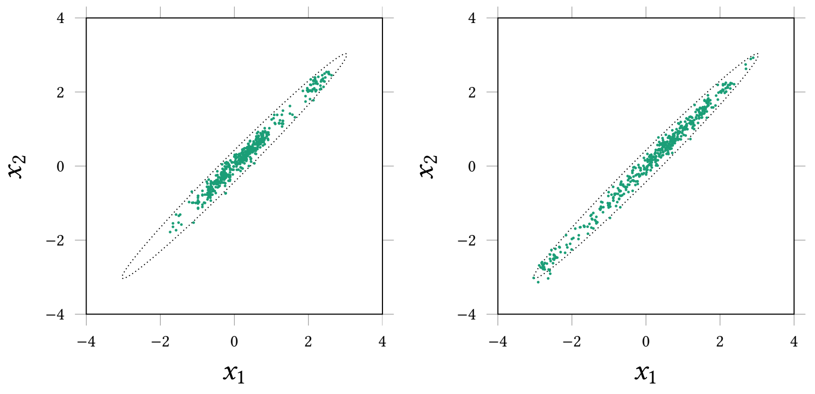
A key contribution of [58] was to introduce an elegant method for constructing a transition operator which leaves $\mathsf{P}_{\mathbf{x}|\mathsf{h}}$ invariant. In particular the algorithm has few free parameters to tune, has an efficiency which is relatively robust to the choices of the free choices that are introduced, and will for smooth target densities always move the target state by some amount (in contrast to the potential for rejections in Metropolis–Hastings methods). This method is summarised in Algorithm 4 and a visualisation of the process shown in Figure 2.10.

An important first step in the algorithm is reducing the problem of generating a point uniformly on the multidimensional slice S_h to making a move on a one-dimensional linear subspace of this slice (motivating our naming of this algorithm *linear slice sampling*) which includes the current $\mathbf{x}$ state. In the original description of the algorithm in [58] the one-dimensional subspace is chosen to be axis-aligned, correponding to updating a single component of the target state.

In the case of an axis-aligned subspace the restriction of the slice to the one-dimensional subspace is entirely specified by the conditional density on the chosen variable component given the current values of the remaining components in the state. Slice sampling transitions for each variable in the target state can then be applied sequentially akin to Gibb sampling, but with the advantage over Gibbs of not requiring the complete conditionals to be of a tractable form which we can generate exact samples from. If conditional independency structure in the target density means the complete conditionals depend only on local subsets of variables in the target state using updates of this form has the advantage of exploiting this locality. As with Gibbs sampling however applying slice sampling in this manner makes performance strongly dependent on the parameterisation of the target density, with large magnitude correlations likely to lead to slow exploration of the space.

In [58] various multivariate extensions of the algorithm are suggested which could help counter this issue, however they add significant implementation complexity compared to the basic algorithm. A simpler alternative is to define the one-dimensional subspace as being the line defined by a randomly chosen vector and passing through the current value of $\mathbf{x}$. If this vector is generated independently of the current state this is sufficient to ensure the overall transition retains the correct invariant distribution.

If little is known about the target distribution a reasonable default choice is to sample a unit vector of the required dimensionality by generating a random zero-mean isotropic covariance multivariate normal vector and then scaling it to unit norm; if an approximate covariance matrix $\hat{\Sigma}$ is known for the target density then instead generating the vector from $\mathcal{N}(\mathbf{0}, \Sigma)$ prior to normalising might be a better choice (as it favours moves aligned with the principle eigenvectors of Σ) however in this case elliptical slice sampling, which we will discuss shortly, will often be a better choice.



(a) Axis-aligned updates (b) Random direction updates

Figure 2.11: Samples generated using (a) axis-aligned versus (b) random-direction linear slice sampling in a correlated bivariate normal distribution. In both cases 1000 transitions where performed (with random selection of axis to update on each iteration in (a)) with every second sampled state shown. the maximum number of step out iterations is $M = 4$ and the initial bracket width is fixed at $w = 1$. The dotted ellipse shows the contour of the target density which contains 0.99 of the mass. The random direction chain is able to explore the typical set of the target distribution more effectively in this case with the axis-aligned updates leading to slower diffusion along the major axis of the elliptical contour.

This random-direction slice sampling variant is discussed in comparison to elliptical slice sampling in [56]. It is also bears resemblance to the scheme proposed in [23] which uses the same auxiliary variable formulation as slice sampling, but there the random direction is chosen in $X \times [0, \infty)$ i.e. to update both $\mathbf{x}$ and h and not used with the remainder of Neal's slice sampling algorithm. An example comparison of applying axis-aligned and random-direction linear slice sampling updates to a strongly positively correlated bivariate normal target distribution is shown in Figure 2.11. In this toy example the isotropic random-direction updates are able to more effectively explore the target density.

The generation of the vector $\boldsymbol{v}$ determining the one-dimensional subspace of the slice the update is performed on is represented in Algorithm 4 by Line 2 by $\boldsymbol{v}$ being generated from a distribution with density q. As well as specifying the *direction* of the slice line, the vector $\boldsymbol{v}$ also specifies a scale along this line. In Neal's description of the algorithm this is represented by the explicit *bracket width* parameter w. Here instead we assume this parameter is implicitly defined by the Euclidean norm of the vector $\boldsymbol{v}$, through suitable choice of q this allowing for direction dependent scales and also the possibility of randomisation of the scale; as we will see shortly however compared to for example random-walk Metropolis updates with a normal proposal, linear slice sampling is much less sensitive to the choice of scale parameters, therefore a single fixed scale will often be sufficient.

Once the slice line direction and scale has been chosen, the remainder of the algorithm can be split into two stages: selection of an initial bracket on the slice line and including the point corresponding to the current state; iteratively uniformly sampling points within the current bracket, accepting the point if it is within the slice S_h otherwise shrinking the bracket and repeating. The algorithm proposed by Neal ensures both these stages are performed reversibly such that the detailed balance condition (2.31) is maintained.

The *slice bracket* defines a contiguous interval $\lambda \in [b_l, b_u]$ on the slice line $\boldsymbol{x}^*(\lambda) = \boldsymbol{x}_n + \lambda \boldsymbol{v}$ and always includes the point $\lambda = 0$ corresponding to the current state. The initial bracket is chosen by sampling a upper bound b_u uniformly from $[0, 1]$ and then setting $b_l \leftarrow b_u - 1$; in the λ slice line coordinate system this corresponds to a bracket width of one, however in general the slice line vector $\boldsymbol{v}$ can have non-unit length and so defines the initial bracket width in the target variable space. Randomising the positioning of the current state within the bracket ensures reversibility as the resulting bracket would have an equal probability (density) of being selected from any other point in the bracket (which the final accepted point will be within).

In general only a subset of the points in the current slice bracket will be within the slice S_h. As new states are proposed by sampling a point uniformly from the current bracket, the probability of such a proposal being in the slice will be equal to the proportion of the bracket that intersects with the slice S_h. In general therefore it is desirable for the bracket to include as much of the slice as possible while not making the proportion of the bracket intersecting with the slice too small such that many points need to be proposed before a point on the slice is found. The magnitude of $\boldsymbol{v}$ determines the initial bracket extents and so should ideally chosen based on any knowledge of the 'typical scale' of the target density. Often we will have little prior knowledge about such scaling however and the scale will often vary significantly across the target space, and so we may choose an initial bracket which includes only a small proportion of the slice.

The stepping out routine proposed by [58] and detailed in Lines 14 to 23 in Algorithm 4 is designed to counter this issue. The initial slice bracket $[b_l, b_u]$ is iteratively 'stepped-out' by incrementing / decrementing the upper / lower bracket bounds until the corresponding endpoint of the bracket lies outside the slice or a pre-determined maximum number of steps out have been performed. Ideally the step out routine will return a bracket which contains all of the intersection of the slice with slice line while not also including too great a proportion of off slice points; in general the slice may be non-convex or consist of multiple disconnected components and so the intersection of the slice line with the slice may consist of multiple disconnected intervals in which case the stepping out routine will likely only expand the slice to include a subset of these intervals. The adaptivity provided by the stepping out routine will still however generally help to make the performance of the sampler much less sensitive to the choice of the bracket scale in contrast to for example random-walk Metropolis algorithms.

Analagously to the randomisation of the initial bracket positioning, in the stepping out routine if a maximum number of step out iterations M is set, the resulting step 'budget' is randomly allocated between increments of the upper bound b_u and decrements of the lower bound b_l such that final extended bracket generated by the step out routine would have an equal probability of being generated from any point within the generated bracket interval. If M is set to zero this correponds to not performing any stepping out and simply using the initial sampled bracket; although reducing the robustness of the algorithm to the choice of the initial bracket width this option has the advantage of minimising the number of target density evaluations by not requiring additional density evaluations at the bracket endpoints during the step-out routine. An alternative 'doubling' step-out routine was also proposed in [58]. This has the advantage of exponentially expanding the slice bracket compared to the linear growth of the step-out routine described in Algorithm 4 and so can be more efficient in target distributions where the typical scales of the density varies across several orders of magnitude. The doubling procedure requires a more complex subsequent procedure for sampling points in the resulting bracket however to ensure reversibility.

Once the initial bracket has been generated and potentially stepped out, the remainder of the algorithm consists of finding a point on the slice

Algorithm 5 Elliptical slice sampling transition.

Input: $\boldsymbol{x}_n$: current chain state, $\tilde{p}$: unnormalised target density,
$\boldsymbol{\mu}, \boldsymbol{\Sigma}$: mean and covariance of normal approximation to target.
Output: $\boldsymbol{x}_{n+1}$: next chain state with $\boldsymbol{x}_n \sim p \implies \boldsymbol{x}_{n+1} \sim p$.

```
1:  $h \sim \mathcal{U}(\cdot \,|\, 0, \tilde{p}(\boldsymbol{x}_n)/\mathcal{N}(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}))$    ▷ Sample slice height
2:  $\boldsymbol{v} \sim \mathcal{N}(\cdot \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma})$    ▷ Sample vector setting slice ellipse
3:  $\theta_u \sim \mathcal{U}(\cdot \,|\, 0, 2\pi)$    ▷ Uniformly sample bracket around current state
4:  $\theta_l \leftarrow \theta_u - 2\pi$
5:  $\theta \leftarrow \theta_u$
6:  while TRUE do
7:      $\boldsymbol{x}^* \leftarrow (\boldsymbol{x}_n - \boldsymbol{\mu})\cos\theta + (\boldsymbol{v} - \boldsymbol{\mu})\sin\theta + \boldsymbol{\mu}$    ▷ Update proposed state
8:      if $\tilde{p}(\boldsymbol{x}^*)/\mathcal{N}(\boldsymbol{x}^* \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \leq h$ then    ▷ Proposed point not on slice
9:          if $\theta < 0$ then $\theta_l \leftarrow \theta$ else $\theta_u \leftarrow \theta$    ▷ Shrink slice bracket
10:         $\theta \sim \mathcal{U}(\cdot \,|\, \theta_l, \theta_u)$    ▷ Sample uniformly from new bracket
11:     else
12:         return $\boldsymbol{x}^*$
```

line bracket which is within the slice S_h. This is done in an iterative manner by first sampling a point uniformly from the current bracket and checking if it is in the slice or not. If the proposed point is in the slice, the corresponding value for the target variables is returned at the new state. Otherwise the proposed point is set as the new upper or lower bound of the bracket such that the point corresponding to the current state remains in the bracket. This shrinks the bracket by removing an interval where it is known at least some points are not in the slice. A new point is then sampled uniformly from the smaller bracket and the procedure repeats until a point in the slice is found.

The iterative shrinking of the slice bracket implemented by this procedure introduces a further level of adaptivity in to the slice sampling algorithm, meaning that even if only a small proportion of the initial bracket lies within the slice only relatively few iterations will be needed still till the bracket is shrunk sufficiently for there to be a high probability of proposing a point within the bracket. By ensuring the point corresponding to the current state always remains within the current bracket, reversibility is maintained.

An alternative to the linear slice sampling procedure just described, is the *elliptical slice sampling* method proposed in [56] and described in Algorithm 5. As suggested by the name, in elliptical slice sampling rather than proposing points on a line instead an elliptical path in the target space is defined and new points proposed on this ellipse.

Elliptical slice sampling is intended for use in target distributions which can be approximated by a known normal distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$. This distribution might correspond to a normal prior distribution on model latent variables where the dependence between the latent and observed variables is only weak and so the posterior remains well approximated by the prior or a normal approximation fitted directly to the target distribution using an optimisation based approximate inference scheme such as those discussed in Appendix C [62].

In each elliptical slice sampling transition an auxiliary vector $\boldsymbol{v}$ is independently sampled from the normal distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$. If the target distribution was exactly described by the normal distribution we could use this independent draw directly as the new chain state (though obviously in this case there would be no advantage in formulating as an MCMC method). In reality the target distribution will only approximately described by $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ and so we wish to instead use this independent draw to define a Markov transition operator that will potentially move the state to a point nearly independent of the current state, but is also able to back off to more conservative proposals closer to the current chain state. This is achieved by defining an elliptical path in target space centred at $\boldsymbol{\mu}$, passing through the current state $\boldsymbol{x}_n$ and the auxiliary vector $\boldsymbol{v}$ and parameterised by an angular variable θ

$$\boldsymbol{x}^*(\theta) = (\boldsymbol{x}_n - \boldsymbol{\mu}) \cos\theta + (\boldsymbol{v} - \boldsymbol{\mu}) \sin\theta + \boldsymbol{\mu}. \tag{2.49}$$

If we generated θ uniformly from $\mathcal{U}(0, 2\pi)$ then the corresponding proposed transition $\boldsymbol{x}^*(\theta)$ would exactly leave the distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ invariant. As we instead wish to leave the target distribution invariant, a slice sampling algorithm is used to find a θ which accounts for the difference between the target distribution and normal approximation. An auxiliary slice height variable h is sampled uniformly from $\mathcal{U}(0, \tilde{p}(\boldsymbol{x}_n) / \mathcal{N}(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}, \Sigma))$ and used to define a slice

$$S_h = \left\{ \boldsymbol{x} \in X : \frac{\tilde{p}(\boldsymbol{x}_n)}{\mathcal{N}(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}, \Sigma)} < h \right\}. \tag{2.50}$$

Similar to the linear slice sampling algorithm, a bracket $[\theta_l, \theta_u]$ on the elliptical path is randomly placed around $\theta = 0$ corresponding to the current state $\boldsymbol{x}_n$. Unlike the requirement to choose a suitable initial bracket width in linear slice sampling however, we can define the initial bracket in elliptical slice sampling to include the entire elliptical path i.e. $\theta_l = \theta_u - 2\pi$; we only need to randomise the 'cut-point' defining the initial end-points of the bracket to ensure reversibility. This removes the need to choose an initial bracket width (defined by $|\boldsymbol{v}|$ in our description of the linear slice algorithm) and for any step out procedure, and so beyond choosing the multivariate normal approximation elliptical slice sampling does not have any free settings which need to be tuned.

Once the initial bracket is defined, a directly analagous iterative procedure to that used in the linear slice sampling algorithm is used to find a θ value corresponding to a point in the slice while using rejected proposed points to shrink the bracket. As with linear slice sampling, providing the target density is a smooth function and so the intersection of the elliptical path with the slice is a non-zero measure set, then the state moved to by the elliptical slice sampling transition operator will never be equal to the previous state.

2.3.2 Hamiltonian Monte Carlo

The MCMC algorithms discussed so far have required only the ability to evaluate a (unnormalised) density function for the target distribution of interest. For distributions defined on real-valued variables the target density function $\tilde{p}$ will often be differentiable - the gradient $\frac{\partial \tilde{p}}{\partial \boldsymbol{x}}$ exists P-almost everywhere. In these cases it is natural to consider using the gradient to help guide updates to the state. In particular we might hope to reduce the random-walk behaviour of simpler methods which leads to a slow diffusive exploration of high-dimensional spaces.

What we refer to as Hamiltonian Monte Carlo here was introduced in [26] as Hybrid Monte Carlo. *The alternative* Hamiltonian Monte Carlo *was suggested by MacKay [49] to emphasise the role of Hamiltonian dynamics in the method while maintaining the same acronym [11].*

A particularly powerful auxiliary variable MCMC method utilising gradient information is *Hamiltonian Monte Carlo* (HMC) [26, 60]. HMC introduces auxiliary *momentum* variables in to the chain state and then uses simulated Hamiltonian dynamics trajectories in the augmented space to generate proposed updates to the momentum–target variables state pair. The simulated Hamiltonian dynamics exhibit key geometric properties that make HMC well suited to performing MCMC in complex target distributions on high-dimensional spaces [13], with the method able to propose long-range moves with a high probability of acceptance. The reduced random-walk behaviour means that HMC often scales better to high-dimensional target distributions than simpler methods such as random-walk Metropolis and Gibbs sampling [10]. Under the assumption of a target distribution consisting of a product of independent factors on D variables, optimally tuned random-walk Metropolis transitions will take $\mathcal{O}(D^2)$ computational effort to achieve near independence between states of the chain while an optimally tuned HMC transition can achieve the same with a $\mathcal{O}(D^{\frac{5}{4}})$ cost [60].

Most implementations of HMC require the target density p is defined with respect to the Lebesgue measure on a Euclidean space $X = \mathbb{R}^D$. Target densities with bounded support on $\mathbb{R}^D$ add complication to the algorithm by requiring checks that proposed updates to the state remain within the support of the target distribution and reflecting at the boundaries of the support [60]. Often however a change of variables can be performed with a bijective transformation (using Equation 1.22) that maps to a density with unbounded support, for example taking a log-transform of a positive variable.

Rather than working directly with the unnormalised target density $\tilde{p}$ the HMC algorithm is more naturally described in terms of a *potential energy* function $\phi : \mathbb{R}^D \to \mathbb{R}$ which is related to $\tilde{p}$ by

$$\tilde{p}(\boldsymbol{x}) = \exp(-\phi(\boldsymbol{x})) \iff \phi(\boldsymbol{x}) = -\log \tilde{p}(\boldsymbol{x}). \tag{2.51}$$

The original *target variables* $\mathbf{x} \in \mathbb{R}^D$ are augmented with a vector of *momentum variables* $\mathbf{p} \in \mathbb{R}^D$. The conditional density on the momenta given the target variables $\mathsf{p}_{\mathbf{p}|\mathbf{x}}$ is defined in terms of a *kinetic energy* function $\tau : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ which is even in its first argument

$$\mathsf{p}_{\mathbf{p}|\mathbf{x}}(\boldsymbol{p}|\boldsymbol{x}) \propto \exp(-\tau(\boldsymbol{p} \,|\, \boldsymbol{x})). \tag{2.52}$$

The joint density on the momentum and target variables is then

$$\mathsf{p}_{\mathbf{x},\mathbf{p}}(\boldsymbol{x},\boldsymbol{p}) \propto \exp(-\phi(\boldsymbol{x}) - \tau(\boldsymbol{p} \,|\, \boldsymbol{x})) = \exp(-h(\boldsymbol{x},\boldsymbol{p})). \tag{2.53}$$

The function $h(\boldsymbol{x},\boldsymbol{p}) = \phi(\boldsymbol{x}) + \tau(\boldsymbol{p} \,|\, \boldsymbol{x})$ is termed the *Hamiltonian* for the system. A common simplification is for the momenta to be chosen to be independent of the target variables with a marginal density defined by a kinetic energy $\tau : \mathbb{R}^D \to \mathbb{R}$

$$\mathsf{p}_{\mathbf{p}}(\boldsymbol{p}) \propto \exp(-\tau(\boldsymbol{p})). \tag{2.54}$$

In this case the Hamiltonian $h(\boldsymbol{x},\boldsymbol{p}) = \phi(\boldsymbol{x}) + \tau(\boldsymbol{p})$ is *separable* - there are no terms jointly dependent on both $\boldsymbol{x}$ and $\boldsymbol{p}$.

Commonly a quadratic form $\tau(\boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}^{\mathsf{T}}\boldsymbol{M}^{-1}\boldsymbol{p}$ is used for the kinetic energy where $\boldsymbol{M}$ is a positive definite matrix typically termed the *mass matrix*. A quadratic kinetic energy corresponds to assuming normally distributed momenta with zero-mean and covariance $\boldsymbol{M}$.

In classical mechanics, the Hamiltonian describes the total energy of a mechanical system, and can be used to define a *canonical Hamiltonian dynamic* via the set of *ordinary differential equations* (ODEs)

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \frac{\partial h}{\partial \boldsymbol{p}}^{\mathsf{T}}, \qquad \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\frac{\partial h}{\partial \boldsymbol{x}}^{\mathsf{T}}. \tag{2.55}$$

We define the *flow map* corresponding to this dynamic as a family of mappings $\boldsymbol{\psi}_t : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}^D \times \mathbb{R}^D$ parameterised by a time $t \in \mathbb{R}$ such that if $(\boldsymbol{x}(t), \boldsymbol{p}(t))$ is the solution to the set of ODEs (2.55) at a time t given an initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0, \boldsymbol{p}(0) = \boldsymbol{p}_0$ then

$$\boldsymbol{\psi}_t(\boldsymbol{x}_0, \boldsymbol{p}_0) = (\boldsymbol{x}(t), \boldsymbol{p}(t)). \tag{2.56}$$

The Hamiltonian flow map has several desirable properties as a proposal generating mechanism for a MCMC method. The Hamiltonian is exactly conserved along the trajectories generated by the flow map, i.e. $h(\boldsymbol{x}, \boldsymbol{p}) = h \circ \boldsymbol{\psi}_t(\boldsymbol{x}, \boldsymbol{p})$ for all $t \in \mathbb{R}$ and for any initial $\boldsymbol{x}, \boldsymbol{p}$ pair. As $\mathsf{p}_{\mathbf{x},\mathbf{p}}(\boldsymbol{x}, \boldsymbol{p}) \propto \exp(-h(\boldsymbol{x}, \boldsymbol{p}))$ this means Hamiltonian trajectories remain confined to constant density surfaces in the augmented state space.

The Hamiltonian flow map is also *volume preserving* - the Jacobian of the flow map, $\mathbf{J}_{\boldsymbol{\psi}_t}$ has determinant one for all t and starting from any initial $(\boldsymbol{x}, \boldsymbol{p})$. This volume preservation is a consequence of a stronger geometric property of the dynamic - that the flow map is *symplectic* [48]. Symplecticity of the flow map is implied by the condition

$$\mathbf{J}_{\boldsymbol{\psi}_t}^{\mathsf{T}} \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} \mathbf{J}_{\boldsymbol{\psi}_t} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} \tag{2.57}$$

being satisfied. The symplectic nature of Hamiltonian dynamics is central to efficient scaling of HMC to high-dimensional spaces [13, 60].

A final crucial property of the Hamiltonian flow map is that it exhibits a time-reversal symmetry under negation of the momenta

$$(\boldsymbol{x}', \boldsymbol{p}') = \boldsymbol{\psi}_t(\boldsymbol{x}, \boldsymbol{p}) \iff (\boldsymbol{x}, -\boldsymbol{p}) = \boldsymbol{\psi}_t(\boldsymbol{x}', -\boldsymbol{p}'). \tag{2.58}$$

If we define $\boldsymbol{r} : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}^D \times \mathbb{R}^D$ as a 'momentum-reversal' operator such that $\boldsymbol{r}(\boldsymbol{x}, \boldsymbol{p}) = (\boldsymbol{x}, -\boldsymbol{p})$ then this time-reversal symmetry means that the composition $\boldsymbol{r} \circ \boldsymbol{\psi}_t$ is an involution. Further as $\boldsymbol{r}$ also has Jacobian determinant one, then the composition $\boldsymbol{r} \circ \boldsymbol{\psi}_t$ also itself has a unit Jacobian determinant.

Using the previous result for the Metropolis–Hastings accept ratio for the special case of a deterministic proposal formed by an involution (2.39), we have that a proposal generated by applying $\boldsymbol{r} \circ \boldsymbol{\psi}_t$ to the current state pair $(\boldsymbol{x}, \boldsymbol{p})$ for any t has an accept probability of one

$$\alpha(\boldsymbol{x}, \boldsymbol{p}) = \min\left\{1, \frac{\exp(-h \circ \boldsymbol{r} \circ \boldsymbol{\psi}_t(\boldsymbol{x}, \boldsymbol{p}))}{\exp(-h(\boldsymbol{x}, \boldsymbol{p}))} \left| \mathbf{J}_{\boldsymbol{r} \circ \boldsymbol{\psi}_t}(\boldsymbol{x}, \boldsymbol{p}) \right| \right\} \tag{2.59}$$

$$= \min\{1, \exp\big(h(\boldsymbol{x}, \boldsymbol{p}) - h \circ \boldsymbol{r} \circ \boldsymbol{\psi}_t(\boldsymbol{x}, \boldsymbol{p})\big)\} = 1. \tag{2.60}$$

This is a result of the conservation of the Hamiltonian under the flow map (and momentum-reversal operator as τ is even in the momenta) and the composed map having unit Jacobian determinant. Therefore proposals formed by integrating the ODEs forward by some length of time from the current state and then reversing the momentum would always be accepted.

On its own the momentum reversal operator $\boldsymbol{r}$ is also an involution with unit Jacobian determinant which exactly conserves the Hamiltonian, and so can also be applied as a 'proposal' with probability of acceptance of one. If we sequentially alternate updates using $\boldsymbol{r} \circ \boldsymbol{\psi}_t$ and $\boldsymbol{r}$, each defines a valid Markov transition operator which leaves the (extended) target invariant, and in sequential composition the momentum reversals cancel. We can therefore construct a Markov chain which leaves the target distribution with density (2.53) invariant by repeatedly generating new states by integrating the Hamiltonian dynamic forward by arbitary lengths of time. Note that though each of $\boldsymbol{r} \circ \boldsymbol{\psi}_t$ and $\boldsymbol{r}$ are individually reversible and so respect detailed balance, the sequential composition is no longer reversible.

There are two major problems with this scheme. Firstly for most ϕ and τ it is not possible to integrate the ODEs (2.55) exactly and so we cannot evaluate the exact flow map $\boldsymbol{\psi}_t$. Secondly the scheme as proposed would not be ergodic as the Hamiltonian is conserved by each application of the flow map, and so all states generated in this way would be confined to a constant Hamiltonian manifold in the joint space.

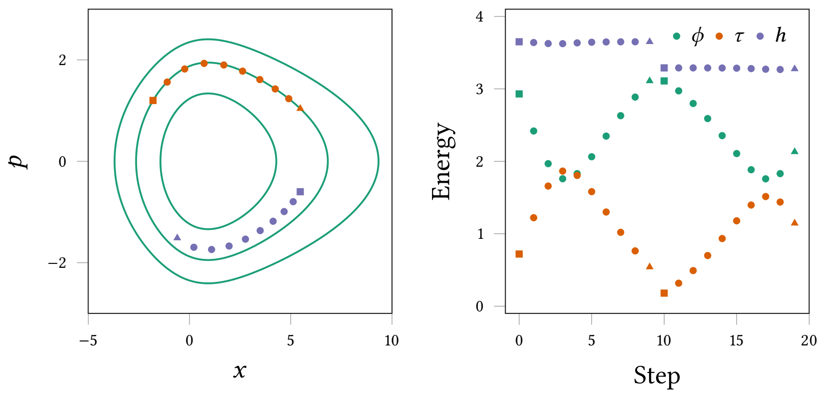


Figure 2.12: Visualisation of Hamiltonian Monte Carlo applied to a univariate target density $p(x) = \alpha \exp(-\alpha x)(1 + \exp(-x))^{-1-\alpha}$ with $\alpha = 0.4$. The left axis shows contours (green curves) of the Hamiltonian function on the augmented (x, p) state space. The orange markers shows a Hamiltonian trajectory simulated using the leapfrog method, starting at the square marker and finishing at the triangular marker. The trajectory nearly exactly traces a Hamiltonian contour due to the approximate energy conservation of the simulated dynamic, with the proposed update (from square to triangular markers) therefore accepted with high probability. At the end of the orange trajectory the momentum is randomly resampled, giving a new initial state (purple square marker) for a second simulated trajectory shown by the purple markers. The right axis shows the variation in the Hamiltonian h, potential energy ϕ and kinetic energy τ over the two trajectories. In each trajectory the Hamiltonian is close to constant, with shifts in the potential energy matched by opposing shifts in the kinetic energy. There is a step change in the kinetic energy and Hamiltonian when then momentum is resampled at the end of first trajectory.

The first issue can be resolved by approximately integrating the ODEs. Importantly by using a *symplectic integrator* we are able to form approximate Hamiltonian flow maps which maintain the key volume-preservation and time-reversibility properties of the exact flow map dynamic and define, as the name suggests, symplectic maps. There is a large class of such symplectic integrators [48] however for separable Hamiltonians an appealingly simple scheme is the *Störmer–Verlet* or *leapfrog* integrator. If we first define the following component maps

$$\hat{\boldsymbol{\psi}}^{\mathrm{A}}_{\delta t}(\boldsymbol{x},\boldsymbol{p}) = \left(\boldsymbol{x}, \boldsymbol{p} - \delta t \nabla\phi(\boldsymbol{x})^{\mathsf{T}}\right), \quad \hat{\boldsymbol{\psi}}^{\mathrm{B}}_{\delta t}(\boldsymbol{x},\boldsymbol{p}) = \left(\boldsymbol{x} + \delta t \nabla\tau(\boldsymbol{p})^{\mathsf{T}}, \boldsymbol{p}\right) \tag{2.61}$$

then a leapfrog step is defined by the symmetric composition

$$\hat{\boldsymbol{\psi}}^{\mathrm{LF}}_{\delta t} = \hat{\boldsymbol{\psi}}^{\mathrm{A}}_{\frac{\delta t}{2}} \circ \hat{\boldsymbol{\psi}}^{\mathrm{B}}_{\delta t} \circ \hat{\boldsymbol{\psi}}^{\mathrm{A}}_{\frac{\delta t}{2}}. \tag{2.62}$$

Each leapfrog step is time-reversible and volume conserving. The composition of L leapfrog steps $\left(\hat{\boldsymbol{\psi}}^{\mathrm{LF}}_{\delta t}\right)^L$ maintains these properties. The alternative symmetric composition $\hat{\boldsymbol{\psi}}^{\mathrm{B}}_{\frac{\delta t}{2}} \circ \hat{\boldsymbol{\psi}}^{\mathrm{B}}_{\delta t} \circ \hat{\boldsymbol{\psi}}^{\mathrm{B}}_{\frac{\delta t}{2}}$ also defines a symplectic integrator and is also sometimes termed the leapfrog method. In practice when multiple leapfrog steps are composed together the intermediate half time-steps can be combined, for example using (2.62)

$$\left(\hat{\boldsymbol{\psi}}^{\mathrm{A}}_{\frac{\delta t}{2}} \circ \hat{\boldsymbol{\psi}}^{\mathrm{B}}_{\delta t} \circ \hat{\boldsymbol{\psi}}^{\mathrm{A}}_{\frac{\delta t}{2}}\right) \circ \left(\hat{\boldsymbol{\psi}}^{\mathrm{A}}_{\frac{\delta t}{2}} \circ \hat{\boldsymbol{\psi}}^{\mathrm{B}}_{\delta t} \circ \hat{\boldsymbol{\psi}}^{\mathrm{A}}_{\frac{\delta t}{2}}\right) = \hat{\boldsymbol{\psi}}^{\mathrm{A}}_{\frac{\delta t}{2}} \circ \hat{\boldsymbol{\psi}}^{\mathrm{B}}_{\delta t} \circ \hat{\boldsymbol{\psi}}^{\mathrm{A}}_{\delta t} \circ \hat{\boldsymbol{\psi}}^{\mathrm{B}}_{\delta t} \circ \hat{\boldsymbol{\psi}}^{\mathrm{A}}_{\frac{\delta t}{2}}$$

and so the two different symmetric compositions only differ by whether initial and final half momentum or position time steps are taken.

Although an approximate flow map will no longer exactly conserve the Hamiltonian, a key property of symplectic integrators, including the leapfrog method, is that they correspond to the exact flow map of a modified Hamiltonian system. Providing the integrator step-size δt is below a stability threshold this modified Hamiltonian $\tilde{h}_{\delta t}$ will be close to the original target Hamiltonian: $\left|h(\boldsymbol{x},\boldsymbol{p}) - \tilde{h}_{\delta t}(\boldsymbol{x},\boldsymbol{p})\right| \leq O(\delta t^k)$ where k is the order of the integrator ($k = 2$ for the leapfrog method) [48]. As the approximate flow map exactly conserves this modified Hamiltonian, this means that the change in the Hamiltonian over long simulated trajectories will remain bounded. If we replace the exact flow map for the approximate flow map corresponding to L steps of the leapfrog integrator with step size δt in (2.59) then we have that the probability of accepting a proposal generated by approximately integrating the ODEs and then negating the momentum is

$$\alpha(\boldsymbol{x},\boldsymbol{p}) = \min\left\{1,\, \exp\left(h(\boldsymbol{x},\boldsymbol{p}) - h \circ \boldsymbol{r} \circ \left(\hat{\boldsymbol{\psi}}^{\mathrm{LF}}_{\delta t}\right)^L(\boldsymbol{x},\boldsymbol{p})\right)\right\}. \tag{2.63}$$

As the change in Hamiltonian over the trajectory remains bounded, if the step-size is small enough the probability of acceptance will remain close to one even for a large number of integrator steps L. This means simulated Hamiltonian dynamics can be used to form long-range proposed moves which maintain a high probability of acceptance. An example of this approximate conservation of the Hamiltonian is shown in Figure 2.12 which shows a visualisation of trajectories simulated using the leapfrog method in a system with a one-dimensional target variable x and the variation in the Hamiltonian, potential and kinetic energies along these trajectories.

Algorithm 6 Hamiltonian Monte Carlo transition.

Input: $\boldsymbol{x}_n$: current target variables state, ϕ : differentiable potential energy function $= -\log \tilde{p}$, δt : leapfrog integrator step size, L : number of leapfrog integration steps, $\boldsymbol{M}$: mass matrix.

Output: $\boldsymbol{x}_{n+1}$: next chain state with $\boldsymbol{x}_n \sim p \implies \boldsymbol{x}_{n+1} \sim p$.

```
1:  $\boldsymbol{p} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{M})$    ▷ Independently sample new momentum vector.
2:  $\boldsymbol{p}^* \leftarrow \boldsymbol{p} - \frac{\delta t}{2} \nabla\phi(\boldsymbol{x}_n)^{\mathsf{T}}$    ▷ Initial half momentum step $\hat{\boldsymbol{\psi}}^{\mathrm{A}}_{\frac{\delta t}{2}}$.
3:  $\boldsymbol{x}^* \leftarrow \boldsymbol{x}_n + \delta t \boldsymbol{M}^{-1} \boldsymbol{p}^*$    ▷ $\hat{\boldsymbol{\psi}}^{\mathrm{B}}_{\delta t}$.
4:  for $s \in \{1 \dots L-1\}$ do
5:      $\boldsymbol{p}^* \leftarrow \boldsymbol{p}^* - \delta t \nabla\phi(\boldsymbol{x}^*)^{\mathsf{T}}$    ▷ $\hat{\boldsymbol{\psi}}^{\mathrm{A}}_{\frac{\delta t}{2}} \circ \hat{\boldsymbol{\psi}}^{\mathrm{A}}_{\frac{\delta t}{2}}$.
6:      $\boldsymbol{x}^* \leftarrow \boldsymbol{x}^* + \delta t \boldsymbol{M}^{-1} \boldsymbol{p}^*$    ▷ $\hat{\boldsymbol{\psi}}^{\mathrm{B}}_{\delta t}$.
7:  $\boldsymbol{p}^* \leftarrow \boldsymbol{p}^* - \frac{\delta t}{2} \nabla\phi(\boldsymbol{x}^*)^{\mathsf{T}}$    ▷ Final half momentum step $\hat{\boldsymbol{\psi}}^{\mathrm{A}}_{\frac{\delta t}{2}}$.
8:  $u \sim \mathcal{U}(0,1)$
9:  $\alpha \leftarrow \exp\left(\phi(\boldsymbol{x}_n) + \frac{1}{2}\boldsymbol{p}\boldsymbol{M}^{-1}\boldsymbol{p} - \phi(\boldsymbol{x}^*) - \frac{1}{2}\boldsymbol{p}^*\boldsymbol{M}^{-1}\boldsymbol{p}^*\right)$    ▷ Accept probability.
10: if $u < \alpha$ then
11:     $\boldsymbol{x}_{n+1} \leftarrow \boldsymbol{x}^*$    ▷ Proposed move accepted.
12: else
13:     $\boldsymbol{x}_{n+1} \leftarrow \boldsymbol{x}_n$    ▷ Proposed move rejected.
14: return $\boldsymbol{x}_{n+1}$
```

When using an approximate flow map as there is now a non-zero probability of rejection, upon rejecting the momentum will remain at its current state, before then being negated. The next simulated trajectory will therefore backtrack along a previous trajectory after a rejection. To prevent this backtracking behaviour and resolve the issue that Markov transitions consisting solely of simulated dynamics proposals and momentum-reversals would remain confined to a constant modified Hamiltonian manifold, a further update is introduced in to the overall HMC transition which changes the momenta while leaving the target variables fixed.

For the common case of a quadratic kinetic energy $\tau(\boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}^{\mathsf{T}}\boldsymbol{M}^{-1}\boldsymbol{p}$ and so normal marginal distribution on the momenta, the simplest way to update the momenta is to independently sample a new vector from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{M})$ at the beginning of each HMC transition. This will both perturb the initial Hamiltonian of the system and also mean the initial direction of any simulated trajectory is randomised so that the negation of the previous momentum upon a rejection does not lead to backtracking. In this case as the momentum is independently resampled in each transition there is no need to store the momentum state between successive transitions and the overall HMC transition will be reversible.

Algorithm 6 describes the overall HMC transition corresponding to using a quadratic kinetic energy $\tau(\boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}^\mathsf{T}\boldsymbol{M}^{-1}\boldsymbol{p}$, independent momentum resampling and a leapfrog integrator. The integrator step size δt and number of steps L together determine the total approximate integration time $T = L\delta t$. Intuitively we want to choose T to minimise the dependence of the generated proposals on the current point. In general we will however not know what this optimal integration time is and in non-toy problems it will depend on the starting state. Choosing an appropriate integration time can therefore be challenging, and will often involve some level of trial and error with pilot runs [60]. Too small values lead to dynamics proposals which remain close to the current state which combined with the independent resampling of the momenta on each transition lead to random-walk like behaviour for the overall transition, with limited gain from using the HMC transition over simpler methods such as random-walk Metropolis.

The computational cost of each HMC transition will however scale linearly with L and so the integration time, therefore it is desirable to not increase the integration time beyond the point where there is any gain in decreased dependence between successive points; further as typically the simulated trajectories will be quasi-periodic increasing the integration time can in some cases lead to proposals moving closer to the original state. The integration time does not need to be the same for each transition and randomising it by for example uniformly sampling from an interval can be helpful in some problems to reduce pathological behaviour due to near periodicity of trajectories [60].

In combination with the integration time an appropriate value must also be chosen for the integrator step size δt. As for a fixed integration time T the step size determines the number integrator steps needed and so computational cost per transition, we ideally want to use as large a step size as possible. The step size however also controls how large the typical change in the Hamiltonian is across a simualted trajectory and so the accept rate for the proposed updates. As the step size increases the average accept rate will decrease and beyond some limit typically the dynamic will become unstable and the Hamiltonian error no longer remain bounded, leading to very low accept rates for large L. Typically this stability limit will vary depending on the starting state, so we may occasionally encounter unstable diverging trajectories even when the step size is small enough for most trajectories to remain stable.

Analagously to results for tuning the proposal width for random-walk Metropolis methods, guidelines have been derived for choosing the integrator step-size in HMC based on an optimal (in the sense of maximising some measure of computational efficiency) average accept probability for the Metropolis step. A target accept rate of 0.65 has been suggested [6, 60] under idealised assumptions of a high-dimensional target distribution in which the individual dimensions are independent and when using the leapgfrog integrator. Under more general assumptions in [12] an accept rate range of 0.6 to 0.9 was instead recommended as giving close to optimal performance for symplectic integrators of order 2 including the leapfrog method.

To help address the challenges of tuning the free integrator step-size and integration time parameters of the standard HMC algorithm, adaptive variants which automatically tune these parameters have been proposed. Of particular note is the *no U-turn sampler* (NUTS) algorithm [41]. Rather than using a single fixed integration time, NUTS dynamically varies the length of the simulated trajectories, expanding the trajectories until a termination criterion corresponding intuitively to the trajectory 'turning back on itself' (hence the name) is met and then using a slice sampling scheme to select a new state from this dynamically generated trajectory. Maintaining reversibility in such a scheme is non-trivial and NUTS uses an elegant recursive method to do so: full details are beyond the scope of this review but both [41] and a subsequent review article [10] provide excellent visual explanations of the algorithm. The dynamic integration time scheme is combined in NUTS with a stochastic optimisation method for tuning the integrator step-size to achieve a target acceptance rate, with a vanishing adaptation rate ensuring convergence of the chains to stationarity [3].

Refinements to NUTS have been suggested including generalised termination criteria [8, 9] and an extension to use multinomial sampling of the final state from the generated trajectory rather than slice sampling [9, 10]. The original NUTS algorithm and these refinements have seen widespread empirical success through their implementation in the probabilistic programming framework *Stan* [19] which combines a general purpose probabilistic model specification language [78] and automatic differentiation library [18] with efficient implementations of approximate inference methods including NUTS.

We have so far neglected to mention how the mass matrix $\boldsymbol{M}$ is chosen. The simplest choice is to use $\boldsymbol{M} = \mathbf{I}$; this is a reasonable choice when the target density has a close to isotropic geometry with approximately equal scaling of each dimension and no strong correlations between variables. Using a non-identity mass matrix is equivalent to running HMC with a identity mass matrix in a linearly transformed reparameterisation of the target density [60] and so can be used to account for non-isotropic target densities by rescaling and decorrelating the variables of the target distribution. Ideally the mass matrix would be chosen based on the covariance of the target distribution [10, 60] however in practice this will typically not be available. One option is to estimate the target covariance during an initial adaptive phase in the chain which is used within the NUTS implementation in Stan [19].

In reality for complex target distributions the geometry of the density will vary across the state space with position-dependent curvature. In these cases a single constant mass matrix will be ineffective at locally decorrelating and normalising the scale of the variables corresponding to the different dimensions of the target distribution. This can degrade the performance of HMC methods, with no single step size appropriate in all regions of the state space and typically a tradeoff needing to be made between choosing a small step size based on the scale of the most constrained directions and choosing a larger step size for efficiency with the possible result of the simulated dynamic being unable to enter tightly constrained regions of the target distribution [7].

For a quadratic kinetic energy function $\tau(\boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}^\mathsf{T}\boldsymbol{M}^{-1}\boldsymbol{p}$, considering the kinetic energy as a random variable $\mathrm{\tau} = \tau(\mathbf{p})$, as $\mathbf{p}$ has a multivariate normal marginal distribution, $\mathrm{\tau}$ will have mean $D/2$ and standard deviation $\sqrt{D}$ [60]. As the kinetic energy is bounded below by zero and the Hamiltonian and so sum of kinetic and potential energies approximately conserved along simulated trajectories, the maximal increase in the potential energy along a trajectory is approximately upper bounded by the initial kinetic energy. The potential energy will therefore typically vary by an amount of order D over simulated trajectories. For complex target distributions with varying curvature and scales, the potential energy will often vary by much more than D across the typical set of the distribution and so any single trajectory will typically be only able to cover a small region in the typical set, with exploration of the full typical set of the distribution then degrading to a random-walk like behaviour as only the momentum resampling steps allows movement up and down the full potential energy range [7, 10].

The naming of RMHMC arises from a connection to Riemannian geometry, with the metric defining a Riemannian manifold, with the length of shortest path between two points on the manifold, i.e. a geodesic, *locally defined by the metric.*

A suggested resolution to the issue of varying curvature across the target density is to use a mass matrix which depends on the target state $\mathbf{x}$. *Riemannian-manifold Hamiltonian Monte Carlo* (RMHMC) [34] defines a non-separable Hamiltonian using a kinetic energy function

$$\tau(\boldsymbol{p} \mid \boldsymbol{x}) = \frac{1}{2}\boldsymbol{p}^{\mathsf{T}}(\boldsymbol{G}(\boldsymbol{x}))^{-1}\boldsymbol{p} + \frac{1}{2}\log|\boldsymbol{G}(\boldsymbol{x})| \tag{2.64}$$

corresponding to $\mathsf{p}_{\mathbf{p}|\mathbf{x}}(\boldsymbol{p} \mid \boldsymbol{x}) = \mathcal{N}(\boldsymbol{p} \mid \mathbf{0}, \boldsymbol{G}(\boldsymbol{x}))$ where $\boldsymbol{G} : \mathbb{R}^D \to \mathbb{R}^{D \times D}$ is a positive-definite matrix function termed the *metric*. In analogy to the earlier mentioned equivalence between using a non-identity constant mass matrix and running HMC with an identity mass matrix in a reparameterised target distribution in terms of a linear transformation of the original target variables [60], RMHMC can be shown to be equivalent to running HMC with an identity mass matrix in a reparameterisation of the target distribution in terms of a *non-linear* transformation of the target variables [63]. This non-linear reparameterisation can locally transform the target distribution so that the resulting density has a geometry more amenable to exploration by the HMC dynamic.

Various schemes have been proposed for choosing a metric for a particular target distribution. In [34] the *Fisher–Rao metric* [2] is suggested as it provides a natural description of the Riemannian geometry of parametric probability distributions and so is particularly relevant for target distributions corresponding to the posterior of Bayesian inference problems for models of IID datasets. The Fisher–Rao metric only has a closed form solution however for a limited set of distributions. An alternative more generally applicable metric based on a regularisation of the Hessian of the log target density to ensure positive-definiteness was suggested in [7]. A 'geometrically tempered' metric designed to help exploration of multimodal distributions was suggested in [63].

The non-separable nature of the Hamiltonian in RMHMC means that the standard leapfrog method cannot be employed to simulate the resulting dynamic, with alternative symplectic integrators such as the *generalised leapfrog method* [48] required. These integrators involve implicit steps which requires solving a set of non-linear equations on each iteration. Further evaluation of the inverse of the metric and its log determinant in general have a cost which scales cubically with the D, therefore the overall computational cost of simulating the RMHMC dynamic is much higher per transition than for standard HMC. In some target distributions the gain in sampling efficiency over standard HMC from using RMHMC updates can signficantly outweigh the increased computational cost per sample however [7, 34].

2.3.3 Simulated tempering

The final auxiliary variable method we consider, *simulated tempering* [50], simulates the dynamics of a thermodynamic system subject to a varying temperature. Simulated tempering was originally proposed to improve the exploration of highly-multimodal distributions defined by undirected models such as the Ising spin model.

A particle of systems with a state described by a vector $\boldsymbol{x} \in X$ and a total energy determined by a function $\phi : X \to \mathbb{R}$ will have an equilibrium distribution on the state at a temperature T which has a density proportional to $\exp(-\beta\phi(\boldsymbol{x}))$ where $\beta = (kT)^{-1}$ is the *inverse temperature* and k is Boltzmann's constant. If the energy function ϕ is 'rough' with multiple local minima, then as the temperature T tends to zero and $\beta \to \infty$ the corresponding peaks in the density function become increasingly sharp and the mass of the distribution more tightly concentrated around these peaks. Conversely as the temperature T increases and $\beta \to 0$, the density becomes increasingly flat across X.

Simulated annealing [1, 45], is a stochastic optimisation method which uses this intutition about the properties of thermodynamical systems to improve the probability of an optimisation routine converging to a global optima in highly multimodal objectives. The objective function to be minimised is identified with the energy function ϕ of the system and the variables being optimised with the state $\boldsymbol{x} \in X$. An increasing *schedule* of K inverse temperatures $\{\beta_k\}_{k=1}^{K}$ is chosen with $0 \leq \beta_1 \leq \beta_2 \leq \cdots \leq \beta_K \leq \infty$. An initial value for the target variables $\boldsymbol{x}_0$ is (randomly) chosen and new values for the target variables are then computed iteratively for each $k \in \{1 \dots K\}$ by applying a Metropolis–Hastings transition operator $\boldsymbol{x}_k \sim \mathsf{T}_k(\cdot \,|\, \boldsymbol{x}_{k-1})$ which leaves the distribution with density proportional to $\exp(-\beta_k\phi(\boldsymbol{x}))$ invariant.

The hope is that the transitions at low inverse temperatures will be able to move freely around the target space due to the relatively flat form of the density function with lowered barriers between modes. Ideally

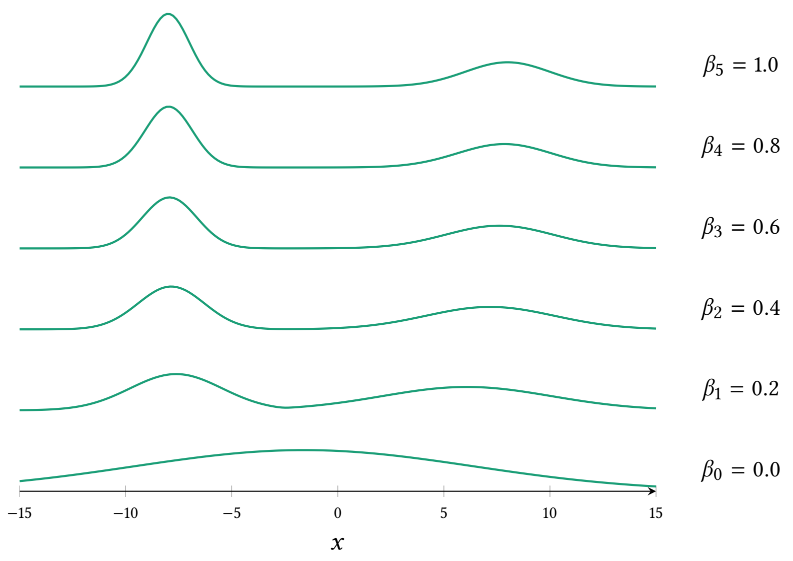


Figure 2.13: Example of using an inverse temperature to geometrically bridge between unimodal base and bimodal target densities. Each curve shows the conditional density $\mathsf{p}_{\mathsf{x}|\mathsf{k}}$ corresponding to the joint target density (2.65) for $\mathsf{k} = 0$ to $\mathsf{k} = 5$ with $\beta_k = k/5$ in this case.

the state will therefore tend to converge towards the modes with the largest mass. As the inverse temperature is increased the density function becomes increasingly peaked and the updates will tend to remain confined to one mode and as $\beta \to \infty$ will become concentrated near to the maximum of this mode. Although there is no guarantee this heuristic will find a global optima, empirically it has been found to be useful in practice in a range of applications.

In simulated tempering, rather than using an inverse temperature to define an optimisation procedure instead a discrete index controlling the inverse temperature is introduced as an auxiliary variable in an MCMC method. The variables $\mathbf{x} \in X$ on which the target distribution P which we wish to estimate integrals with respect to are augmented with a discrete index variable $\mathsf{k} \in \{0 \dots K\}$. A corresponding set of inverse temperature values $\{\beta_k\}_{k=0}^{K}$ are specified, as with simulated annealing these chosen to form an ordered sequence but in this case over the interval $[0, 1]$ with $0 = \beta_0 < \beta_1 < \beta_2 < \cdots < \beta_K = 1$. A joint density on the target variables $\mathbf{x}$ and temperature index k is then defined as

$$\mathsf{p}_{\mathbf{x},\mathsf{k}}(\boldsymbol{x}, k) = \frac{1}{C} \exp(-\beta_k \phi(\boldsymbol{x}) - (1 - \beta_k)\psi(\boldsymbol{x}) + w_k). \tag{2.65}$$

Algorithm 7 Simulated tempering transition.

Input: $(\boldsymbol{x}_n, k_n)$: current target variables – temperature index state pair, T_1 : transition operator updating only target variables $\mathbf{x}$ and leaving distribution with density in (2.65) invariant, T_2 : transition operator updating only temperature index k and leaving distribution with density in (2.65) invariant.
Output: $(\boldsymbol{x}_{n+1}, k_{n+1})$: new target variables – temperature index state pair.

```
1: $\boldsymbol{x}_{n+1} \sim \mathsf{T}_1(\cdot \mid \boldsymbol{x}_n,\, k_n)$
2: $k_{n+1} \sim \mathsf{T}_2(\cdot \mid \boldsymbol{x}_{n+1},\, k_n)$
3: **return** $(\boldsymbol{x}_{n+1},\, k_{n+1})$
```

The values $\{w_k\}_{k=0}^{K}$ are a set of 'prior' weights associated with each inverse temperature value, and which can be used to help shape the marginal distribution on the temperature index k. As in the preceding subsection the energy function $\phi : X \to \mathbb{R}$ is defined as the negative logarithm of the unnormalised target density i.e. $\phi(\boldsymbol{x}) = -\log \tilde{p}(\boldsymbol{x})$.

The function $\psi : X \to \mathbb{R}$ defines a corresponding energy function for a *base distribution* Q with normalised density $q(\boldsymbol{x}) = \exp(-\psi(\boldsymbol{x}))$ with respect to μ. The base distribution is typically chosen to have a simple unimodal density with mass covering as many of the regions of high density under the target density in X as possible. When the target distribution corresponds to the posterior in a Bayesian inference task, Q is often chosen as the prior distribution on the target variables which will typically have a simple unimodal form and be much more diffuse than the posterior. If the state space X consists of a finite set of values, the base distribution can be chosen to be uniform across X in which case $\psi(\boldsymbol{x})$ is constant and can be omitted from (2.65).

Importantly the conditional distribution $\mathsf{P}_{\mathbf{x}|\mathsf{k}}$ on the target variables $\mathbf{x}$ for $\mathsf{k} = 0$ ($\beta_0 = 0$) corresponds to the base distribution Q and to the target distribution P for $\mathsf{k} = K$ ($\beta_K = 1$). We can therefore use the $\mathbf{x}$ components of sampled chain states for which $\mathsf{k} = K$ to estimate expectations with respect to the target distribution P. For intermediate values of k the conditional distribution geometrically interpolates between P and Q. Figure 2.13 shows a simple example of this geometric bridging between a unimodal univariate base density and bimodal target density for $K = 5$ inverse temperature values.

In simulated tempering, a Markov chain with an invariant distribution corresponding to (2.65) is constructed by alternating updates of the target variables $\mathbf{x}$ given the current value of temperature index k, with updates of the temperature index k given the current value of the target variables $\mathbf{x}$ as summarised in Algorithm 7. For the transition operator T_1 updating the target variables, any of the previously discussed methods such as random-walk Metropolis, slice sampling or HMC can be used. In the case of Metropolis–Hastings based updates, it may be desirable to adjust the proposal generating mechanism to depend on the current temperature index k as for example we might generally expect for k corresponding to lower inverse temperatures β_k and so conditional densities $\mathsf{p}_{\mathbf{x}|\mathsf{k}}$ closer to the base density that larger moves can be made while maintaining reasonable accept rates; as k remains fixed this can validly be done without breaking reversibility.

In the original description of the simulated tempering algorithm in [50], the transitions to the index variable k given fixed values of the target variables $\mathbf{x}$ were performed using a random-walk Metropolis operator for T_2 which proposes to randomly increment or decrement k by one (except at the end-points $\mathsf{k} = 0$ and $\mathsf{k} = K$ where it always proposed to increment and decrement respectively). For large K this can lead to slow mixing up and down the inverse temperature scale - if the marginal density p_k is uniform we would expect $\mathcal{O}(K^2)$ updates would be needed to traverse the full inverse temperature range. An alternative is to use a Gibbs sampling step with the conditional distribution $\mathsf{P}_{\mathsf{k}|\mathbf{x}}$ here being a multinomial distribution with density

$$\mathsf{p}_{\mathsf{k}|\mathbf{x}}(k \mid \boldsymbol{x}) = \frac{\exp(\beta_k(\psi(\boldsymbol{x}) - \phi(\boldsymbol{x})) + w_k)}{\sum_{k'=0}^{K} \exp(\beta_{k'}(\psi(\boldsymbol{x}) - \phi(\boldsymbol{x})) + w_{k'})} \tag{2.66}$$

which we can tractably generate independent samples from. For arbitrary $\{\beta_k, w_k\}_{k=0}^{K}$ this will require explicit summation over $K+1$ values to calculate the normalising constant and so the cost of generating an independent index will scale linearly with K.

For $\beta_k = \frac{k}{K}$ and $w_k = \alpha\beta_k\ \forall k \in \{0 \dots K\}$ for some $\alpha \in \mathbb{R}$, the normalising constant in (2.66) takes the form of a geometric series

$$\sum_{k=0}^{K} \exp\left(\frac{\psi(\boldsymbol{x}) - \phi(\boldsymbol{x}) + \alpha}{K}\right)^{k} = 1 + \frac{\exp(\psi(\boldsymbol{x}) - \phi(\boldsymbol{x}) + \alpha)}{1 - \exp\left(\frac{\psi(\boldsymbol{x}) - \phi(\boldsymbol{x}) + \alpha}{K}\right)}. \tag{2.67}$$

The conditional distribution $\mathsf{P}_{\mathsf{k}|\mathbf{x}}$ in this case has the form of a geometric distribution with parameter $\exp\left(\frac{\psi(\boldsymbol{x}) - \phi(\boldsymbol{x}) + \alpha}{K}\right)$ truncated to $\{0 \dots K\}$ which we can generate samples at a cost independent of K.

The marginal distribution P_k on the index variable k has density

$$\mathsf{p}_\mathsf{k}(k) = \frac{\exp(w_k)}{C} \int_X \exp(-\beta_k \phi(\boldsymbol{x}) - (1-\beta_k)\psi(\boldsymbol{x}))\, \mu(\mathrm{d}\boldsymbol{x}). \tag{2.68}$$

As $\int_X \exp(-\psi(\boldsymbol{x}))\, \mu(\mathrm{d}\boldsymbol{x}) = 1$ and $\int_X \exp(-\phi(\boldsymbol{x}))\, \mu(\mathrm{d}\boldsymbol{x}) = Z$ we have

$$\mathsf{p}_\mathsf{k}(0) = \frac{\exp(w_0)}{C} \qquad \text{and} \qquad \mathsf{p}_\mathsf{k}(K) = \frac{\exp(w_K)Z}{C}. \tag{2.69}$$

If Z is much more than one and $w_k = 0$ for all $k \in \{0 \dots K\}$ then we would have $\mathsf{p}_\mathsf{k}(K) \gg \mathsf{p}_\mathsf{k}(0)$ and a simulated tempering chain will tend to spend many more iterations with $\mathsf{k} = K$ than $\mathsf{k} = 0$. This will give a large number of samples with which to estimate expectations with respect to P however it will also limit the gain from using simulated tempering over running a Markov chain in the original non-augmented target variable space, as the chain will rarely visit the lower inverse temperatures which aid exploration. Conversely if Z is much less than one, we have $\mathsf{p}_\mathsf{k}(K) \ll \mathsf{p}_\mathsf{k}(0)$. In this case the chain will tend to remain at k values corresponding to low inverse temperatures and so few samples are available for computing expectations with respect to P.

If we could set $w_0 - w_K = \log Z$ we would have $\mathsf{p}_\mathsf{k}(K) = \mathsf{p}_\mathsf{k}(0)$ however for the target distributions of interest we will generally not be able to evaluate Z and a similar result holds for the normalising constants of the conditional distributions $\mathsf{P}_{\mathbf{x}|\mathsf{k}}$ corresponding to intermediate inverse temperatures and so the appropriate values for $\{w_k\}_{k=1}^{K-1}$. In general therefore it will be difficult to identify reasonable values to set the weights $\{w_k\}_{k=0}^{K}$ to a-priori. This is typically solved in practice by using an iterative scheme [42, 50]: an initial pilot chain is run with $w_k = 0\ \forall k$ to estimate the marginal density p_k by constructing a histogram of counts of samples for each k and then this histogram used to set the weights so as to approximately flatten the marginal density.

The relationship between the marginal density p_k and Z although presenting challenges in terms of choosing the weights $\{w_k\}_{k=0}^{K}$ also however demonstrates that simulated tempering chains can be used to estimate Z. In particular we have that

$$Z = \exp(w_0 - w_K)\frac{\mathsf{p}_\mathsf{k}(K)}{\mathsf{p}_\mathsf{k}(0)}. \tag{2.70}$$

Given the sampled states $\{\boldsymbol{x}^{(n)}, k^{(n)}\}_{n=1}^{N}$ of a simulated tempering chain, one way to form a consistent estimate of Z is therefore to compute the ratio of the counts of samples with $\mathsf{k} = K$ to those with $\mathsf{k} = 0$,

$$Z = \lim_{N\to\infty} \exp(w_0 - w_K) \frac{\sum_{n=1}^{N} \mathbb{1}_{\{K\}}(k^{(n)})}{\sum_{n=1}^{N} \mathbb{1}_{\{0\}}(k^{(n)})}. \tag{2.71}$$

This estimate will typically have a high variance however as it uses information only from the subset of sampled states with $\mathsf{k} = 0$ or $\mathsf{k} = K$. Expanding p_{k} as a marginalisation integral of the joint density (2.65) we can reformulate the identity in (2.70) as

$$Z = \exp(w_0 - w_K) \frac{\int_X \mathsf{p}_{\mathsf{k}|\mathbf{x}}(K \mid \boldsymbol{x})\, \mathsf{p}_{\mathbf{x}}(\boldsymbol{x})\, \mu(\mathrm{d}\boldsymbol{x})}{\int_X \mathsf{p}_{\mathsf{k}|\mathbf{x}}(0 \mid \boldsymbol{x})\, \mathsf{p}_{\mathbf{x}}(\boldsymbol{x})\, \mu(\mathrm{d}\boldsymbol{x})}. \tag{2.72}$$

This is an example of what is sometimes termed *Rao-Blackwellisation* [20] and was used in [17] to suggest a *Rao-Blackwellised estimator* for the normalising constant Z from the samples $\{\boldsymbol{x}^{(s)}, k^{(s)}\}_{s=1}^{S}$ of a simulated tempering chain

$$Z = \lim_{N\to\infty} \exp(w_0 - w_K) \frac{\sum_{n=1}^{N} \mathsf{p}_{\mathsf{k}|\mathbf{x}}(K \mid \boldsymbol{x}^{(n)})}{\sum_{n=1}^{N} \mathsf{p}_{\mathsf{k}|\mathbf{x}}(0 \mid \boldsymbol{x}^{(n)})}. \tag{2.73}$$

This estimator uses all of the sampled chain states and will typically be lower variance than the count-based estimator (2.71). Importantly this estimator can still give reasonable estimates for Z when there are no sampled states for which $\mathsf{k} = 0$ (or $\mathsf{k} = K$) unlike the count-based estimator. This is particularly important when using an iterative scheme to choose the weights $\{w_k\}_{k=0}^{K}$ as if $Z \gg 1$ or $Z \ll 1$ an initial short pilot chain will typically remain confined to one end of the inverse temperature scale for all iterations, giving limited count-based information with which to update weights for subsequent iterations.

2.4 DISCUSSION

The sampling approaches to approximate inference described in this chapter allow tractable estimation of the integrals involved in many inference problems. In cases where we can generate independent samples from the target distribution, the $\frac{1}{N}$ scaling of the variance of Monte Carlo estimates of expectations with the number of samples N allows computation of estimates with sufficient accuracy for most practical purposes without the exponential blow-up in computation of quadrature methods with dimensionality.

Generating independent samples from arbitrary distributions on high-dimensional spaces is often infeasible however. Transform sampling methods offer a scalable approach for only a few special cases such as the multivariate normal distribution. Rejection sampling is more generally applicable however the usually exponential decrease in the proportion of accepted samples with dimension means that it is only useful in relatively low-dimensionsal distributions. Simple importance sampling schemes similarly scale poorly with dimensionality, with mismatch between the proposal and target distribution in high-dimensions meaning the variance of the resulting estimators is impractically high.

Although these Monte Carlo methods are not directly applicable to performing inference in the complex probabilistic models of interest, they are still useful building blocks and will appear as components of the methods we will discuss in the rest of this thesis. In Chapter 3 we will discuss MCMC methods which use importance sampling estimators of the target density to construct the chain. The simulator models discussed in Chapter 4 can be considered an extension of the idea of transform sampling, with a complex series of deterministic operations transforming inputs from a pseudo-random number generator to simulated values for the variables in a probabilistic model. One of the standard approaches for performing approximate inference in simulator models is based on rejection sampling, and our discussion of the poor scaling of rejection sampling with dimensionality will be relevant when considering the limitations of these methods.

Markov chain Monte Carlo methods offer a more scalable approach to inference in complex probabilistic models and are the main focus of the work discussed in this thesis. The local pertubative updates typically employed in MCMC methods avoid the curse of dimensionality effects which lead to the exponential blow up in the computational effort required by methods such as rejection sampling as the dimension increases. MCMC methods such as random-walk Metropolis and Gibbs sampling typically require minimal implementation effort and have successfully applied in a wide range of settings.

For target distributions with more complex geometries however such as due to the non-linear relationships between variables often present in hierarchical models or the multimodal distributions typically arising from inference in undirected models, MCMC methods such as random-walk Metropolis and Gibbs sampling can exhibit pathological behaviour that means impractically long chains are needed for MCMC estimators to give useful results. In these cases methods which exploit more information about the geometry of the distribution in each update can offer significant improvements in efficiency and robustness.

The introduction of auxiliary variables in to the chain state has proved a particularly successful approach for proposing MCMC methods which can accelerate the exploration of complex target distributions. We conlcuded this chapter by reviewing three auxiliary variable MCMC methods that will be central to the contributions made in this thesis: slice sampling, Hamiltonian Monte Carlo and simulated tempering.

Slice-sampling offers a very generally applicable approach for constructing Markov chains which are able to adapt the scale of proposed moves to the local geometry of the target distribution. The information controlling this adaptation comes from allowing multiple evaluations of the target density per update in slice sampling compared to for example the single target density evaluation per iteration of random-walk Metropolis methods. The overhead from these multiple density evaluations will mean that for target distribution in which the geometry of the density does not vary signficantly across the space, well-tuned random-walk Metropolis updates will often be able to outperform slice sampling transition operators in terms of the computational cost per effective independent sample. However the ease of use of slice sampling methods, with typically minimal user tuning required of the free algorithmic parameters, and increased robustness to distributions with more complex geometries, are in our opinion often more important than a potential improvement in peak efficiency.

Hamiltonian Monte Carlo methods put a requirement of differentiability on the target density and so are not as widely applicable as slice sampling approaches. When available however gradient information can be a significant help in guiding the exploration of the target space by a MCMC dynamic. Using reverse-mode automatic differentiation (as described in Appendix B) code for evaluating the exact gradients of a density function can be automatically generated given just the definition of the original density function and the resulting gradient function evaluated at a cost which has only a constant factor overhead over the cost of the original density function evaluations. When optimally tuned HMC methods can overcome the random-walk behaviour inherent to simpler MCMC methods and so offer significantly improved performance in complex high-dimensional target distributions. Although implementation of HMC algorithms is more complex than approaches such as Gibbs sampling and random-walk Metropolis and the tuning of the algorithm parameters can be vital for good performance, the availability of efficient, adaptive implementations in probabilistic programming frameworks such as Stan [19] and PyMC3 [75] has supported the use of HMC in a wide range of inference problems.

Simulated tempering offers a complementary approach to the improved local exploration afforded by slice sampling and HMC methods by potentially improving the global exploration of challenging multimodal target distributions. As the updates to the target variables at a fixed inverse temperature can be performed using any valid Markov transition operator applicable to the original target distribution, both slice sampling and HMC transition operators can be used within a simulated tempering chain and both potentially offer an improved ability to adapt to the varying geometry of the density on the target variables at different inverse temperatures compared to simpler methods such as random-walk Metropolis. In addition to the possible improved exploration of multimodal targets, the ability to use simulated tempering chains to estimate an unknown normalising constant of the target density, often corresponding to a model evidence term, offers a further distinct advantage over standard MCMC methods.

Although simulated tempering can provide several important benefits, use of the algorithm in statistical applications seems relatively rare in practice. This can perhaps be partially attributed to the need to tune the values of the free inverse temperature β_k and prior weight w_k parameters introduced in the algorithm, with any improvement in exploration of the target distribution strongly dependent on the simulated tempering chain being able to move up and down the inverse temperature range. Further the lack of standard implementations in frameworks such as Stan and PyMC3, and relative wastefulness of the standard approach of estimating expectations with respect to the target distribution by averaging over only sampled states corresponding to an inverse temperature of $\beta_K = 1$, add further discouragements to widespread use of the algorithm.

2.5 OUTLINE OF CONTRIBUTIONS

Having now completed our reviews of both the inference tasks and MCMC methods underlying the work in this thesis, we are now in a position to outline the contributions made in the rest of the thesis.

Inference in hierarchical models is often a challenging task for MCMC methods due to the strong dependencies between the global and local latent variables and resulting complex geometry of the target density on the model variables. We will sometimes only be directly interested in inferring plausible values for the global latent variables in the model but will typically be unable to analytically integrate out the local latent variables. The *pseudo-marginal* framework shows how an unbiased estimator of the marginal density on the global latent variables can be used to construct a Metropolis–Hastings method for sampling values of the global latent variables. Pseudo-marginal Metropolis–Hastings methods however suffer from 'sticking' pathologies where chains reject updates over long series of iterations and are challenging to tune.

In Chapter 3 we demonstrate that by including the auxiliary variables used in the density estimator in the chain state alternative transition operators can be used in a pseudo-marginal setting, including adaptive rejection-free methods like slice-sampling. The resulting *auxiliary pseudo-marginal* methods are able to prevent the sticking artifacts common to existing pseudo-marginal methods, are easier to tune and in some cases give significant improvements in sampling efficiency.

We described simulator models as a challenging setting for approximate inference methods in Chapter 1 due to the lack of an explicit target density on the model variables. *Approximate Bayesian computation* (ABC) is a class of methods for performing inference in such models by conditioning on simulated observations being 'close' rather than exactly equal to the observed data. ABC methods based on both rejection sampling and pseudo-marginal Metropolis–Hastings have been proposed, but both suffer from curse of dimensionality effects that mean further approximation is typically required by reducing the simulated observations and data to lower-dimensional summary statistics.

In Chapter 4 we show that any generative model can be considered as a deterministic transformation of a vector of auxiliary variables from a known distribution. We use this intution to demonstrate how MCMC methods such as slice sampling and HMC can be applied within an ABC setting, the improved performance of these methods compared to approaches based on pseudo-marginal Metropolis–Hastings meaning that in some cases ABC inference can be peformed without the need for summary statistics. For a restricted class of *differentiable generative models* we derive an expression for conditional expectations under the model in terms of an integral against a distribution defined on an implicitly-defined manifold. We use this to propose a novel constrained HMC method for performing approximate inference in differentiable generative models without an explicit density function on the model variables. This method allows computationally tractable inference when conditioning high-dimensional simulated observations being arbitarily close to observed data.

Simulated tempering provides an approach for tackling two of the key challenges identified in Chapter 1: performing inference in multimodal distributions such as those defined by undirected models like the Boltzmann machine; estimating the model evidence normalising constant terms required for model comparison. However as noted above simulated tempering is used relatively rarely in practice. In the above discussion we suggested factors which may have discouraged more widespread adoption of the algorithm: the difficulty in choosing the set of inverse temperature and prior weight values to use, the relative inefficiency of using only a small proportion of the samples in a chain to compute estimates and the lack of support for simulated tempering methods in existing inference packages.

In Chapter 5 we suggest approaches to overcome these issues. We propose using a continuous auxiliary variable to control the inverse temperature rather than a discrete index. This sidesteps the need to choose a set of inverse temperature values and allows the auxiliary variable to be jointly updated with the target variables in a HMC update making it straightforward to use tempering within existing HMC-based inference packages. Further we show how all of the samples in a tempered chain can be used to estimate expectations with respect to the target distribution. Finally we demonstrate that variational inference methods provide a natural approach for choosing the base distribution bridged to during tempering and show that cheap biased approximations to the normalising constant of the target density can be exploited to help flatten the marginal density on the inverse temperature.

BIBLIOGRAPHY

[1] David H Ackley, Geoffrey E Hinton and Terrence J Sejnowski. ‘A learning algorithm for Boltzmann machines’. In: *Cognitive science* 9.1 (1985), pp. 147–169.

[2] Shun-Ichi Amari. ‘Differential geometry of curved exponential families-curvatures and information loss’. In: *The Annals of Statistics* (1982), pp. 357–385.

[3] Christophe Andrieu and Johannes Thoms. ‘A tutorial on adaptive MCMC’. In: *Statistics and computing* 18.4 (2008), pp. 343–373.

[4] IEEE Standards Association. ‘Standard for Floating-Point Arithmetic’. In: *IEEE 754-2008* (2008).

[5] Alessandro Barp, Francois-Xavier Briol, Anthony D Kennedy and Mark Girolami. ‘Geometry and Dynamics for Markov Chain Monte Carlo’. In: *arXiv preprint arXiv:1705.02891* (2017).

[6] Alexandros Beskos, Natesh Pillai, Gareth Roberts, Jesus-Maria Sanz-Serna and Andrew Stuart. ‘Optimal tuning of the hybrid Monte Carlo algorithm’. In: *Bernoulli* 19.5A (2013), pp. 1501–1534.

[7] Michael Betancourt. ‘A general metric for Riemannian manifold Hamiltonian Monte Carlo’. In: *Geometric science of information*. Springer, 2013.

[8] Michael Betancourt. ‘Generalizing the no-U-turn sampler to Riemannian manifolds’. In: *arXiv preprint arXiv:1304.1920* (2013).

[9] Michael Betancourt. ‘Identifying the Optimal Integration Time in Hamiltonian Monte Carlo’. In: *arXiv preprint arXiv:1601.00225* (2016).

[10] Michael Betancourt. ‘A Conceptual Introduction to Hamiltonian Monte Carlo’. In: *arXiv preprint arXiv:1701.02434* (2017).

[11] Michael Betancourt. ‘The Convergence of Markov chain Monte Carlo Methods: From the Metropolis method to Hamiltonian Monte Carlo’. In: *arXiv preprint arXiv:1706.01520* (2017).

[12] Michael Betancourt, Simon Byrne and Mark Girolami. ‘Optimizing the integrator step size for Hamiltonian Monte Carlo’. In: *arXiv preprint arXiv:1411.6669* (2014).

[13] Michael Betancourt, Simon Byrne, Sam Livingstone and Mark Girolami. ‘The geometric foundations of Hamiltonian Monte Carlo’. In: *Bernoulli* 23.4A (2017), pp. 2257–2298.

[14] Joris Bierkens. ‘Non-reversible Metropolis–Hastings’. In: *Statistics and Computing* 26.6 (2016), pp. 1213–1228.

[15] George D Birkhoff. ‘Proof of the ergodic theorem’. In: *Proceedings of the National Academy of Sciences* 17.12 (1931), pp. 656–660.

[16] George EP Box and Mervin E Muller. ‘A note on the generation of random normal deviates’. In: *The Annals of Mathematical Statistics* 29.2 (1958), pp. 610–611.

[17] David Carlson, Patrick Stinson, Ari Pakman and Liam Paninski. ‘Partition Functions from Rao-Blackwellized Tempered Sampling’. In: *Proceedings of The 33rd International Conference on Machine Learning*. 2016.

[18] Bob Carpenter, Matthew D Hoffman, Marcus Brubaker, Daniel Lee, Peter Li and Michael Betancourt. ‘The Stan Math Library: Reverse-Mode Automatic Differentiation in C++’. In: *arXiv preprint arXiv:1509.07164* (2015).

[19] Bob Carpenter, Andrew Gelman, Matt Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Michael A Brubaker, Jiqiang Guo, Peter Li and Allen Riddell. ‘Stan: A probabilistic programming language’. In: *Journal of Statistical Software* (2016).

[20] George Casella and Christian P. Robert. ‘Rao-Blackwellisation of sampling schemes’. In: *Biometrika* 83.1 (1996), pp. 81–94. DOI: 10.1093/biomet/83.1.81. URL: +http://dx.doi.org/10.1093/biomet/83.1.81.

[21] KS Chan. ‘Asymptotic behavior of the Gibbs sampler’. In: *Journal of the American Statistical Association* 88.421 (1993), pp. 320–326.

[22] Kung Sik Chan and Charles J Geyer. ‘Discussion: Markov chains for exploring posterior distributions’. In: *The Annals of Statistics* 22.4 (1994), pp. 1747–1758.

[23] Ming-Hui Chen and Bruce Schmeiser. 'Toward black-box sampling: A random-direction interior-point Markov chain approach'. In: *Journal of Computational and Graphical Statistics* 7.1 (1998), pp. 1–22.

[24] P Damien, John Wakefield and Stephen Walker. 'Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables'. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.2 (1999), pp. 331–344.

[25] Persi Diaconis, Susan Holmes and Radford M Neal. 'Analysis of a nonreversible Markov chain sampler'. In: *Annals of Applied Probability* (2000), pp. 726–752.

[26] Simon Duane, Anthony D Kennedy, Brian J Pendleton and Duncan Roweth. 'Hybrid Monte Carlo'. In: *Physics Letters B* (1987).

[27] Alan E Gelfand and Adrian FM Smith. 'Sampling-based approaches to calculating marginal densities'. In: *Journal of the American Statistical Association* (1990).

[28] Andrew Gelman, Walter R Gilks and Gareth O Roberts. 'Weak convergence and optimal scaling of random walk Metropolis algorithms'. In: *The annals of applied probability* 7.1 (1997), pp. 110–120.

[29] Stuart Geman and Donald Geman. 'Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images'. In: *IEEE Transactions on pattern analysis and machine intelligence* (1984).

[30] Charles J Geyer. *Markov chain Monte Carlo lecture notes (Statistics 8931)*. Tech. rep. University of Minnesota, 1998.

[31] Charles J Geyer. 'The Metropolis-Hastings-Green Algorithm'. 2003. URL: http://www.stat.umn.edu/geyer/f05/8931/bmhg.pdf.

[32] Wally R Gilks, Andrew Thomas and David J Spiegelhalter. 'A language and program for complex Bayesian modelling'. In: *The Statistician* (1994), pp. 169–177.

[33] Walter R Gilks and Pascal Wild. 'Adaptive rejection sampling for Gibbs sampling'. In: *Applied Statistics* (1992), pp. 337–348.

[34] Mark Girolami and Ben Calderhead. 'Riemann-manifold Langevin and Hamiltonian Monte Carlo methods'. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73.2 (2011), pp. 123–214.

[35] Peter J Green. ‘Reversible jump Markov chain Monte Carlo computation and Bayesian model determination’. In: *Biometrika* (1995), pp. 711–732.

[36] J. E. Gubernatis. ‘Marshall Rosenbluth and the Metropolis algorithm’. In: *Physics of Plasmas* 12.5 (2005), p. 057303. URL: http://dx.doi.org/10.1063/1.1887186.

[37] Heikki Haario, Eero Saksman and Johanna Tamminen. ‘An adaptive Metropolis algorithm’. In: *Bernoulli* (2001), pp. 223–242.

[38] Theodore E Harris. ‘The existence of stationary measures for certain Markov processes’. In: *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*. Vol. 2. 1956, pp. 113–124.

[39] W Keith Hastings. ‘Monte Carlo sampling methods using Markov chains and their applications’. In: *Biometrika* (1970).

[40] Bryan D He, Christopher M De Sa, Ioannis Mitliagkas and Christopher Ré. ‘Scan Order in Gibbs Sampling: Models in Which it Matters and Bounds on How Much’. In: *Advances in Neural Information Processing Systems*. 2016, pp. 1–9.

[41] Matthew D Hoffman and Andrew Gelman. ‘The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo.’ In: *Journal of Machine Learning Research* (2014).

[42] Yukito Iba. ‘Extended ensemble Monte Carlo’. In: *International Journal of Modern Physics C* 12.05 (2001), pp. 623–656.

[43] Akihisa Ichiki and Masayuki Ohzeki. ‘Violation of detailed balance accelerates relaxation’. In: *Physical Review E* 88.2 (2013), p. 020101.

[44] Herman Kahn and Theodore E Harris. ‘Estimation of particle transmission by random sampling’. In: *National Bureau of Standards applied mathematics series* 12 (1951), pp. 27–30.

[45] Scott Kirkpatrick, C Daniel Gelatt and Mario P Vecchi. ‘Optimization by simulated annealing’. In: *Science* (1983).

[46] Dirk P. Kroese, Thomas Taimre and Zdravko I. Botev. ‘Variance Reduction’. In: *Handbook of Monte Carlo Methods*. John Wiley & Sons, Inc., 2011, pp. 347–380. ISBN: 9781118014967. DOI: 10.1002/9781118014967.ch9. URL: http://dx.doi.org/10.1002/9781118014967.ch9.

[47] Derrick H Lehmer. ‘Mathematical methods in large-scale computing units’. In: *Proceedings of a Second Symposium on Large-Scale Digital Calculating Machinery (1949)*. 1951, pp. 141–146.

[48] Benedict Leimkuhler and Sebastian Reich. *Simulating Hamiltonian dynamics*. Cambridge University Press, 2004.

[49] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

[50] Enzo Marinari and Giorgio Parisi. ‘Simulated tempering: a new Monte Carlo scheme’. In: *Europhysics Letters* (1992).

[51] George Marsaglia. ‘Random numbers fall mainly in the planes’. In: *Proceedings of the National Academy of Sciences* 61.1 (1968), pp. 25–28.

[52] Makoto Matsumoto and Takuji Nishimura. ‘Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator’. In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 8.1 (1998), pp. 3–30.

[53] Kerrie L Mengersen and Richard L Tweedie. ‘Rates of convergence of the Hastings and Metropolis algorithms’. In: *The annals of Statistics* 24.1 (1996), pp. 101–121.

[54] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller and Edward Teller. ‘Equation of state calculations by fast computing machines’. In: *The Journal of Chemical Physics* (1953).

[55] Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 1993.

[56] Iain Murray, Ryan Prescott Adams and David J.C. MacKay. ‘Elliptical slice sampling’. In: JMLR: W&CP 9 (2010), pp. 541–548.

[57] Radford M Neal. *Markov chain Monte Carlo methods based on ‘slicing’ the density function*. Tech. rep. 9722. Department of Statistics, University of Toronto, 1997.

[58] Radford M Neal. ‘Slice sampling’. In: *Annals of statistics* (2003).

[59] Radford M Neal. ‘Improving asymptotic variance of MCMC estimators: Non-reversible chains are better’. In: *arXiv preprint math/0407281* (2004).

[60] Radford M Neal. ‘MCMC using Hamiltonian dynamics’. In: *Handbook of Markov Chain Monte Carlo*. Chapman & Hall/CRC, 2011. Chap. 5, pp. 113–162.

[61] John von Neumann. ‘Various techniques used in connection with random digits’. In: *National Bureau of Standards applied mathematics series* 3 (1951), pp. 36–38.

[62] Robert Nishihara, Iain Murray and Ryan P Adams. ‘Parallel MCMC with generalized elliptical slice sampling.’ In: *Journal of Machine Learning Research* 15.1 (2014), pp. 2087–2112.

[63] Akihiko Nishimura and David Dunson. ‘Geometrically Tempered Hamiltonian Monte Carlo’. In: *arXiv preprint arXiv:1604.00872* (2016).

[64] Sheehan Olver and Alex Townsend. ‘Fast inverse transform sampling in one and two dimensions’. In: *arXiv preprint arXiv:1307.1223* (2013).

[65] Art B. Owen. ‘Importance sampling’. In: *Monte Carlo theory, methods and examples*. 2013. URL: http://statweb.stanford.edu/~owen/mc/Ch-var-is.pdf.

[66] Peter H Peskun. ‘Optimum Monte-Carlo sampling using Markov chains’. In: *Biometrika* 60.3 (1973), pp. 607–612.

[67] Martyn Plummer. ‘JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling’. In: *Proceedings of the 3rd international workshop on distributed statistical computing*. Vol. 124. Vienna. 2003, p. 125.

[68] Martyn Plummer, Nicky Best, Kate Cowles and Karen Vines. ‘CODA: Convergence Diagnosis and Output Analysis for MCMC’. In: *R News* 6.1 (2006), pp. 7–11. URL: http://CRAN.R-project.org/doc/Rnews/Rnews_2006-1.pdf.

[69] Adrian E Raftery and Steven Lewis. *How many iterations in the Gibbs sampler?* Tech. rep. Washington University, Seattle, Department of Statistics, 1991.

[70] Gareth O Roberts and Jeffrey S Rosenthal. ‘Optimal scaling for various Metropolis-Hastings algorithms’. In: *Statistical science* 16.4 (2001), pp. 351–367.

[71] Gareth O Roberts and Jeffrey S Rosenthal. ‘General state space Markov chains and MCMC algorithms’. In: *Probability Surveys* 1 (2004), pp. 20–71.

[72] Gareth O Roberts and Adrian FM Smith. 'Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms'. In: *Stochastic processes and their applications* 49.2 (1994), pp. 207–216.

[73] Gareth O Roberts and Richard L Tweedie. 'Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms'. In: *Biometrika* (1996), pp. 95–110.

[74] Jeffrey S Rosenthal. 'Optimal proposal distributions and adaptive MCMC'. In: *Handbook of Markov Chain Monte Carlo* (2011), pp. 93–112.

[75] John Salvatier, Thomas V Wiecki and Christopher Fonnesbeck. 'Probabilistic programming in Python using PyMC3'. In: *PeerJ Computer Science* (2016).

[76] Yi Sun, Jürgen Schmidhuber and Faustino J Gomez. 'Improving the asymptotic performance of Markov chain Monte-Carlo by inserting vortices'. In: *Advances in Neural Information Processing Systems*. 2010, pp. 2235–2243.

[77] Hidemaro Suwa and Synge Todo. 'Markov chain Monte Carlo method without detailed balance'. In: *Physical review letters* 105.12 (2010), p. 120603.

[78] Stan Development Team. *Stan Modeling Language Users Guide and Reference Manual*. Version 2.16.0. URL: http://mc-stan.org.

[79] Madeleine B Thompson. 'A comparison of methods for computing autocorrelation time'. In: *arXiv preprint arXiv:1011.0175* (2010).

[80] Luke Tierney. 'Markov chains for exploring posterior distributions'. In: *The Annals of Statistics* (1994), pp. 1701–1728.

[81] Konstantin S Turitsyn, Michael Chertkov and Marija Vucelja. 'Irreversible Monte Carlo algorithms for efficient sampling'. In: *Physica D: Nonlinear Phenomena* 240.4 (2011), pp. 410–414.

[82] Stanislaw Ulam and Nicholas Metropolis. 'The Monte Carlo method'. In: *Journal of the American Statistical Association* 44.247 (1949), pp. 335–341.