Larissa STOISER

Markov Chain Quasi Monte Carlo

MASTERARBEIT

zur Erlangung des akademischen Grades einer Diplom-Ingenieurin

Mathematische Computerwissenschaften

Technische Universität Graz

Betreuer:
Univ.-Prof. Dipl.-Ing. Dr.techn. Wolfgang WOESS

Institut für Mathematische Strukturtheorie

Graz, Jänner 2012

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am ....
(Unterschrift)

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitely marked all material which has been quotes either literally or by content from the used sources.

....
date (signature)

Abstract

This master thesis is about comparing the two main categories of Monte Carlo methods for simulating random processes and to find combinations of the two methods. At first I did research into the Markov chain Monte Carlo (MCMC) techniques. Their task is the approximation of distributions on large but finite sets with the help of Markov chains, which have the corresponding distribution as invariant measure. The construction of a Markov chain with the desired properties is usually not that hard. The more tricky part is to determine how many steps of the Markov chain are needed to converge to the stationary distribution within an acceptable error. Thereafter I studied the quasi Monte Carlo (QMC) method. The theory behind QMC is the uniform distribution modulo 1 and deals with the behavior of distributions of real number sequences in the interval $[0, 1]$. In conclusion I present versions of MCMC algorithms using quasi Monte Carlo inputs.

Contents

1 Introduction

The object of investigation of this master thesis are computer based simulation algorithms for the evaluation of the statistical behavior of objects or processes of scientific interest. In this context the term Monte Carlo (MC) simulation arises, which summarizes a huge variety of different simulation algorithms. The main ingredients for Monte Carlo simulation are independent and uniformly distributed random variables on the unit interval $[0, 1]$. They build the basis for this kind of algorithms. Since we won't have access to truly random numbers we provide a remedy through random number generators. These random number generators try to imitate the properties of real random sequences, see Knuth (1981). Based on the fact that there have been huge advances in random number generation in the last decades, the assumption that computers can generate such true independent and uniformly distributed random variables won't cause any serious failures. The random variables simulated with standard random number generators are called pseudo-random numbers.

A subclass of the Monte Carlo algorithms is formed by quasi-Monte Carlo (QMC) simulations. Their basis is built on a deterministic version of random or pseudo-random sequences. These sequences are called quasi-random. The aim of quasi-random sequences is to provide better uniformity on the interval $[0, 1]$ than a random sequence, and therefore achieving faster convergence, see e.g. Kuipers and Niederreiter (1974), Niederreiter (1992). The uniformity of a sequence is measured in terms of its discrepancy, which is a distance between a finite point set and the uniform distribution on $[0, 1]$. Therefore quasi-random sequences are also called low-discrepancy sequences.

The main field of application for QMC simulation is numerical integration. If it comes to the point, where one wants to simulate more complicated circumstances, e.g. the evolution of certain objects in time, then more sophisticated simulation methods are called into action. They become known under the term Markov chain Monte Carlo (MCMC) simulation. The practice of MCMC goes back to the paper of Metropolis et al. (1953) and in greater generality to Hastings (1970). The principle behind these MCMC algorithms is the simulation of time stationary equilibria of objects or processes. More

precisely, the Markov chain Monte Carlo algorithms deal with the problem of sampling on a given finite but huge state space and a given (stationary) distribution in an efficient way. MCMC is based on an appropriate Markov chain which converges to its stationary limit distribution. See Gilks et al. (1996) for a broad introduction to the theory and applications of MCMC.

The theory of Markov chains stands for a theory on its own until the last decades. As recently as computer performance - in sense of CPU power and memory space - increases, meaningful applications to other various areas of mathematics and other sciences arise. Nowadays Markov chains are applicable in most parts of mathematics (above all statistics), image analysis, physics, biology, social sciences, and many more.

The main result of the propaedeutic Chapter 2 will be the fact that there exist Markov chains which won't take the information about the initial state into account after sufficiently many steps. Markov chains which forget this information within reasonable time are called rapidly mixing. Several techniques have been proposed to deal with the problem of proving whether a Markov chain is rapidly mixing or not, see Behrends (2000) or the more recent book by Levin et al. (2009).

The reason why one should be interested in knowing these things is that these Markov chains are well qualified for the MCMC algorithms. Often Markov chain Monte Carlo is identified with the general Metropolis-Hastings algorithm presented in Chapter 3. The main concepts and some representative examples of the Metropolis-Hastings algorithm and Gibbs sampling are given in Chapter 3. The error analysis and convergence rates involved are specified there as well. Furthermore a general framework for the exact simulation of Markov chains using the Propp-Wilson coupling from the past approach introduced by Propp and Wilson (1996) is proposed in Chapter 3.

In Chapter 4 the Monte Carlo method for numerical integration and as a result of this the quasi-Monte Carlo method is introduced. In the quasi-Monte Carlo method sampling with pseudo random numbers is replaced by sampling with quasi-random sequences which approximate the uniform distribution more uniformly.

The main justification of MCMC methods is based on the assumption of using independent uniform points. What happens if these points are replaced by a quasi-random sequence? One may believe that the structure which guarantees the consistency of the Metropolis-Hastings algorithm would be difficult to maintain through a sequence of chosen points, since the samples are now dependent. Until recently there haven't been any publications which handle this problem. Furthermore there seemed to be hardly any intersections between the research done on MCMC and the research on quasi-Monte Carlo

methods. This changed a few years ago with the work of Owen and Tribble (2005). Their paper presents a so-called quasi-Monte Carlo Metropolis algorithm. After that further papers covering this topic was published. Chapter 5 presents the method of Owen and Tribble using low-discrepancy sequences as input for the Metropolis-Hastings algorithm. Further it includes an example where the different Monte Carlo methods are compared with each other. As you will see, with the hybrid of QMC and MCMC one can obtain great variance reductions (in comparison with the other Monte Carlo methods).

Finally I give an overview of another hybrid version of Markov chain Monte Carlo and quasi Monte Carlo. This randomized quasi Monte Carlo simulation method for Markov chains was introduced first by Lécot and Tuffin (2004). For some problems form queuing and finance, this hybrid achieves variance reductions of many thousand fold. Again I'll give an example to illustrate the variance reductions.

2 Markov Chains

2.1 Introduction

This chapter provides the general background on Markov chains which will be needed in the next chapter. Most of this background can also be found in the book by Behrends (2000) and in the lecture notes by Schmidt (2010) and Hohendorff (2005). The stochastic model of a discrete time Markov chain which we consider here will have finitely many states and consists of three components. The non-void finite state space, the initial distribution and the transition matrix.

The state space of the Markov chain is the set of all possible states $S = \{x_1, x_2, \ldots, x_l\}$ where l is an arbitrary but fixed natural number. The initial distribution of the Markov chain is defined by $\mu = (\mu_1, \ldots, \mu_l)$, where each μ_i is defined as the probability of a sample to be in state x_i at time $n = 0$, for each $x_i \in S$. It is assumed that

$$\mu_i \in [0, 1] \quad \text{and} \quad \sum_{i=1}^{l} \mu_i = 1.$$

The transition matrix of the Markov chain is a $l \times l$ matrix $\mathbf{P} = \mathbf{P}(x_i, x_j) = (p_{ij})_{i,j=1,\ldots,l}$ containing all the transition probabilities $\mathbf{P}(x_i, x_j)$ for each pair $x_i, x_j \in S$. $\mathbf{P}(x_i, x_j)$ is the probability that the Markov chain moves from state x_i to state x_j in one step, where

$$\mathbf{P}(x_i, x_j) \geq 0 \quad \text{and} \quad \sum_{j=1}^{l} \mathbf{P}(x_i, x_j) = 1.$$

A Markov chain is a stochastic process where given the present state, the future states are independent of the past. More formally let $X_0, X_1, \ldots : \Omega \to S$ be a sequence of random variables defined on the probability space $(\Omega, \mathcal{A}, \mathbf{P})$ which map into the set S. To simplify matters I will identify the state space S with the first l natural numbers

$S = \{1, 2, \ldots, l\}$ for the rest of this introductory chapter. Then

$$P(X_n = i_n | X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) = p_{i_{n-1} i_n}$$

for any $n = 1, 2, \ldots$ and $i_0, i_1, \ldots, i_n \in S$ such that $P(X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) > 0$. This property is called the *Markovian property.*

We are interested in the distribution of the Markov chain after n steps. One way to obtain this information is by simple matrix multiplication.

Proposition 1. *Consider a Markov chain with transition matrix* $\mathbf{P}$ *and initial distribution* $\mu^{(0)}$*, and denote the distribution of the Markov chain after the* n*th transition with* $\mu^{(n)}$*. Then* $\mu^{(n)} = \mu^{(0)} \mathbf{P}^n$*.*

2.2 Ergodicity

Definition 1. *Denote the transition matrix for* n *transitions by* $\mathbf{P}^{(n)}$*. Then a Markov chain with state space* S*, transition matrix* $\mathbf{P}$ *and corresponding* n*-step transition matrix* $\mathbf{P}^{(n)}$ *is called* ergodic *if the limits*

$$\pi_j = \lim_{n \to \infty} p_{ij}^{(n)} \tag{2.1}$$

1. *exist for all* $j \in S$*,*
2. *are positive and independent of* $i \in S$*,*
3. *and form a probability function* $\pi = (\pi_1, \ldots, \pi_l)^T$*, i.e.* $\sum_{j \in S} \pi_j = 1$*.*

Definition 2. *The* $l \times l$ *matrix* $\mathbf{A} = (a_{ij})$ *is called* non-negative *if all entries* a_{ij} *of* $\mathbf{A}$ *are non-negative. A non-negative matrix* $\mathbf{A}$ *is called* quasi-positive *or* primitive *if there is a natural number* $n_0 \geq 1$ *such that all entries of* $\mathbf{A}^{n_0}$ *are positive.*

Theorem 1. *The Markov chain* $X_0, X_1, \ldots$ *with state space* $S = \{1, \ldots, l\}$ *and transition matrix* $\mathbf{P}$ *is ergodic if and only if* $\mathbf{P}$ *is primitive.*

The ergodicity of Markov chains in Theorem 1 was characterized by the primitivity of the transition matrix $\mathbf{P}$. If the size of the state space gets large it became difficult to show if the transition matrix $\mathbf{P}$ is primitive. Therefore, we will derive another probabilistic way to characterize the ergodicity of a Markov chain with finite state space. For this purpose we will need the following definition.

Definition 3. *Define* $\tau_j = \min\{n \geq 0 : X_n = j\}$ *to be the number of steps until the Markov chain* $\{X_n\}$ *reaches the state* $j \in S$ *for the first time. Set* $\tau_j = \infty$ *if* $X_n \neq j$ *for all* $n \geq 0$.

Definition 4. *For arbitrary but fixed states* $i, j \in S$*, we say that the state* j *is* accessible *from state* i*, if* $p_{ij}^{(n)} > 0$ *for some* $n \geq 1$.

Lemma 1. *Let* $i \in S$ *be such that* $P(X_0 = i) > 0$*. In this case* j *is accessible from* i *if and only if* $P(\tau_j < \infty | X_0 = i) > 0$.

In the case where i is accessible from j and j is accessible from i we say that the states i and j communicate, $i \leftrightarrow j$. The property of communicating defines an equivalence relation for the states in S. Therefore the state space S can be completely subdivided into disjoint equivalence classes of communicating states.

Definition 5 (Irreducibility). *A Markov chain* $\{X_n\}$ *with transition matrix* $\mathbf{P}$ *is called* irreducible *if the state space* S *consists of only one equivalence class, i.e. all states communicate.*

Definition 6 (Aperiodicity). *The* period d_i *of the state* $i \in S$ *is given by* $d_i = \gcd\{n \geq 1 : p_{ii}^{(n)} > 0\}$*. We define* $d_i = \infty$ *if* $p_{ii}^{(n)} = 0$ *for all* $n \geq 1$*. A state* $i \in S$ *is called aperiodic if* $d_i = 1$*. A Markov chain* $\{X_n\}$ *and its transition matrix* $\mathbf{P} = (p_{ij})$ *are said to be* aperiodic *if all the states are aperiodic.*

Theorem 2. *The transition matrix* $\mathbf{P}$ *is ergodic if and only if* $\mathbf{P}$ *is irreducible and aperiodic.*

Note that this previous statement is only true if the state space is finite.

2.3 Stationarity

Definition 7 (Stationarity). *Let* $\pi = (\pi_1, \ldots, \pi_l)$ *be a probability vector. Then* π *is said to be a* stationary distribution *for the Markov chain* $\{X_n\}$ *if* $\pi = \pi P$.

Theorem 3. *Let* $X_0, X_1, \ldots$ *be an irreducible and aperiodic Markov chain with finite state space* S*. Then there exists exactly one stationary distribution* π.

Theorem 3 quotes that for an irreducible and aperiodic Markov chain with finite state space and transition matrix $\mathbf{P}$, there exists exacly one unique solution of the matrix equation

$$\pi = \pi\mathbf{P}. \tag{2.2}$$

If the Markov chain is not irreducible, there may be more solutions for equation (2.2).

Definition 8. *Let $\{X_n\}$ be a random process on a state space S. Then $\{X_n\}$ is called a stationary sequence of random variables if for arbitrary $k, n \in \{0, 1, \ldots\}$ and $i_0, \ldots, i_n \in S$*

$$P(X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n) = P(X_k = i_0, X_{k+1} = i_1, \ldots, X_{k+n} = i_n). \tag{2.3}$$

Theorem 4. *A Markov chain $\{X_n\}$ is a stationary sequence of random variables if and only if the Markov chain $\{X_n\}$ has a stationary initial distribution.*

2.3.1 A Markov chain Convergence Theorem

We go on to consider the asymptotic behavior of the distribution $\mu^{(n)}$ of a Markov chain with arbitrary initial distribution $\mu^{(0)}$. Therefore we state the main result, the Markov chain convergence theorem (Theorem 5), of this propaedeutic chapter. This convergence theorem is a version of the ergodic theorem for Markov chains.

Theorem 5 (Markov chain convergence theorem). *Consider an irreducible and aperiodic Markov chain $\{X_n\}$ with finite state space S. If we denote the distribution of the chain after the n^{th} transition by $\mu^{(n)}$ we have for any initial distribution $\mu^{(0)}$ that*

$$\mu^{(n)} \longrightarrow \pi \quad \textit{for} \quad n \to \infty, \tag{2.4}$$

where π is the unique stationary distribution according to Theorem 3.

Theorem 5 states that an ergodic Markov chain tends to forget the information about the initial distribution after sufficiently many steps. So if one is interested in sampling from the equilibrium distribution π, one way of doing this is by matrix multiplication. This is maybe a good thing if the state space is of moderate size. In case where the state space is huge, Monte-Carlo simulation will be a more efficient method to sample from the limit distribution π as you will see in the next Chapter 3.

According to Theorem 5, if we run the Markov chain for a sufficiently long time n, its distribution will be very close to the stationary distribution π, regardless of what

the initial distribution $\mu^{(0)}$ was. This is often referred to as the Markov chain reaching equilibrium as $n \to \infty$. A proof of Theorem 5, which uses nice coupling-arguments can be found in Häggström (2002). Another proof of this theorem appears in Levin et al. (2009).

2.4 Reversibility

Definition 9 (Reversibility). *A probability distribution π on the state space $S = \{1, \ldots, l\}$ is* reversible *for the Markov chain $\{X_n\}$ with transition matrix $\mathbf{P}$ if for all $i, j \in S$ we have*

$$\pi_i \mathbf{P}(i, j) = \pi_j \mathbf{P}(j, i).$$

Proposition 2. *If the probability distribution π is reversible for a Markov chain, then it is also a stationary distribution for the chain.*

The statement of this proposition is useful, since it is often easier to show that a probability distribution is reversible than showing that it is stationary. Moreover most of the Markov chains we'll deal with later on are constructed in a way that they are reversible.

2.5 Recursive Representation

Besides the Markov chain convergence theorem, the representation of a Markov chain in a recursive manner is the other main result of this chapter.

We will show that a Markov chain can be constructed from a sequence of IID random variables. Conversely we also show that one can think of a Markov chain as a solution of a recursive stochastic equation.

Let $S = \{1, 2, \ldots, l\}$ be a finite set and let $([0, 1], \mathcal{B}([0, 1])$ be the measurable space with $\mathcal{B}([0, 1])$ the Borel σ-algebra on $[0, 1]$. Recall that $(\Omega, \mathcal{A}, \mathbf{P})$ is the probability space we work with. Let $U_n : \Omega \to [0, 1]$ for $n \geq 1$ be a sequence of IID random variables taking values in $[0, 1]$. Further on let $X_0 : \Omega \to S$ be independent of the U_n's.

Theorem 6. *With the setup from above let the random variables $X_1, X_2, \ldots : \Omega \to S$ be given by the recursive stochastic equation*

$$X_n = \phi(X_{n-1}, U_n), \tag{2.5}$$

where $\phi : S \times [0,1] \to S$ is an arbitrary measurable function. Then

$$P(X_n = i_n | X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) = P(X_n = i_n | X_{n-1} = i_{n-1})$$

holds for any $n \geq 1$ and $i_0, i_1, \ldots, i_n \in S$ on the assumption that $P(X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) > 0$.

The transition probabilities

$$p_{ij} = P(X_n = j | X_{n-1} = i)$$

are given by

$$p_{ij} = P(\phi(i, U_n) = j). \tag{2.6}$$

Since the $U_1, U_2, \ldots$ are IID, p_{ij} does not depend on n. The joint probability $P(X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n)$ is given by

$$P(X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n) = \mu_{i_0} p_{i_0 i_1} \ldots p_{i_{n-1} i_n},$$

where $\mu_{i_0} = P(X_0 = i_0)$. Therefore the sequence $X_0, X_1, \ldots$ of random variables defined by the recursive equation (2.5) forms a Markov chain.

Vice versa, we show now that every Markov chain can be considered as the solution of a recursive stochastic equation. As before let $\{X_n\}$ be a Markov chain with state space S, initial distribution μ and transition matrix $\mathbf{P} = (p_{ij})$. By the help of a recursive equation of the form (2.5) we will construct a Markov chain $\{\hat{X}_n\}$ with initial distribution μ and transition matrix $\mathbf{P}$ such that

$$P(X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n) = P(\hat{X}_0 = i_0, \hat{X}_1 = i_1, \ldots, \hat{X}_n = i_n)$$

for all $i_0, \ldots, i_n \in S$ and for all $n \geq 0$.

Let $U_0, U_1, \ldots$ be a sequence of IID uniform random variables on $[0,1]$. The random variable $\hat{X}_0$ can be defined as

$$\hat{X}_0 = k \quad \text{if and only if} \quad U_0 \in \left[\sum_{i=1}^{k-1} \mu_i, \sum_{i=1}^{k} \mu_i \right), \tag{2.7}$$

for all $k \in \{1, \ldots, l\}$. Define the random variables $\hat{X}_1, \hat{X}_2, \ldots$ by the recursive equation

$$\hat{X}_n = \phi(\hat{X}_{n-1}, U_n), \tag{2.8}$$

where $\phi : S \times [0,1] \to S$ is given by

$$\phi(i,u) = \sum_{k=1}^{l} k \, \mathbb{1}_{\{\sum_{j=1}^{k-1} p_{ij} < u \leq \sum_{j=1}^{k} p_{ij}\}} \tag{2.9}$$

One will agree that the probabilities $P(\hat{X}_0 = i_0, \ldots, \hat{X}_n = i_n)$ for the sequence $\{\hat{X}_n\}$ defined by (2.7) - (2.9) are given by the stochastic recursive representation (2.5). Therefore the Markov chain $\{X_n\}$ can be considered as a solution of recursive stochastic equation.

This introductory and more general information about Markov chains can be found in many books and articles containing Markov chain theory, or using Markov chains for further assignments. Some of the more recent books are by Häggström (2002), Behrends (2000), and Levin et al. (2009). Most of the definitions and theorems stated in this chapter are taken from there and from the lecture notes of Schmidt (2010) and Hohendorff (2005).

3 Markov Chain Monte Carlo

The previous chapter taught us that given an irreducible and aperiodic transition matrix $\mathbf{P}$, there exists exactly one stationary distribution π satisfying $\pi = \pi\mathbf{P}$. We will now have a look at the inverse problem. Given a probability distribution π on a finite set S, is it possible to construct a transition matrix $\mathbf{P}$ for which π is its stationary distribution? The following Section 3.1 gives an understanding of why this is a natural problem to consider.

It can be challenging to construct random samples directly. An alternative way of constructing random samples is via approaching the samples with Markov chains. Suppose that $\{X_n\}$ is an ergodic Markov chain with state space S and equilibrium distribution π. Then by the convergence theorem, Theorem 5, X_n is approximately π-distributed when n is large.

3.1 Motivation

We consider the following problem: Given a probability distribution π on $S = \{x_1, \ldots, x_k\}$, how to simulate a random object with distribution π? To motivate this problem, we begin with a very common example, which can also be found in Häggström (2002) and Levin et al. (2009).

Example (The hard-core model). Consider a graph $G = (V, E)$ with vertex set $V = \{v_1, \ldots, v_k\}$ and edge set $E = \{e_1, \ldots, e_l\}$. Now 0 or 1 is assigned randomly to every vertex $v_i \in V$ in such a way that no two adjacent vertices take both value 1. Assignments of 0's and 1's to the vertices of G are called configurations. If a configuration fulfills the above condition that no two 1's occupy adjacent vertices, it is called feasible. We now pick a feasible configuration from the set of all feasible configurations uniformly at random. Let $\xi \in \{0, 1\}^V$ be any configuration, set Z to the total number

of feasible configurations and define a probability measure μ on $\{0,1\}^V$ by

$$\mu(\xi) = \begin{cases} \frac{1}{Z} & , \quad \xi \text{ is feasible} \\ 0 & , \quad \text{otherwise.} \end{cases}$$

Häggström (2002) noted that this model (with a graph G being a three-dimensional grid) was introduced in statistical physics to capture some of the behavior of a gas whose particles have non-negligent radii and must not overlap. In this model the 1's represent particles and the 0's represent empty locations.

Now one may wonder about the expected number of 1's in a random configuration chosen according to μ. Denote the number of 1's in a configuration by $n(\xi)$ and write X for a random configuration chosen according to μ. Then we are interested in

$$E(n(X)) = \sum_{\xi \in \{0,1\}^V} n(\xi)\mu(\xi) = \frac{1}{Z} \sum_{\xi \in \{0,1\}^V} n(\xi) \mathbb{1}_{\xi \text{ is feasible}}.$$

Even for moderately sized graphs it is clearly beyond the bounds of possibility to evaluate this sum, since the number of configurations grows exponentially in the size of the graph. Note also that the calculation of Z is computationally nontrivial. A good idea to handle this may be to change course to simulations. If we know how to simulate a random configuration X with distribution μ, then we can do this many times and estimate $E(n(X))$ by the average number of 1's in the simulations.

In this sort of situation the *Markov chain Monte Carlo (MCMC)* method comes into operation. The idea behind Markov chain Monte Carlo is the following. Suppose we can construct an irreducible and aperiodic Markov chain $\{X_n\}$ whose unique stationary distribution is π. Start to run the Markov chain with an arbitrary initial distribution. The ergodic theorem, see Theorem 5, assures that the distribution of the Markov chain converges to the stationary distribution π, at time n when $n \to \infty$.

An important focus is to determine how large n must be to obtain a sufficient approximation. But first we will focus on the task of finding Markov chains with a given stationary distribution.

There will come more examples in the next sections. A very common example is again from physics and will be the Ising model which is discussed in Section 3.4.1.

3.2 Metropolis Chains

Assume we have a Markov chain with state space S and an arbitrary stationary distribution π. Is it possible to construct a new Markov chain out of the given one in such a way that the new chain has stationary distribution π? The Metropolis algorithm will take care of this problem. The *Metropolis chain* was introduced by Metropolis et al. (1953) for a specific stationary distribution. Hastings (1970) extended the method to general chains and distributions. Since these papers are fairly old and amongst others hard to read, I took most of the following information from the more recent book by Levin et al. (2009).

3.2.1 Symmetric Base Chain

Let Φ be a symmetric transition matrix on the state space S. Φ is therefore reversible with respect to the uniform distribution on S. Let π be any probability distribution on S. We will show next how the transitions of the chain according to Φ are transformed such that a chain with stationary distribution π is achieved. The new Markov chain proceeds the following way. When at state x, the next step is done by generating a proposal from the probability distribution $\Phi(x, \cdot)$. Let the proposed new state be y. Then this step is ignored with probability $1 - a(x, y)$. This means that with probability $a(x, y)$, the state y is accepted and the next state of the Markov chain is y. With complementary probability $1 - a(x, y)$ the proposed state y is rejected and the Markov chain remains at x. If many moves are rejected the chain will slow down and in consequence of this the computational efficiency may be reduced. Nevertheless rejecting moves is sometimes necessary to achieve the desired distribution. The transition matrix $\mathbf{P}$ of the new Markov chain appears to be

$$\mathbf{P}(x, y) = \begin{cases} \Phi(x, y)a(x, y) & \text{if } x \neq y, \\ 1 - \sum_{z:\, x \neq z} \Phi(x, z)a(x, z) & \text{if } x = y. \end{cases}$$

How to pick the acceptance probability $a(x, y)$ arises from the following data which we will choose wisely.

We already know that the transition matrix $\mathbf{P}$ has stationary distribution π if it holds that

$$\pi(x)\Phi(x, y)a(x, y) = \pi(y)\Phi(y, x)a(y, x) \tag{3.1}$$

for all states $x \neq y$, remember Proposition 2. Since we have assumed that Φ is symmetric, equality (3.1) holds if and only if

$$\pi(x)a(x,y) = \pi(y)a(y,x). \tag{3.2}$$

Since $a(x,y) \leq 1$, the constraints

$$\begin{aligned} \pi(x)a(x,y) &\leq \pi(x), \\ \pi(x)a(x,y) = \pi(y)a(y,x) &\leq \pi(y). \end{aligned} \tag{3.3}$$

must be obeyed. Rejecting moves from the original chain Φ will be inefficient. Therefore the largest possible solution πa for (3.2) and (3.3) should be chosen. All solutions are bounded above by $\pi(x)a(x,y) = \min\{\pi(x), \pi(y)\}$. For this reason, the acceptance probability $a(x,y)$ is chosen to be $\min\{1, \pi(y)/\pi(x)\}$.

With the information from above we can define the Metropolis chain for a probability π and a symmetric transition matrix Φ precisely as

$$\mathbf{P}(x,y) = \begin{cases} \Phi(x,y)\, \min\left\{1, \dfrac{\pi(y)}{\pi(x)}\right\} & \text{if } x \neq y, \\ 1 - \displaystyle\sum_{z:\, x \neq z} \Phi(x,z)\, \min\left\{1, \dfrac{\pi(z)}{\pi(x)}\right\} & \text{if } x = y. \end{cases}$$

Due to the above discussion one sees that π is in fact a stationary distribution for the Metropolis chain. One also notices that the Metropolis chain only depends on the ratios $\pi(x)/\pi(y)$. This can be helpful since the $\pi(x)$ are often of the form $h(x)/Z$, where the function $h : S \to [0, \infty)$ is known and $Z = \sum_{x \in S} h(x)$ is a normalizing constant. The explicit calculation of Z is often proved do be difficult, in particular if the state space S is large. Since the Metropolis chain only depends on the $h(x)/h(y)$, the computation of the constant Z is not relevant for the simulation.

3.2.2 General Base Chains

We can define the Metropolis chain also if the original transition matrix Φ is a general matrix, not necessarily symmetric. Let Φ be a transition matrix and let π be an arbitrary probability distribution on S. Then the Metropolis chain is implemented as follows. Suppose the chain is at state x. In the next step a state y from the distribution $\Phi(x, \cdot)$

is generated. The state is accepted with probability

$$a(x,y) = \min\left\{1, \frac{\pi(y)\Phi(y,x)}{\pi(x)\Phi(x,y)}\right\}. \tag{3.4}$$

and so the chain moves to state y with acceptance probability $a(x,y)$. With probability $1-a(x,y)$ the chain stays at state x. The resulting transition matrix $\mathbf{P}$ for this Markov chain is given by

$$\mathbf{P}(x,y) = \begin{cases} \Phi(x,y)\ \min\left\{1, \dfrac{\pi(y)\Phi(y,x)}{\pi(x)\Phi(x,y)}\right\} & \text{if } x \neq y, \\ 1 - \displaystyle\sum_{z:\, x\neq z} \Phi(x,z)\ \min\left\{1, \dfrac{\pi(z)\Phi(z,x)}{\pi(x)\Phi(x,z)}\right\} & \text{if } x = y. \end{cases} \tag{3.5}$$

Theorem 7. *The transition matrix (3.5) constructed above defines a reversible Markov chain with stationary distribution π.*

Proof. First recall that we have to show $\pi(x)P(x,y) = \pi(y)P(y,x)$. We assume $x \neq y$, since the other case is obvious. We can write

$$\begin{aligned} \pi(x)P(x,y) &= \pi(x)\Phi(x,y)\ \min\left\{1, \frac{\pi(y)\Phi(y,x)}{\pi(x)\Phi(x,y)}\right\} \\ &= \min\left\{\pi(x)\Phi(x,y)\ ,\ \pi(y)\Phi(y,x)\right\} \end{aligned}$$

The fact that this equation is symmetric in x and y completes the proof. □

3.3 Gibbs Sampler

The Gibbs sampler, also known as Glauber dynamics is another commonly used special class of MCMC algorithms. In practice there are many Markov chains whose state spaces are subsets of sets of the form A^V. We thought about V as a finite vertex set of a graph and about A as a finite set of attainable values for each vertex. The hard-core model introduced before in Section 3.1 and the random q-coloring which will be introduced shortly are just two of them. A^V is the set of all possible configurations of the set A with values of set V (compare with the definition of the hard-core model introduced before). For the two finite sets V and A let π be the distribution of some random assignment of values in A to the vertices in V. Now suppose that our state space S is a subset of A^V and let π be a probability distribution on S. Once more our aim is to obtain π distributed samples. The *Gibbs sampler* for π produces a reversible Markov

chain on the state space S, with stationary distribution π. The transition probabilities are defined as follows. When at state x, the Gibbs sampler moves on in the following way. It first chooses a vertex v uniformly at random from V. Next, the Gibbs sampler chooses the new state according to the distribution π. π will be conditioned on the set of all states agreeing with x everywhere except maybe at v. Let us define this more precise. For a state $x \in S$ and a vertex $v \in V$, let

$$S(x, v) = \{y \in S : x(w) = y(w) \text{ for all } w \neq v\} \tag{3.6}$$

be the set of states equal to x at all vertices except possibly at v, and define

$$\pi^{x,v}(y) = \pi(y|S(x, v)) = \begin{cases} \dfrac{\pi(y)}{\pi(S(x, v))} & \text{if } y \in S(x, v), \\ 0 & \text{if } y \notin S(x, v) \end{cases}$$

as the conditional distribution of π with respect to the set $S(x, v)$. The update rule for a configuration x picks a vertex v uniformly at random, and chooses a new configuration with respect to $\pi^{x,v}$. Note that the distribution π is always reversible and therefore stationary for the Gibbs sampler. Now have a look at the following example.

Example (A Gibbs sampler for random q-colorings). Let $G = (V, E)$ be a graph, and let $q \geq 2$ be an integer. A q-coloring of the graph G is defined as an assignment of values from the set of colors $\{1, \ldots, q\}$, where a feasible q-coloring has no two adjacent vertices with the same color. By a random q-coloring for G we mean that one q-coloring is chosen uniformly from the set of all feasible q-colorings for G. Define π as the corresponding probability distribution on A^V, where we have that $A = \{1, \ldots, q\}$.

For a vertex $v \in V$ let x be feasible configuration. The distribution π conditioned on the set $S(x, v)$ defined in (3.6) is uniform over the set of all colors that are not used in x for a neighbor of v. For that reason the Gibbs sampler for a random q-coloring is an A^V-valued Markov chain with the following update rule. At time n do the following:

1. Pick a vertex $v \in V$ uniformly at random.

2. Pick $X_{n+1}(v)$ according to the conditioned uniform distribution $\pi^{x,v}$.

Note that the colors of all other vertices $w \neq v$ stay unchanged. The resulting Markov chain is aperiodic and has stationary distribution π. If the chain is also irreducible, the above construction method became a useful MCMC algorithm. Nevertheless, the

irreducibility of the Markov chain depends on G and q, and is a nontrivial problem to solve (The graph coloring problem is NP-complete.).

3.3.1 Metropolis Chains in Comparison to Gibbs Sampler

Again let A be a finite set, V the vertex set of a graph and π a probability distribution on the state space A^V. Additionally assume that we have a Markov chain which chooses a vertex v at random and has some arbitrary update procedure for the configuration at vertex v. In general, the resulting Markov chain won't have stationary distribution π. However, Section 3.2 including Metropolis chains teaches us that this Markov chain can be modified so that the new chain has stationary distribution π.

One may believe that the Metropolis chain coincides with the Gibbs chain described before. But this isn't entirely true. The chains may be very similar but won't coincide exactly. For a better comprehension consider the following example.

Example (random q-colorings revisited). Take a Markov chain on the state space A^V of all q-colorings. The colorings don't have to be feasible. Choose a vertex $v \in V$ and a color among all q colors uniformly at random. Then recolor the vertex v with the chosen color. Now we want to transform this chain into a Metropolis chain. Note that π is the probability distribution which is uniform over the space of all feasible configurations. Let the chain be in a feasible configuration. Then the Metropolis update recolors a chosen vertex with probability 1 if the recoloring with the proposed color keeps the configuration feasible. Otherwise there is no recoloring.

The Gibbs sampler acts in a different way. The difference between the Gibbs sampler and the Metropolis chain is the probability that a configuration remains the same. If there are a admissible colors for a vertex $v \in V$ chosen to be updated, the chance that the coloring remains the same is $1/a$ for the Gibbs chain. For the Metropolis chain the probability that the configuration remains the same is $1 - (a-1)/q$.

3.4 Metropolis-Hastings Algorithm

The term Metropolis-Hastings is used for the generalization by Hastings (1970) of the Metropolis algorithm. Again let let S be the finite state space. Let π be a probability distribution over the state space S. We assume that $\pi(x) > 0$ for all $x \in S$. Once more we want to construct a Markov chain $\{X_n\}$ with stationary limit distribution π. The

transition matrix $\mathbf{P} = \mathbf{P}(x, y)$ for this Markov chain is given as

$$\mathbf{P}(x, y) = \Phi(x, y)\, a(x, y) \quad \forall x, y \in S \quad \text{with} \quad x \neq y, \tag{3.7}$$

where $\Phi = \Phi(x, y)$ is the arbitrary irreducible and aperiodic transition matrix we want to modify. The matrix $A = (a(x, y))$ defines the acceptance probabilities. So when at state x, a proposed new state y is accepted with probability $a(x, y)$. The matrix $A = (a(x, y))$ is defined in the most general sense as

$$a(x, y) = \frac{m(x, y)}{1 + t(x, y)}. \tag{3.8}$$

The matrix $M = (m(x, y))$ is an arbitrary symmetric matrix bounded by

$$0 < m(x, y) \leq 1 + \min\{t(x, y), t(y, x)\}, \tag{3.9}$$

and $t(x, y)$ is defined as

$$t(x, y) = \begin{cases} \frac{\pi(x)\Phi(x,y)}{\pi(y)\Phi(y,x)} & \text{if } \Phi(x, y) > 0. \\ 0 & \text{if } \Phi(x, y) = 0, \end{cases} \tag{3.10}$$

Theorem 8. *The transition matrix $\mathbf{P} = \mathbf{P}(x, y)$ made up of (3.7) to (3.10) is ergodic and the pair $(\mathbf{P}, \pi)$ is reversible.*

With this setup in mind, we figure out how the Metropolis-Hastings will proceed.

Algorithm (Metropolis-Hastings Algorithm)

1. *Let π to be the stationary distribution on the state space S. Let $\Phi = \Phi(x, y)$ be an arbitrary transition matrix on S.*

2. *Pick a deterministic starting value X_0.*

3. *When at state X_n generate a proposal Y_{n+1} from $\Phi(X_n, \cdot)$.*

4. *Perform a Bernoulli experiment with acceptance probability $a(X_n, Y_{n+1})$. The probability $a(x, y)$ is defined as in (3.8).*

5. *If the experiment was successful then accept the proposal and set $X_{n+1} = Y_{n+1}$. Otherwise reject Y_{n+1} and stay at state X_n, i.e. $X_{n+1} = X_n$.*

6. *Increment n by 1 and continue at step 3.*

This is how the Metropolis-Hastings algorithm works in its most generality. In the following I will list some more specific examples of the Metropolis-Hastings algorithm and some typical choices for the transition probabilities Φ.

- The original Metropolis chain arises if we set $m(x, y)$ in (3.9) equal to the upper bound $1 + \min\{t(x, y), t(y, x)\}$ for all $x, y \in S$. Compare this result with the chain constructed in Section 3.2.
- The Barker algorithm, Barker (1965), is obtained by setting $m(x, y) = 1$ for all $x, y \in S$. The acceptance rule then results in

$$a(x, y) = \frac{\pi(x)\Phi(x, y)}{\pi(y)\Phi(y, x) + \pi(x)\Phi(x, y)}.$$

Some common choices for the transition matrix Φ are

- the original Metropolis algorithm (where Φ is supposed to be symmetric): $\Phi(x, y) = \Phi(y, x)$,
- the random walk Metropolis-Hastings method: $\Phi(x, y) = \Phi(y - x)$,
- and the independence sampler: $\Phi(x, y) = \Phi(y)$ independently of x.

In addition to the above examples of the random q-coloring, and to gain a better insight of how the algorithms work in practice, another famous example of an important Markov chain follows.

3.4.1 Example: Ising Model

The following example is a famous problem from physics and is cited from the book by MacKay (2003). An Ising model is an array of spins (e.g., atoms that can take states ± 1) that are magnetically coupled to each other. If one spin is, say in the $+1$ state, then it is energetically favourable for its immediate neighbors to be in the same state, in the case of an ferromagnetic model.

Let the state x of an Ising model with N spins be a vector in which each component x_n takes values -1 or $+1$. If two spins m and n are neighbors we write $m \sim n$. The coupling between neighboring spins is J. We define $J_{mn} = J$ if m and n are neighbors and $J_{mn} = 0$ otherwise. The energy of a state x is

$$E(x; J, H) := -\left(\frac{1}{2}\sum_{m,n} J_{mn}x_m x_n + \sum_n H x_n\right),$$

where H is the applied external field. If $J > 0$ then the model is ferromagnetic, and if $J < 0$ it is antiferromagnetic. We included the factor $1/2$ because each pair is counted twice in the first sum. At equilibrium at temperature T, the probability that the state is x is

$$P(x|\beta, J, H) = \frac{1}{Z(\beta, J, H)} \exp(-\beta E(x; J, H)). \tag{3.11}$$

The function $Z(\beta, J, H)$ is a normalizing constant and is defined as

$$Z(\beta, J, H) := \sum_x \exp(-\beta E(x; J, H)).$$

The parameter $\beta \geq 0$ determines the importance of the energy function. In the physical interpretation, β is the reciprocal of temperature.

Monte Carlo Simulation

Here we study two-dimensional planar Ising models using a simple Gibbs sampling method. Starting from some initial state, a spin n is selected at random, and the probability that it should be $+1$ given the state of the other spins and the temperature is computed,

$$P(+1|b_n) = \frac{1}{1 + \exp(-2\beta b_n)}, \tag{3.12}$$

where b_n is the local field

$$b_n := \sum_{m:m\sim n} J x_m + H.$$

The factor of 2 appears in equation (3.12) because the two spin states are $\{+1, -1\}$ rather than $\{+1, 0\}$. Spin n is set to $+1$ with that probability, and otherwise to -1. Then the next spin to update is selected at random. After sufficiently many iterations, this procedure converges to the equilibrium distribution (3.11).

An alternative to the Gibbs sampling formula (3.12) is the Metropolis algorithm, in which we consider the change in energy that results from flipping the chosen spin from its current state x_n,

$$\Delta E = 2 x_n b_n,$$

and adopt this change in the configuration with probability

$$P(\text{accept}; \Delta E, \beta) = \begin{cases} 1 & \Delta E \leq 0 \\ \exp(-\beta \Delta E) & \Delta E > 0. \end{cases}$$

For graphics and a more precise analysis of the Ising model see e.g. MacKay (2003) or Levin et al. (2009).

3.5 Error Analysis for MCMC Simulation

An ergodic Markov chain converges to its unique stationary distribution after sufficiently many steps. Since we just have a finite amount of time available, no matter how long the run of a chain is, the states will never have exactly the stationary distribution. Therefore one should be interested in the behavior of the error of the Markov chain Monte Carlo estimation and the variance of the distribution of the chain.

3.5.1 Asymptotic Variance of Estimation

Consider the following general setup of a statistical model. Let X be a discrete random vector, taking values in a finite state space $S = \{x_1, \ldots, x_l\}$ of cardinality $|S| = l$ according to a probability vector π. Moreover define $\theta = E(\varphi(X))$ to be the expected value of an arbitrary vector $\varphi = (\varphi_1, \ldots, \varphi_l)$ of functions $\varphi_i : S \to \mathbb{R}$. So

$$\theta = \sum_{i=1}^{l} \pi(i)\varphi_i.$$

Our intention is the simulation of the expectation θ with MCMC simulation. Let the random variable

$$\hat{\theta}_n = \frac{1}{n} \sum_{k=0}^{n-1} \varphi(X_k), \quad \forall n \geq 1,$$

be an estimator for θ, where $\{X_n\}$ denotes an ergodic Markov chain with state space S. Let μ be an arbitrary initial distribution and let $\mathbf{P} = \mathbf{P}(x_i, x_j)$ be the transition matrix such that π is the stationary limit distribution of the chain.

In general the two distributions μ and π do not coincide. This results in a biased estimation of the expectation θ. The following representation formula is useful for determining the bias

$$E(\hat{\theta}_n) - \theta.$$

Let

$$E(\hat{\theta}_n) = \frac{1}{n} \mu^T \sum_{k=0}^{n-1} P^k \varphi \quad \text{for all} \quad n \geq 1. \tag{3.13}$$

$\mathbf{P}$ is irreducible and aperiodic. With the Markov chain convergence theorem (see Theorem 5) in mind

$$\lim_{n\to\infty} E(\hat{\theta}_n) = \theta$$

results from equation (3.13). This means that the MCMC estimator $\hat{\theta}_n$ for the expectation θ is asymptotically unbiased. Equipped with this setup we now want to investigate the asymptotic behavior of the variance $\mathrm{Var}(\hat{\theta}_n)$.

The following definition and lemmata are quoted from Schmidt (2010).

Definition 10. *Let Π be the $l \times l$ matrix consisting of the l identical row vectors π. Then the inverse matrix*

$$Z = (I - (\mathbf{P} - \Pi))^{-1} \tag{3.14}$$

is called the fundamental matrix *of $\mathbf{P}$.*

Note that matrix $I - (\mathbf{P} - \Pi)$ is always invertible. This is because of the following fact: For a $l \times l$ matrix A with $\lim_{n\to\infty} A^n = 0$ it follows that $I - A$ is invertible.

Lemma 2. *Define $\sigma^2 := \sum_{i=1}^{l} \pi(i)(\varphi_i - \theta)^2$ and let $Z = (I - (\mathbf{P} - \Pi))^{-1}$ be the fundamental matrix of $\mathbf{P}$. Then*

$$\lim_{n\to\infty} n\,Var(\hat{\theta}_n) = \sigma^2 + 2\pi^T diag(\varphi)(Z - I)\varphi. \tag{3.15}$$

The following representation formulae for the fundamental matrix Z are helpful to prove Lemma 2.

Lemma 3. *The fundamental matrix $Z = (I-(\mathbf{P}-\Pi))^{-1}$ of the irreducible and aperiodic transition matrix $\mathbf{P}$ has the representation formulae*

$$Z = I + \sum_{k=1}^{\infty} (\mathbf{P}^k - \Pi) \tag{3.16}$$

and

$$Z = I + \lim_{n\to\infty} \sum_{k=1}^{n-1} \frac{n-k}{n} (\mathbf{P}^k - \Pi). \tag{3.17}$$

(3.16) converges because of the facts that $\mathbf{P}^k - \Pi = (\mathbf{P} - \Pi)^k$ and $I + A + \ldots + A^{n-1} = (I - A)^{-1}(I - A^n)$ holds for a $l \times l$ matrix A. For the complete proof of Lemma 3 see e.g. Schmidt (2010). The following proof is also taken from there.

Proof of Lemma 2. Let us represent the variation $\mathrm{Var}(\hat{\theta}_n)$ as

$$n^2\mathrm{Var}(\hat{\theta}_n) = E\left(\sum_{k=0}^{n-1}\varphi(X_k)\right)^2 - \left(\sum_{k=0}^{n-1}E\big(\varphi(X_k)\big)\right)^2. \tag{3.18}$$

Therefore

$$\begin{aligned} n^2\mathrm{Var}(\hat{\theta}_n) = &\sum_{k=0}^{n-1}E\big(\varphi^2(X_k)\big) \\ &+ 2\sum_{0\le j<k\le n-1}E\big(\varphi(X_k)\varphi(X_j)\big) \\ &- \left(\sum_{k=0}^{n-1}E\big(\varphi(X_k)\big)\right)^2. \end{aligned}$$

With this representation in mind we will first prove equation (3.15) for the case where the initial distribution $\mu = \pi$. In this case we get

$$\left(\sum_{k=0}^{n-1}E\big(\varphi(X_k)\big)\right)^2 = (n\theta)^2 \quad \text{and} \quad \sum_{k=0}^{n-1}E\big(\varphi^2(X_k)\big) = n\sum_{i=1}^{l}\pi(i)\varphi_i^2.$$

Furthermore, since the Markov chain $\{X_n\}$ is stationary,

$$\sum_{0\le j<k\le n-1}E\big(\varphi(X_k)\varphi(X_j)\big) = \sum_{k=1}^{n-1}(n-k)\sum_{0\le j<k\le n-1}E\big(\varphi(X_0)\varphi(X_k)\big),$$

where

$$E\big(\varphi(X_0)\varphi(X_k)\big) = \sum_{i=1}^{l}\sum_{j=1}^{l}\pi(i)\varphi_i\mathbf{P}(x_i,x_j)^{(k)}\varphi_j = \pi^T\mathrm{diag}(\varphi)\mathbf{P}^k\varphi.$$

where $\mathbf{P}^k = \mathbf{P}^{(k)} = \mathbf{P}(x_i,x_j)^{(k)}$ denotes the transition matrix after the kth step. Now

we put these results from above together and get

$$\begin{aligned}\frac{1}{n}\mathrm{Var}\left(\sum_{k=0}^{n-1}\varphi(X_k)\right) &= \sum_{i=1}^{l}\pi(i)\varphi_i^2 + 2\pi^T\mathrm{diag}(\varphi)\sum_{k=1}^{n-1}\frac{n-k}{n}\mathbf{P}^k\varphi - n\theta^2 \\ &= \sigma^2 + 2\pi^T\mathrm{diag}(\varphi)\Big(\sum_{k=1}^{n-1}\frac{n-k}{n}\mathbf{P}^k\varphi - \frac{n-1}{2}\Pi\varphi\Big) \\ &= \sigma^2 + 2\pi^T\mathrm{diag}(\varphi)\Big(\sum_{k=1}^{n-1}\frac{n-k}{n}(\mathbf{P}^k - \Pi)\Big)\varphi.\end{aligned}$$

Note that the second equality holds because of the fact

$$\theta^2 = \pi^T\mathrm{diag}(\varphi)\Pi\varphi.$$

With the representation formula (3.17) for $Z - I$ this implies (3.15).

After proving the statement for $\mu = \pi$, we move on and prove equation (3.15) for an arbitrary initial distribution μ. Therefore we introduce a more precise notation. We write $X_0^\mu, X_1^\mu, \ldots$ instead of $X_0, X_1, \ldots$ and $\hat{\theta}_n^\mu$ instead of $\hat{\theta}_n$. The proof is completed if we show that

$$\lim_{n\to\infty} n\left(\mathrm{Var}(\hat{\theta}_n^\pi) - \mathrm{Var}(\hat{\theta}_n^\mu)\right) = 0. \tag{3.19}$$

Therefore we denote for $0 < r < n-1$

$$Y_r^{\bullet} = \sum_{k=0}^{r-1}\varphi(X_k^{\bullet}) \quad \text{and} \quad Z_{nr}^{\bullet} = \sum_{k=r}^{n-1}\varphi(X_k^{\bullet}).$$

Due to equation(3.18) the following equation results:

$$\begin{aligned}&n^2\left(\mathrm{Var}(\hat{\theta}_n^\pi) - \mathrm{Var}(\hat{\theta}_n^\mu)\right) \\ &= \big(E(Y_r^\pi + Z_{nr}^\pi)^2 - E(Y_r^\mu + Z_{nr}^\mu)^2\big) - \big((EY_r^\pi + EZ_{nr}^\pi)^2 - (EY_r^\mu + EZ_{nr}^\mu)^2\big) \\ &= \big(E(Y_r^\pi)^2 - (EY_r^\pi)^2 - E(Y_r^\mu)^2 + (EY_r^\mu)^2\big) \\ &\quad + 2E\big((Y_r^\pi - EY_r^\pi)(Z_{nr}^\pi - EZ_{nr}^\pi)\big) - 2E\big((Y_r^\mu - EY_r^\mu)(Z_{nr}^\mu - EZ_{nr}^\mu)\big) \\ &\quad + \big(E(Z_{nr}^\pi)^2 - (EZ_{nr}^\pi)^2 - E(Z_{nr}^\mu)^2 + (EZ_{nr}^\mu)^2\big).\end{aligned}$$

Denote the three summands in the last expression by $\mathcal{I}_r$, $\mathcal{II}_{rn}$ and $\mathcal{III}_{rn}$, respectively.

Because of the fact that $\mathcal{I}_r$ does not depend on n it immediately follows

$$\lim_{n\to\infty} \frac{1}{n} \mathcal{I}_r = 0.$$

Next set $c = \max_{x_i \in S} |\varphi(x_i)|$. Since the state space S is finite it follows that c is finite too. Therefore we get that

$$\frac{1}{n}\mathcal{II}_{rn} \leq 4rcE\left(\frac{1}{n}\left|Z^{\pi}_{nr} - E(Z^{\pi}_{nr})\right|\right) + 4rcE\left(\frac{1}{n}\left|Z^{\mu}_{nr} - E(Z^{\mu}_{nr})\right|\right),$$

and

$$\frac{1}{n}|Z^{\cdot}_{nr} - E(Z^{\cdot}_{nr})| \leq 2c \quad \text{with probability } 1 \quad \forall n > r.$$

Now

$$\lim_{n\to\infty} \frac{1}{n} |Z^{\pi}_{nr} - E(Z^{\pi}_{nr})| = \frac{1}{n} |Z^{\mu}_{nr} - E(Z^{\mu}_{nr})| = 0$$

implies that

$$\lim_{n\to\infty} \frac{1}{n} \mathcal{II}_{rn} = 0 \quad \text{for any } r > 0.$$

For the third summand $\mathcal{III}_{rn}$ and for $n > r > 0$ the following estimate holds:

$$\begin{aligned}
\frac{1}{n}\mathcal{III}_{rn} &\leq \frac{1}{n}\sum_{i=1}^{l}\left(E(Z^{\delta_i}_{n-r,0})^2 - (EZ^{\delta_i}_{n-r,0})^2\right)\left|\pi(i) - \mu^{(r)}_i\right| \\
&\leq \underbrace{\sup_{n>0} \max_{j\in\{1,\ldots,l\}} \frac{1}{n+r} E\left(Z^{\delta_j}_{n0} - EZ^{\delta_j}_{n0}\right)^2}_{<\infty} \sum_{i=1}^{l}\left|\pi(i) - \mu^{(r)}_i\right|.
\end{aligned}$$

The supremum is finite and because of the fact that the Markov chain $X^{\mu}_0, X^{\mu}_1, \ldots$ is ergodic the last sum gets arbitrarily small for sufficiently large r. Therefore (3.19) holds which completes the proof. □

The mean square error $E((\hat{\theta}_n - \theta)^2)$ of the MCMC estimator $\hat{\theta}_n$ for the estimation θ is given by the sum of the squared bias $(E(\hat{\theta}_n - \theta))^2$ and the variance $\mathrm{Var}(\hat{\theta}_n)$ of the estimator $\hat{\theta}_n$,

$$E\left((\hat{\theta}_n - \theta)^2\right) = \left(E(\hat{\theta}_n - \theta)\right)^2 + \mathrm{Var}(\hat{\theta}_n). \tag{3.20}$$

Both summands in equation (3.20) converge to 0 if $n \to \infty$. The convergence rates, however, differ. The variance $\mathrm{Var}(\hat{\theta}_n)$ converges with a rate of $\mathcal{O}(n^{-1})$, see Lemma 2. Whereas the convergence rate of the squared biases $(E(\hat{\theta}_n - \theta))^2$ converges faster, at a rate of $\mathcal{O}(n^{-2})$. The consequence for the convergence of the mean square error

$E((\hat{\theta}_n - \theta)^2)$ of $\hat{\theta}_n$ is the following. Because of the fact that the convergence rate of the asymptotic variance is slower than the one for the squared biases, it has a greater influence on the asymptotic behavior of the mean square error. This is why it makes sense choose a Markov chain which obtains a small asymptotic variance and to accept that the bias may increase due to this.

The next theorem characterizes the behavior of the asymptotic variance under certain conditions for the transition matrix. We therefore introduce the following notation for the asymptotic variance given in (3.15). For some arbitrary function $\varphi : S \to \mathbb{R}$, a transition matrix $\mathbf{P}$ on S and reversible probability distribution π on S let

$$V(\varphi, \mathbf{P}, \pi) = \lim_{n\to\infty} n\mathrm{Var}(\hat{\theta}_n).$$

Theorem 9. *Let $\mathbf{P}_1 = \mathbf{P}_1(x_i, x_j)$ and $\mathbf{P}_2 = \mathbf{P}_2(x_i, x_j)$ be two transition matrices on S. Let π be a probability distribution on S such that the pairs $(\mathbf{P}_1, \pi)$ and $(\mathbf{P}_2, \pi)$ are reversible. Further let $\mathbf{P}_1$ and $\mathbf{P}_2$ be such that $\mathbf{P}_1(x_i, x_j) \geq \mathbf{P}_2(x_i, x_j)$ for all $x_i, x_j \in S$ with $x_i \neq x_j$. This means that all entries of the transition matrix $\mathbf{P}_1$ are greater or equal than the corresponding entries of the transition matrix $\mathbf{P}_2$, accept the ones in the diagonal. Then, for some arbitrary function $\varphi : S \to \mathbb{R}$,*

$$V(\varphi, \mathbf{P}_1, \pi) \leq V(\varphi, \mathbf{P}_2, \pi).$$

The following proof is taken from Schmidt (2010).

Proof. Let $\mathbf{P} = \mathbf{P}(x_i, x_j)$ be a transition matrix on the state space S and let π be a probability distribution on S such that the pair $(\mathbf{P}, \pi)$ is reversible. In this proof I'll write p_{ij} instead of $\mathbf{P}(x_i, x_j)$ for reasons of simplicity, so that the forumlae in the proof are clearer. Theorem 9 is proven if we can show that

$$\frac{\partial}{\partial p_{ij}} V(\varphi, \mathbf{P}, \pi) \leq 0, \quad \forall x_i, x_j \in S \quad \text{with } x_i \neq x_j. \tag{3.21}$$

From equation (3.15) in Lemma 2 it follows

$$\frac{\partial}{\partial p_{ij}} V(\varphi, \mathbf{P}, \pi) = 2\pi^T \mathrm{diag}(\varphi) \frac{\partial Z}{\partial p_{ij}} \varphi. \tag{3.22}$$

Remember that $Z = (I - (P - \Pi))^{-1}$ denotes the fundamental matrix of $\mathbf{P}$. Next we

have a look at the partial derivative of $ZZ^{-1} = I$. We get that

$$\frac{\partial Z}{\partial p_{ij}} Z^{-1} + Z \frac{\partial Z^{-1}}{\partial p_{ij}} = 0$$

which is equivalent with

$$\frac{\partial Z}{\partial p_{ij}} = -Z \frac{\partial Z^{-1}}{\partial p_{ij}} Z.$$

We apply this result to equation (3.22) and get

$$\frac{\partial}{\partial p_{ij}} V(\varphi, \mathbf{P}, \pi) = -2\pi^T \mathrm{diag}(\varphi) Z \frac{\partial Z^{-1}}{\partial p_{ij}} Z\varphi. \tag{3.23}$$

Now we express Z by its representation formula (3.16), see Lemma 3. Since π is reversible for $\mathbf{P}$ we get that for arbitrary $x_i, x_j \in S$

$$\begin{aligned}
\pi(i) z_{ij} &= \pi(i)\delta_{ij} + \sum_{k=1}^{\infty} \Big(\pi(i) p_{ij}^{(k)} - \pi(i)\pi(j) \Big) \\
&= \pi(j)\delta_{ji} + \sum_{k=1}^{\infty} \Big(\pi(j) p_{ji}^{(k)} - \pi(j)\pi(i) \Big) \\
&= \pi(j) z_{ji}.
\end{aligned}$$

This implies

$$\begin{aligned}
\pi^T \mathrm{diag}(\varphi) Z &= \Big(\sum_{i=1}^{l} \pi(i)\varphi_i z_{i1}, \ldots, \sum_{i=1}^{l} \pi(i)\varphi_i z_{il} \Big) \\
&= \Big(\pi(1) \sum_{i=1}^{l} z_{1i}\varphi_i, \ldots, \pi(l) \sum_{i=1}^{l} z_{li}\varphi_i \Big) \\
&= (Z\varphi)^T \mathrm{diag}(\pi).
\end{aligned}$$

If we apply this result to equation (3.23) we get

$$\frac{\partial}{\partial p_{ij}} V(\varphi, \mathbf{P}, \pi) = -2(Z\varphi)^T \mathrm{diag}(\pi) \frac{\partial Z^{-1}}{\partial p_{ij}} Z\varphi = 2(Z\varphi)^T \mathrm{diag}(\pi) \frac{\partial \mathbf{P}}{\partial p_{ij}} Z\varphi. \tag{3.24}$$

The second equality is a consequence of the fact that

$$\frac{\partial Z^{-1}}{\partial p_{ij}} = -\frac{\partial \mathbf{P}}{\partial p_{ij}}$$

which is clear due to the definition of the fundamental matrix Z (3.14).

Since $\mathbf{P} = (p_{ij})$ is a stochastic matrix and the pair $(\mathbf{P}, \pi)$ is reversible, only the probabilities p_{ij} where $x_i < x_j$ (or alternatively the entries p_{ij} where $x_i > x_j$) can be chosen arbitrarily. This means that for each pair $x_i, x_j \in S$ such that $x_i \neq x_j$ the transition probabilities p_{ji}, p_{ii} and p_{jj} can be expressed by p_{ij}.

For each pair $x_{i'}, x_{j'} \in S$ the entry $(\mathrm{diag}(\pi)(\partial\mathbf{P}/\partial p_{ij}))_{i'j'}$ of the matrix product $\mathrm{diag}(\pi)(\partial\mathbf{P}/\partial p_{ij})$ is given by

$$\left(\mathrm{diag}(\pi)\frac{\partial \mathbf{P}}{\partial p_{ij}}\right)_{i'j'} = \begin{cases} -\pi(i) & \text{if } (x_{i'}, x_{j'}) = (x_i, x_i) \text{ or } (x_{i'}, x_{j'}) = (x_j, x_j), \\ \pi(i) & \text{if } (x_{i'}, x_{j'}) = (x_i, x_j) \text{ or } (x_{i'}, x_{j'}) = (x_j, x_i), \\ 0 & \text{else.} \end{cases}$$

This implies that the matrix $\mathrm{diag}(\pi)(\partial\mathbf{P}/\partial p_{ij})$ is non-negative definite. Therefore it holds that for all $x \in \mathbb{R}^l$

$$x^T \mathrm{diag}(\pi)\frac{\partial \mathbf{P}}{\partial p_{ij}} x \leq 0.$$

Hence this implies for equation (3.24) that for all $x_i, x_j \in S$ with $x_i \neq x_j$

$$\frac{\partial}{\partial p_{ij}} V(\varphi, \mathbf{P}, \pi) = 2(Z\varphi)^T \mathrm{diag}(\pi)\frac{\partial \mathbf{P}}{\partial p_{ij}} Z\varphi \leq 0.$$

Which was to be proven. □

A consequence of Theorem 9 is the following: Given an arbitrary but fixed transition matrix $\Phi = \Phi(x_i, x_j)$, the Metropolis chain introduced in Section 3.2 has the smallest asymptotic variance amongst all other Metropolis-Hastings algorithms.

3.5.2 Burn-in and Allocating Capacities

As discussed in the previous section, the mean square error of the MCMC estimator $\hat{\theta}_n$ for θ depends not only on the asymptotic variance but also on the bias $E(\hat{\theta}_n - \theta)$. The bias decreases if the first k samples of the estimating expectation formula (3.13), which are still affected by the initial setting, are skipped for the estimation of θ. But how much of a run of the Markov chain should be thrown away on grounds that the chain may not yet have reached equilibrium and therefore $E(\hat{\theta}_n - \theta)$ not yet has vanished.

The term *burn-in time* is to be understood as the period of time which should elapse to the beginning of the simulation in order to eliminate the crudest errors and distribution

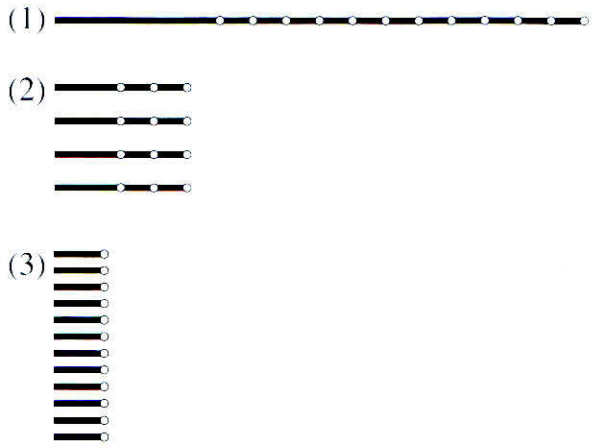

Figure 3.1: Three possible MCMC ways of obtaining twelve samples. The time is represented in horizontal lines, and the samples by white circles. (1) A single run having a long burn-in period followed by a long sampling period. (2) Four medium-length runs and a medium-length burn in period. (3) Twelve short runs.

variances caused by the initial setup for the simulation. After the burn-in, the recording of the simulation starts.

Once we have decided how many samples N are required for the estimation of θ, the next question arises. How can the available computer capacities be utilized well in view of obtaining these samples. A typical Markov chain Monte Carlo experiment has an initial period in which the parameters for the simulation, e.g. the initial states, are adjusted. Thereafter a burn-in period follows. During this time the simulation should start to forget about the initial information and should get closer to equilibrium so that the bias of the simulation $E(\hat{\theta}_n - \theta)$ vanishes. After the burn-in, as the simulation continues, we start gathering the data of the simulation. Let $\{X_n\}_{n=1}^N$ be the list of states after the burn-in. We hope that those states in the list are roughly independent samples from our state space. There are several strategies for recording this data (see Figure 3.1, taken from MacKay (2003)):

1. Start one Markov chain and do a long run. After a long burn-in period, start to obtain all N samples from it.

2. Start a couple of Markov chains and do some medium length runs with different initial conditions. After a medium length burn-in time, start to obtain a few samples from each of the chains.

3. Start N Markov chains and make some short runs on each of them. Each chain is started from another initial state and the only state that is used for our estimation is the final state of each simulation.

The first strategy has the best chance of reaching equilibrium. The last method may have the advantage of less correlations between the recorded samples, since each sample

comes from another chain, with an initial state independent of the other chains. The second strategy is popular for Markov chain Monte Carlo simulation. It combines the benefits of the first and third method and avoids their drawbacks of not being able to spot bad convergence rates if just one chain is started and discarding a lot of samples of the burn-in period for many chains. See Gilks et al. (1996). After determining the right strategy of simulating, one may be interested in the convergence rate of the simulation.

3.6 Rate of Convergence and Mixing Time

There exist Markov chains which will 'forget' the information of the starting position and converge to their stationary distribution. This is the case if the Markov chain is irreducible and aperiodic and when this happens within a reasonable time, the chain is called rapidly mixing. It is obvious that the faster the chain 'forgets' the information about the initial distribution, the faster it converges to its equilibrium distribution and consequently the more exact will the result of the simulation be. Therefore this section deals with the question how close the distribution of the chain is to equilibrium. Further subjects of interest are the convergence rate of the simulation, the mixing time of the chain and determining some bounds of these parameters.

Consider a primitive transition matrix $\mathbf{P}$ with l different eigenvalues $\lambda_1, \ldots, \lambda_l \in [-1, 1]$, stationary distribution $\pi = (\pi_1, \ldots, \pi_l)^T$on S and initial distribution $\mu^{(0)}$ on S. The assumption that $\mathbf{P}$ has l different eigenvalues is quite natural. To convince yourself see for instance Behrends (2000). Order the eigenvalues as follows $1 = |\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_l|$.

Theorem 10 (Perron-Frobenius). *With the setup from above*

$$\sup_{x_j \in S} |\mu_j^{(n)} - \pi(j)| = \mathcal{O}(|\lambda_2|^n).$$

If the aperiodic and irreducible transition matrix $\mathbf{P}$ is in addition reversible one can show the more accurate upper bound

$$\max_{x_i \in S} |\mu_i^n - \pi(i)| \leq \frac{1}{\sqrt{\min_{x_i \in S} \pi(i)}} |\lambda_2|^n. \tag{3.25}$$

A precise derivation of (3.25) can be found in Schmidt (2010). However, the practical benefit of the estimate (3.25) is quite limited, since it can be difficult to determine the eigenvalue λ_2 if the number of states is large. Furthermore, the factor in front of $|\lambda_2|^n$

does not depend on the choice of the initial distribution $\mu^{(0)}$. Therefore we consider an alternative convergence estimate, the so called χ^2-constrast, Schmidt (2010). For the explanation of this concept some more definitions are needed.

Definition 11. *Let* $\mathbf{P}$ *be a transition matrix of an ergodic Markov chain with stationary distribution* π. *The matrix* $M := \mathbf{P}\tilde{\mathbf{P}}$ *is called the* multiplicative reversible version *of the transition matrix* $\mathbf{P}$ *if we set*

$$\tilde{\mathbf{P}}(x_i, x_j) := \frac{\pi(j)\mathbf{P}(x_j, x_i)}{\pi(i)}$$

For M the multiplicative reversible version of $\mathbf{P}$ it holds that M is reversible.

Definition 12. *The* χ^2-constrast *of* μ *given* ν *is defined as*

$$\chi^2(\mu, \nu) := \sum_{i \in S} \frac{(\mu_i - \nu_i)^2}{\nu_i}$$

where we require $\nu_i > 0$ *for all* $i \in S$.

Definition 13 (Distance in Variation). *Let* S *be a countable space and let* μ *and* ν *be probability distributions on* S. *Then the total variation distance* $d_{TV}(\mu, \nu)$ *between* μ *and* ν *is defined by*

$$d_{TV}(\mu, \nu) = \frac{1}{2}|\mu - \nu| = \frac{1}{2}\sum_{i \in S} |\mu_i - \nu_i|.$$

An upper bound for the variation distance $d_{\mathrm{TV}}(\mu^{(n)}, \pi)$ between the distribution of the Markov chain after the nth transition and the stationary distribution is given in the next proposition.

Theorem 11. *For any initial distribution* $\mu^{(0)}$ *and for all* $n \in \mathbb{N}$ *it holds that*

$$d^2_{TV}(\mu^{(n)}, \pi) \leq \frac{\chi^2(\mu^{(0)}, \pi)}{4} \lambda^n_{M,2},$$

where $\lambda^n_{M,2}$ denotes the second largest eigenvalue of the matrix M^n, where M is defined as in Definition 11. The reader is referred to Schmidt (2010) for the proof of Theorem 11.

Markov chains converge (under the right conditions, see Convergence Theorem 5) to their stationary distribution. The key tools for quantifying this convergence are the total variation distance and the mixing time.

3.6.1 Standardizing Distance from Stationarity

A natural thing one might want to know is how far apart is the Markov chain from the desired distribution after n steps of simulation in the worst case. We therefore define

$$d(n) := \max_{x \in S} \|\mathbf{P}^n(x, \cdot) - \pi\|_{TV} .$$

Often it is possible to bound $\|\mathbf{P}^n(x, \cdot) - \mathbf{P}^n(y, \cdot)\|_{TV}$, uniformly over all pairs of states (x, y). We therefore define

$$\bar{d}(n) := \max_{x,y \in S} \|\mathbf{P}^n(x, \cdot) - \mathbf{P}^n(y, \cdot)\|_{TV} .$$

The relation between $d(n)$ and $\bar{d}(n)$ can be seen in the following lemma.

Lemma 4. *Let $d(n)$ and $\bar{d}(n)$ be defined as above. Then*

$$d(n) \leq \bar{d}(n) \leq 2d(n). \tag{3.26}$$

The proof of the inequality (3.26), and further inequalities related to the distances $d(n)$ and $\bar{d}(n)$ are given in Levin et al. (2009).

3.6.2 Bounds on Mixing Time

Both in the introductory chapter and later the terms 'rapidly mixing' and 'mixing time' have arisen. Now it is time to define these parameters. The mixing time is the time a Markov chain needs for being near to stationarity. This parameter can then be used to determine the running time for the MCMC simulation.

Definition 14. *The* mixing time *is defined by*

$$t_{mix}(\epsilon) := \min\{n : \ d(n) \leq \epsilon\} \quad \textit{and} \quad t_{mix} := t_{mix}(1/4). \tag{3.27}$$

The $1/4$ in (3.27) is arbitrary, but a value of ϵ less than $1/2$ is needed for the following.

Levin et al. (2009) give some upper bounds for the mixing time t_{mix}. It holds that for every non-negative integer k

$$d(kt_{\text{mix}}(\epsilon)) \leq \bar{d}(kt_{\text{mix}}(\epsilon)) \leq \bar{d}(t_{\text{mix}}(\epsilon))^k \leq (2\epsilon)^k.$$

By taking $\epsilon = 1/4$ above yields $d(kt_{\text{mix}}) \leq 2^{-k}$ and

$$t_{\text{mix}}(\epsilon) \leq \lceil \log_2 1/\epsilon \rceil \, t_{\text{mix}}.$$

The better the upper bounds on the mixing the higher is the confidence that the simulation algorithm performs well. So the question arises if a given upper bound is as good as possible. Therefore one is also interested in lower bounds on the mixing time.

Counting Bound

One idea to obtain lower bounds on the mixing time is the following. Let $\{X_n\}$ be an irreducible and aperiodic Markov chain on the state space S and transition matrix $\mathbf{P}$. For reasons of simplicity suppose that the stationary distribution π is uniform over S. Let the chain run for n steps. If after this run the reachable states don't form a significant fraction of the state space, the distribution of the chain cannot be close to uniform. Define the number of states reachable from x as

$$d_{\text{out}}(x) := |\{y : \mathbf{P}(x, y) > 0\}|,$$

and set

$$\Delta = \max_{x \in S} d_{\text{out}}(x).$$

Define S_n^x to be the set of states reachable from x in n steps, and note that $|S_n^x| \leq \Delta^n$. If $\Delta^n < (1 - \epsilon)|S|$ holds, it follows from the definition of total variation distance that

$$\|\mathbf{P}^n(x, \cdot) - \pi\|_{TV} \geq \mathbf{P}^n(x, S_n^x) \geq 1 - \frac{\Delta^n}{|S|} > \epsilon.$$

This implies that

$$t_{\text{mix}}(\epsilon) \geq \frac{\log(|S|(1 - \epsilon))}{\log \Delta}.$$

Diameter Bound

Let $\mathbf{P}$ be an irreducible and aperiodic transition matrix on S. Let the graph $G = (S, E)$ have the vertex set equal to the state space S and edge set

$$E = \{(x, y) : P(x, y) + P(y, x) > 0, \forall x, y \in S\}.$$

The shortest path between two vertices x and y is a path from x to y of minimal length. The diameter of a graph is defined as the length of the longest of all shortest paths between two distinct vertices. Define the diameter of the Markov chain to be the diameter of the graph G.

Suppose L to be the diameter of the Markov chain, and let x_0 and y_0 be states at maximal distance L. Then $\mathbf{P}^{\lfloor (L-1)/2 \rfloor}(x_0, \cdot)$ and $\mathbf{P}^{\lfloor (L-1)/2 \rfloor}(y_0, \cdot)$ are both positive on strictly disjoint vertex sets. Therefore $\bar{d}(\lfloor (L-1)/2 \rfloor) = 1$ and for any $\epsilon < 1/2$,

$$t_{\text{mix}}(\epsilon) \geq \frac{L}{2}.$$

These bounds for the mixing time are model-specific bounds and were taken from Levin et al. (2009), where more bounds on the mixing time can be found. For explicit examples the bounds can be calculated more precisely. Furthermore, the mixing time collaborates with another time called coupling time, introduced in the next section.

3.7 Coupling Algorithms and Perfect MCMC Simulation

One of the main drawbacks in Markov chain Monte Carlo simulation is the difficulty of determining the burn-in time, see Section 3.5.2. Another drawback is determining the correlation between X_n and X_{n+1} in terms of variance calculation of the estimate, see Section 3.5.1. The exact sampling method introduced in this section avoids both problems by producing independent samples which are exactly π distributed.

As mentioned above this section will explain algorithms that are still based on Markov chains, but this class of algorithms simulates a given distribution π not only approximately but in a certain sense exactly. Therefore, these techniques are called *perfect* or *exact* MCMC methods.

As already remarked, these algorithms didn't have the main drawbacks of the classical MCMC algorithms. As we have seen in the previous section, some practical problems

of the Markov chain Monte Carlo method appear. The desired distribution can just be reached approximately. Therefore it requires a lot of work to show that the distribution which we want to simulate is approximated with given accuracy. Furthermore the optimal choice of the stopping time requires huge theoretical investigation or much intuition and practical experience.

The algorithm presented in this section produces an exact sample, and has a clear stopping criterion. In general the exact sampling algorithms won't be faster and will need a huge amount of memory space. The basic principle is again an irreducible and aperiodic Markov chain, with a stationary limit distribution π which we want to simulate. Instead of starting the Markov chain at time 0 and run to the future, the algorithm starts far away in the past and stops at time 0. A main aspect is that for each state in the state space a copy of this Markov chain is started and that the paths will couple in an adequate way. In most of the applications the state space will be large, so that the implementation of the algorithm will fail. For a more precise mathematical model of this procedure we need the following definition and notation.

Definition 15 (Coupling). *Let $\{X'_n\}$ and $\{X''_n\}$ be two stochastic processes taking their values in the same countable (or finite) state space S. The two processes are said to* couple *if there exists an almost surely finite random time τ such that*

$$n \geq \tau \Rightarrow X'_n = X''_n.$$

The random variable τ is called the coupling time *for the two processes.*

Once two Markov chains are in the same state at the same time, they stay together and pass through the same path, we say that coalescence has occurred.

We can define the coupling time not only for two but also for finitely many Markov chains. Let $S = \{x_1, \ldots, x_l\}$ be a finite state space. Define for each state $x_i \in S$ an irreducible and aperiodic Markov chain

$$X^{(m,i)} = (X_m^{(m,i)}, X_{m+1}^{(m,i)}, \ldots)$$

where $m \in \{-1, -2, \ldots\}$ denotes the initial time (starting position) of the chain.

Let $X_m^{(m,i)} = x_i$ be the deterministic initial state and let $\mathbf{P}$ be the transition matrix of the chain, such that π is the equilibrium distribution of $X^{(m,i)}$. Let $U = (U_0, U_{-1}, \ldots)$ be a sequence of independent and uniformly distributed random variables on the interval $[0, 1]$. The Markov chain $X^{(m,i)}$ can be constructed recursively through an update

function $\phi : S \times [0,1] \to S$, see Section 2.5. So let $X^{(m,i)}$ be given by

$$X_n^{(m,i)} = \phi(x_k, U_n) \quad \text{if} \quad X_{n-1}^{(m,i)} = x_k.$$

Definition 16. *The random variable* $\tau = \min\{-m \geq 1 : X_0^{(m,1)} = \ldots = X_0^{(m,l)}\}$ *is called* coupling time. *Set* $\tau = \infty$ *if there is no integer* $-m$ *such that coalescence can be reached.*

Theorem 12. *Suppose that* $P(\tau < \infty) = 1$. *Then for all* $m \leq -\tau$,

$$X_0^{(m,1)} = \ldots = X_0^{(m,l)}.$$

Moreover, for arbitrary $m \leq -\tau$ *and* $i, j \in \{1, \ldots, l\}$, $X_0^{(m,i)} = X_0^{(-\tau,j)}$ *and* $X_0^{(m,i)}$ *is distributed according to the ergodic measure* π.

3.7.1 Propp Wilson

We use the setup of a finite state space S, the IID random numbers, and an increasing sequence $N_1, N_2, \ldots$ of positive integers from above. A commonly used sequence is $\{1, 2, 4, 8, \ldots\}$. Propp and Wilson (1996) showed that the choice 2^{m-1} for N_m results in an almost optimal run of the algorithm. With this information in mind we now formulate the Propp Wilson algorithm, first introduced by Propp and Wilson (1996). An interruptible version of Propp and Wilson's perfect sampling algorithm is Fill's algorithm, given by Fill (1998).

Propp and Wilson's algorithm is based on the idea that if a chain was started at time $n = -\infty$ in any state $X_{-\infty}$ it would be in equilibrium by time $n = 0$. So X_0 would be an exact sample of the ergodic limit distribution π. For the implementation of this idea we make use of the coupling arguments defined before. We first find a time $-N$ such that X_0 does not depend on X_{-N}, and then we determine X_0 by starting chains from all states at time $n = -N$ and following them to time $n = 0$. The Propp-Wilson algorithm is a procedure for finding $-N$ and X_0. Since the procedure gets successively deeper into the past and the coupling process, this sort of algorithm is also called *Coupling from the Past (CFTP)*. Here is how the algorithm proceeds.

Algorithm (Propp-Wilson Algorithm)

1. *Set* $m = 1$

2. *For each state* $x_i \in S$ *start a Markov chain in state* x_i *at time* $-N_m$ *and let it run*

until time 0. *Update the chain by successively applying the update function* ϕ *and the random numbers* $U_{-N_m+1}, \ldots, U_{-1}, U_0$.

3. *If all* l *chains have coalesced and have reached the same state* $\tilde{x}$ *at time* 0 *then stop the algorithm and use* $\tilde{x}$ *as a sample. Otherwise increase* m *by* 1 *and go on with step* 2.

An important fact is that the random numbers $U_{-N_m+1}, \ldots, U_{-1}, U_0$ and the update function ϕ have to be the same for all l chains started at time $-N_m$. It is also crucial, when going back in time (from $-N_m$ to $-N_{m+1}$), to reuse the same random numbers $U_{-N_m+1}, \ldots, U_{-1}, U_0$ already drawn. These conditions have to be considered so that the algorithm produces correct results.

The questions one can ask now are: Will the algorithm terminate? And if so, will it give a correct and unbiased sample? The answers to this are given in the next section.

Correctness and Termination of the Propp Wilson algorithm

Theorem 13 ($0-1$**-Law for Termination).** *If there exists an* $m* \in \mathbb{N}$, $\tilde{x} \in S$ *and random numbers* $U_{-N_{m*}}, \ldots, U_{-1} \in [0,1]$ *with*

$$\phi(\phi(\ldots\phi(\phi(x_k, U_{-N_{m*}}), U_{-N_{m*}+1}), \ldots, U_{-2}), U_{-1}) = \tilde{x}$$

for all $k \in \{1, \ldots, l\}$, *then the Propp-Wilson algorithm terminates with probability* 1, *and the CFTP coupling time is finite. Otherwise the algorithm never terminates.*

Proof. For sure, the algorithm will never terminate if there are no realizations which fulfill these conditions. So the second part of the proof is trivial. Let

$$\phi^{(N)}(x_k, (U)_{-N}^{-1}) := \phi(\phi(\ldots\phi(\phi(x_k, U_{-N}), U_{-N+1}), \ldots, U_{-2}), U_{-1})$$

for $N \in \mathbb{N}$ and $U_{-N_{m*}}, \ldots, U_{-1} \in (0,1]$. We take a look at the vector

$$\Phi_1 = (\phi^{(N_{m*})}(x_k, (U)_{-N_{m*}}^{-1}))_{x_k \in S} \in S^S,$$

which contains the position of all chains at time 0, if the algorithm was started at time $-N_{m*}$. Suppose that the stepfunction $\phi(x_k, .)$ has just nontrivial steps. The condition of the theorem implies that there exists an $\epsilon > 0$ such that $P(\Phi_1 \in M) > \epsilon$, where M is the set of all constant mappings from $S \to S$. The probability that the coupling time τ

of the Propp-Wilson chain is larger than N_{m*} is therefore $\leq 1-\epsilon$. The mappings

$$\Phi_r = (\phi^{(N_{m*})}(x_i, (U)_{-rN_{m*}}^{-(r-1)N_{m*}-1}))_{x_i \in S} \in S^S \quad r \in \mathbb{N}$$

are independent copies of Φ_1. Let $R_* = \inf\{r \in \mathbb{N} : \Phi_r \in M\}$ be the smallest $r \in \mathbb{N}$, such that Φ_r gets coupled. Then R_* is bounded above by a geometric random variable with parameter ϵ. R_* thus has a finite expected value. If we insert the first R_* chains into each other we get the mapping $\Phi_1 \circ \Phi_2 \circ \ldots \circ \Phi_{R_*} = (\phi^{(R_* N_{m*})}(x_k, (U)_{-R_* N_{m*}}^{-1}))_{x_k \in S}$ which lies in M, and is therefore coupled. Because of $\tau \leq R_* N_{m*}$ also τ has got a finite expected value. If we choose m large enough such that $T_m \geq R_* N_{m*}$ then the Propp-Wilson algorithm terminates after the mth step. □

Corollary 1. *If we have an irreducible, aperiodic and reversible Markov chain, then the Propp-Wilson algorithm terminates and returns a random sample distributed exactly according to the equilibrium distribution of the Markov chain.*

For more details about the perfect sampler and for the proof of Corollary 1 see Casella et al. (2001).

Theorem 14 (Correctness). *Consider an ergodic Markov chain with finite state space $S = \{x_1, \ldots, x_l\}$ and stationary distribution π. Suppose that $\phi : S \times [0,1] \to S$ is an update function for the Markov chain and let $N_1, N_2, \ldots$ be an increasing sequence of positive integers. If the perfect sampler terminates, then we have*

$$P(\tilde{x} = x_i) = \pi(i) \qquad \forall i \in \{1, \ldots, k\}$$

where $\tilde{x}$ represents the state in which all l chains ended up at time 0, i.e. $\tilde{x}$ is the output of the algorithm.

Proof. For fixed $\epsilon > 0$ and $x_i \in S$ one have to show that

$$|P(\tilde{x} = x_i) - \pi(i)| < \epsilon$$

holds. By assumption, the algorithm terminates with probability 1. So if we choose m large enough we can achieve that

$$P(\text{after } N_m \text{ steps at most the algorithm has terminated}) \geq 1 - \epsilon.$$

For such an m we start a Markov chain at time $-N_m$ and run it up to time 0. By using the same update function and random numbers, we run a second Markov chain with initial distribution equal to the stationary distribution π. Suppose that this second chain ends up in a state $\tilde{y}$ at time 0. Note that $\tilde{y}$ is also distributed exactly according to π, since this is the stationary distribution. Furthermore,

$$P(\tilde{x} \neq \tilde{y}) \leq \epsilon$$

since we chose m large enough. Therefore

$$\begin{aligned} P(\tilde{x} = x_i) - \pi(i) &= P(\tilde{x} = x_i) - P(\tilde{y} = x_i) \\ &\leq P(\tilde{x} = x_i, \tilde{y} \neq x_i) \\ &\leq P(\tilde{x} \neq \tilde{y}) \leq \epsilon. \end{aligned}$$

Similarly, one gets that $\pi(i) - P(\tilde{x} = x_i) \leq \epsilon$. By combining these two results we get

$$|P(\tilde{x} = x_i) = \pi(i)| < \epsilon.$$

□

The proof of Theorem 13 is taken from König (2003) and the proof of Theorem 14 from Hohendorff (2005).

3.8 Monotone Coupling Algorithms

If the state space S is large, it is nearly impossible to implement a Markov chain for each state of the state space and the practical usage of the CFTP algorithm introduced in the previous Section 3.7.1 fails. In many applications there are possibilities to make use of structural characteristics, such that the number of the required chains enormously decreases. One of these instances is monotonicity.

Suppose now that the state space S of our Markov chain has a natural partial ordering $\leq$. Furthermore suppose that the update rule ϕ has the nice characteristic that $x \leq y$ implies $\phi(x, U) \leq \phi(y, U)$ almost surely with respect to U and $x, y \in S$. If a Markov chain has this property, we say that it is monotone. Note that the property of preserving the partial ordering depends only on the randomizing operations and not on the Markov chain itself. The appropriate coupling algorithm for the monotone Markov chain is then called *monotone CFTP* algorithm.

If there exists an ordering on the state space S, then there also exists a smallest element $\hat{0}$ and a greatest element $\hat{1}$ with $\hat{0} \leq x \leq \hat{1}$ for all $x \in S$. Suppose we can construct a Markov chain for which $x \leq y$ implies $\phi(x, U) \leq \phi(y, U)$ almost surely with respect to U. Then coalescence is reached if and only if the chain started in $\hat{0}$ is coupled with the chain started in $\hat{1}$.

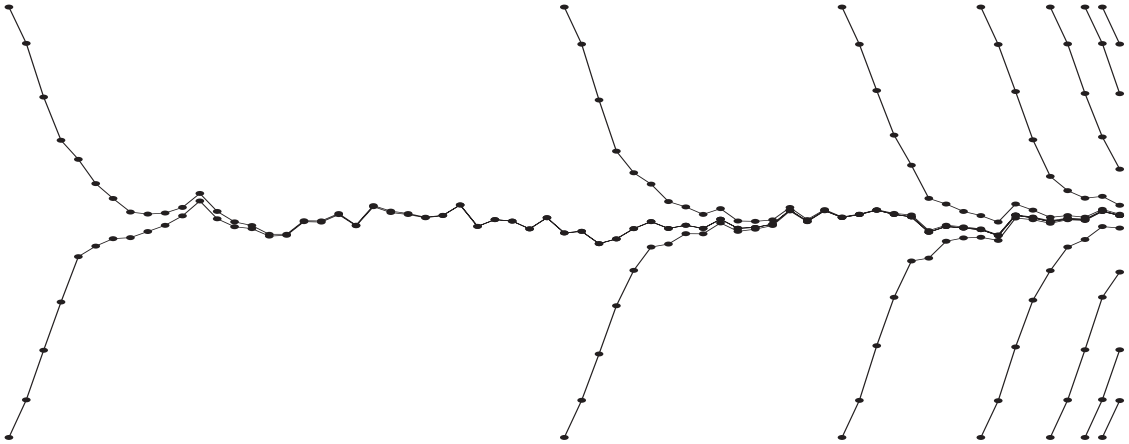

Figure 3.2: Illustration of a monotone coupling. Pictured are the heights of the upper and lower trajectories started at various starting times in the past.

If we take a look at the last N steps of the Markov chain, we can figure out what would happen if the Markov chain were in state $\hat{1}$ at time $-N$, and determine where it would be at time 0. The Markov chain admits a partial ordering and therefore it is for sure in a state which is $\leq \hat{1}$ at time $-N$, and since the randomizing operations respect the partial order, we obtain an upper bound on the state at time 0. In the same way one can get a lower bound on the state at time 0 by applying the last N steps of the Markov chain to the state $\hat{0}$. If it happens that the upper and lower bounds coincide, then we are done since we have determined the state at time 0. If this is not the case, we can go deeper into the past and start a chain at time e.g. $-2N$. This gives us new and better upper and lower bounds for the state at the present time. Repeat this procedure of going deeper into the past as long as the upper and lower bounds are equal. When the algorithm goes back in time, it reuses the randomizing operations for the period of time already passed in the previous runs, i.e. it uses the same update function and the same U_i's. For an illustration see Figure 3.2 taken from Levin et al. (2009).

A more mathematically formal description of the above is given now. Define

$$\phi_a^b(x, U) = \phi_{b-1}(\phi_{b-2}(\dots \phi_{a+1}(\phi_a(x, U_a), U_{a+1}), \dots, U_{b-2}), U_{b-1}),$$

where $U = (\ldots, U_{-1}, U_0)$ and $a < b$ integers. If for a sequence of random numbers $(U_{-N}, U_{-N+1}, \ldots, U_{-2}, U_{-1})$ with $U_i \in [0,1]$ the equation

$$\phi^0_{-N}(\hat{0}, U) = \phi^0_{-N}(\hat{1}, U), \tag{3.28}$$

is satisfied, then (3.28) implies that $\phi^0_{-N}(x, U)$ takes on the same value for all $x \in S$. This is because of the monotonic random operations and the fact that the upper and lower bounds coincide. This means that it is suffices to consider two trajectories and that there is no need to start trajectories in all $|S|$ possible states. Let N_* denote the smallest value of N for which $\phi^0_{-N}(\hat{0}, U) = \phi^0_{-N}(\hat{1}, U)$.

Some applications of the (monotone) CFTP technique are illustrated in Propp and Wilson (1996), Propp and Wilson (1998) and Levin et al. (2009), including examples like the hardcore model and the Ising model, already mentioned in previous sections.

3.8.1 Bounding the Coupling Time

This section is about determining the coupling time of a monotone Markov chain by dint of upper and lower bounds. The expected run time of the perfect sampling procedure is related to both the coupling time, see Section 3.7, and the mixing time defined in Section 3.6. So one may guess that the coupling time is linked to the mixing time of the Markov chain , which means that if the monotone Markov chain is rapidly mixing, then coalescence is also reached quite quickly. If the mixing time is not known, it can be estimated from the coupling time. Further we want to bound the probability that the algorithm takes much longer than the estimates told us.

For reason of simplicity define $f_t(x) = \phi(x, U_t)$ and let F^0_t be

$$F^0_t = f_{-1} \circ f_{-2} \circ \ldots \circ f_t.$$

Recall that the random variable N_* denotes the smallest N such that $F^0_{-N}(\hat{0}) = F^0_{-N}(\hat{1})$ which is equal to the smallest N for which F^0_{-N} is constant. Define N^* to be the smallest N such that

$$F^n_0(\hat{0}) = F^n_0(\hat{1}).$$

Note that the probability that $F^0_{-n}(\cdot)$ is not constant equals the probability that F^n_0 is not constant, $P(N_* > n) = P(N^* > n)$. Although the running time of the CPFT procedure depends on N_* we will switch from N_* to the theoretically simpler N^*. This hasn't any effect on the following calculations since N_* and N^* are governed by the same

probability distribution.

Since we want to identify the coupling time with the mixing time, we will consider the following three measures of progress related to the equilibrium distribution π. The measures are

$$\begin{aligned} &E(N^*), \\ &P(N^* > k) \quad \text{for particular or random } k, \text{ and} \\ &\bar{d}(k) = \max_{\mu_1,\mu_2} ||\mu_1^{(k)} - \mu_2^{(k)}|| \quad \text{for a particular } k. \end{aligned}$$

Here $\mu^{(k)}$ denotes the distribution of the Markov chain after the kth step when started at time 0 in a random state given by the initial distribution $\mu^{(0)}$.

Lemma 5. *Let s be the length of the longest chain (totally ordered subset) in the partially ordered state space S. Then*

$$\frac{P(N^* > k)}{s} \leq \bar{d}(k) \leq P(N^* > k).$$

Next we note that $P(N^* > k)$ is submultiplicative. To check this see Propp and Wilson (1996).

Lemma 6. *Let k_1 and k_2 be two nonnegative integer variables which might be constant. Then*

$$P(N^* > k_1 + k_2) \leq P(N^* > k_1) \cdot P(N^* > k_2).$$

The next lemma gives an estimate of the tail probabilities for N^* with respect to the expected value of N^*, and vice versa.

Lemma 7.

$$kP(N^* > k) \leq E(N^*) \leq \frac{k}{P(N^* \leq k)}$$

It was mentioned before that the mixing time and the coupling time go hand in hand with each other. Here follows the formal characterization of the statement. The mixing time t_{mix} is defined to be the smallest k for which $\bar{d}(k) \leq \epsilon$. Define s to be the length of the longest chain in the partially ordered state space. Because of the fact that $\bar{d}(k)$ is submultiplicative (a proof of this statement can be found in Section 4.4 of Levin et al.

(2009) or in Aldous and Diaconis (1987)), after $k = t_{\text{mix}}(1 + \ln s)$ steps, it holds that $\bar{d}(k) \leq \epsilon/s$. Hence $P(N^* > k) \leq \epsilon$ by Lemma 6. According to Lemma 7 it follows that

$$E(N^*) \leq k/(1-\epsilon) < 2k = 2t_{\text{mix}}(1 + \ln s).$$

Many Markov chains have a sharp threshold phenomenon also known as 'cutoff phenomenon'. This means that after running the chain for a duration of length $(1-\epsilon)t_{\text{mix}}$ the states are far apart from being random, but after a duration of length $(1+\epsilon)t_{\text{mix}}$ the states are very close to being random. For such chains it holds that the coupling time is less than $\mathcal{O}(t_{\text{mix}} \log s)$. Further information on this topic as well as the proofs of the previous lemmata can be found in Propp and Wilson (1996).

4 Quasi Monte Carlo Methods

4.1 Introduction

The *Monte Carlo (MC)* simulation is a method from stochastics. The basis of this method are random experiments repeated for many times. Problems which are too complicated to be solved analytically can be solved numerically by Monte Carlo simulation. The foundation is built by theorems from probability theory and first of all by the law of large numbers. With the help of todays computer power, sufficiently many random experiments can be produced. The Monte Carlo method is used for the simulation of problems of statistical behavior, e.g. the simulation of complex processes, where the straight analysis isn't possible, or the numerical integration of functions which can't be solved directly. The focus here is on numerical integration. Monte Carlo is a very popular method and is in wide use ranging from finance to atomic physics. This is because it is a simple method which is easy to use and still robust. The robustness is also reflected in the convergence rate of the method. The convergence rate of Monte Carlo integration is $\mathcal{O}(N^{-1/2})$ which is independent of the dimension of the integral. On the other hand the Monte Carlo method can be extremely slow.

It is possible to accelerate the convergence rate by using variance reduction methods, which reduce the constant in front of the $\mathcal{O}(N^{-1/2})$. Some of these variance reducing methods are outlined in Caflisch (1998). Another approach for improving the convergence rate is a subclass of Monte Carlo algorithms and is called *quasi-Monte Carlo (QMC)*. This sort of algorithms use deterministic sequences instead of pseudo-random sequences. These deterministic versions of random sequences are called quasi-random sequences or low-discrepancy sequences. The aim of quasi-random point sequences is not the imitation of true random point sequences but to provide a better uniformity of the points. Therefore the points are correlated to each other. This results in an acceleration of the convergence rate. The quasi-Monte Carlo method converges faster than Monte Carlo integration using pseudo-random points, at a rate of $\mathcal{O}(N^{-1}\log^k N)$ for some constant k.

4.2 Random Numbers and Pseudo-Random Numbers

The foundation of Monte Carlo simulation is built on random numbers and random sampling, respectively. The success of the Monte Carlo method depends not only on the appropriate choice of the statistical model, but also on the proper choice of the random numbers, which will simulate the random variables in the stochastic model. Generating random numbers means that given a distribution function F on the interval $[0, 1]$, we want to produce a sequence of real numbers in $[0, 1]$ that reflects the behavior of an independent and identically distributed (IID) random variable with distribution F. Actually, if we talk of random numbers, we mean pseudo-random numbers generated on the computer by a deterministic algorithm with rather few input parameters. This fact implies that therefore reproducing the pseudo-random numbers and storage of them is reasonably viable. To see if the generated pseudo-random numbers perform in a reasonable manner, they should pass a number of statistical tests for randomness. Since pseudo-random numbers are produced deterministically, they can not pass each arbitrary test of randomness. Therefore they should be generated according to the needs of every single stochastic model and its special statistical properties to pass the corresponding statistical tests. Different requirements on random numbers and tests for randomness are defined and presented in Knuth (1981).

After we got to know something about random numbers and pseudo-random numbers, which built the core of Monte Carlo simulation, we continue with the method itself.

4.3 Monte Carlo Method

The object or process to be simulated will be seen as a statistical model. Therefore it can be estimated by random sampling. In this thesis, I will focus on the problem of numerical integration.

A main ingredient for Monte Carlo simulation is the rearrangement of the numerical problem into a stochastic model. In case of numerical integration this won't be too tricky, since each integral can be interpreted as the expectation of some random variable. The statistical properties of these random variables should be considered and analyzed as well, e.g. one should know about the statistical dependence or independence of the variables. The next important step is producing random samples which imitate these properties. To guarantee a sort of accuracy of the simulation-result one should repeat the calculation for a considerable amount of time with new modified random samples.

The quantity to be simulated is seen as a statistical model and is then estimated by random sampling, as mentioned above. Consider the integral

$$\int_\Omega f(u)\,du.$$

with integration domain $\Omega \subseteq \mathbb{R}^s$ and $0 < \lambda_s(\Omega) < \infty$, where λ_s denotes the s-dimensional Lebesgue measure. If we want an approximation of this integral we can interpret it as follows. Define Ω as a probability space with measure $d\mu = du/\lambda_s(\Omega)$ and let f be Lebesgue-integrable. Then we have

$$\int_\Omega f(u)\,du \quad = \quad \lambda_s(\Omega)\int_\Omega f\,d\mu \quad = \quad \lambda_s(\Omega)E(f), \tag{4.1}$$

where $E(f)$ is the expectation of the random variable f. Our goal is to estimate the expectation $\theta = E(f)$ via Monte Carlo integration. Therefore the problem of numerical integration is transformed into the problem of estimating an expectation.

Consider f to be a random variable on an arbitrary probability space $(\Omega, \mathcal{A}, \lambda)$. The Monte Carlo estimation for the expected value θ is obtained as follows. Let $a_1, \ldots, a_N \in \Omega$ be N independent λ-distributed random samples and

$$\hat{\theta}_n \;=\; \frac{1}{N}\sum_{n=1}^{N} f(a_n). \tag{4.2}$$

Then the strong law of large numbers assures that $\hat{\theta}_n$ converges almost surely to θ,

$$\lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} f(a_n) = E(f) \quad \lambda^\infty\text{-almost everywhere.} \tag{4.3}$$

λ^∞ defines the product measure of denumerable many copies of λ.

Now let us have a look back at the initial situation and the aim to approximately calculate the integral $\int_\Omega f(u)\,du$. As an immediate consequence of (4.3) we get the following: When applying (4.2) to (4.1) we obtain the Monte Carlo estimate

$$\int_\Omega f(u)\,du \approx \frac{\lambda_s(\Omega)}{N}\sum_{n=1}^{N} f(x_n), \tag{4.4}$$

where $x_1, \ldots, x_N$ are independent μ-distributed random samples from Ω.

In the following we assume Ω to be the normalized integration domain $[0,1]^s$. This is for reasons of simplicity.

4.3.1 Convergence of the Monte Carlo Method

After estimating θ with $\hat{\theta}_n$ we want to concretize the integration error

$$\epsilon_N(f) = \theta(f) - \hat{\theta}(f).$$

Define the bias as $E(\epsilon_N(f))$ and the mean square error as $E(\epsilon_N(f)^2)^{1/2}$. With the central limit theorem (CLT) as theoretical background, the following assertion can be made, see e.g. Feller (1968). Recall that the CLT says that for finite variance, the sum of independent and identically distributed random variables converges to a normal distribution. The next theorem describes the size and statistical properties of the Monte Carlo integration error.

Theorem 15. *Suppose that f has finite variance σ^2. Then*

$$\epsilon_N(f) \approx \sigma N^{-1/2}\nu \quad \textit{for large } N,$$

where ν is $N(0,1)$-distributed and the constant $\sigma = \sigma(f)$ is the square root of the variance of f. That is

$$\sigma(f) = \left(\int_{[0,1]^s} (f(x) - \theta(f))^2 dx\right)^{1/2}.$$

More precisely it holds that

$$\begin{aligned}\lim_{N\to\infty} P\left(a < \frac{\sqrt{N}}{\sigma}\epsilon_N < b\right) &= P\left(a < \nu < b\right) \\ &= \int_a^b (2\pi)^{-1/2} e^{-x^2/2} dx.\end{aligned}$$

So the assertion of this theorem is that the Monte Carlo integration error $\epsilon_N(f)$ is of size $\mathcal{O}(N^{-1/2})$ and is influenced by a constant σ which is the square root of the variance of f. The more precise statement says that the Monte Carlo integration error is approximately normal distributed. The fact that the bounds are tight, since the result is an equality, can be seen as an advantage. As one may have noticed, the given bound is of probabilistic nature, so the result is no absolute upper bound for the integration error. This is a drawback since if one handles sensitive data, one may wish to have exact results and not just probabilistic ones. Another fact is that the probabilistic error bound for Monte Carlo integration holds under the condition that the function f is square integrable. If f fulfills additional regularity conditions, they won't influence the

integration error. Practical experience has shown that this is not quite true and that a more regular function leads to a faster convergence rate and therefore to a smaller error ϵ_N. This can be seen as another drawback of Monte Carlo integration.

As already mentioned in the introductory section there are two methods for the error reduction of Monte Carlo integration. The first consists in variance reduction methods which transform the integrand with the result of decreasing the variance of f and therefore the constant σ. The second is to replace the pseudo-random points by a quasi-random sequence. This second method is discussed in Section 4.5 more deeply.

4.4 Monte Carlo versus quasi-Monte Carlo

Quasi random sequences are constructed in a way such that they are more uniform than pseudo-random numbers which are independent and uniformly distributed. This fact of more uniformity is easily seen in Figure 4.1. This figure shows a pseudo-random sequence in contrast to a quasi random sequence, namely a Sobol sequence in two dimensions, and is taken from Caflisch (1998). Pseudo-random numbers are used for standard Monte Carlo methods. Numerical integration methods which use Monte Carlo converge at a rate of $\mathcal{O}(N^{-1/2})$ where N is the number of samples. The problem with pseudo-random points in Monte Carlo simulations is the clumping of the points. This clumping of the points is clearly evident in Figure 4.1. It occurs because the points of the pseudo-random sequence are independent and know nothing about each other. As opposed to this, the points of a quasi random sequence are dependent of each other. These correlations of the points prevent clumping. The better uniformity is well seen in Figure 4.1. This better uniformity has an effect on the convergence rate. Quasi Monte Carlo integration using quasi random sequences converges at a rate of $\mathcal{O}(N^{-1}(\log N)^k)$.

The field of application for Monte Carlo methods reaches from optimizing problems via simulation problems right through to numerical integration. In contrast, quasi Monte Carlo methods are limited to numerical integration due to the correlation between the quasi random points. Nevertheless, simulation resultings can often be represented as expectations. These are integrals and therefore quasi Monte Carlo methods are applicable.

After outlining several characteristics of the quasi Monte Carlo method the time has come to become more concrete.

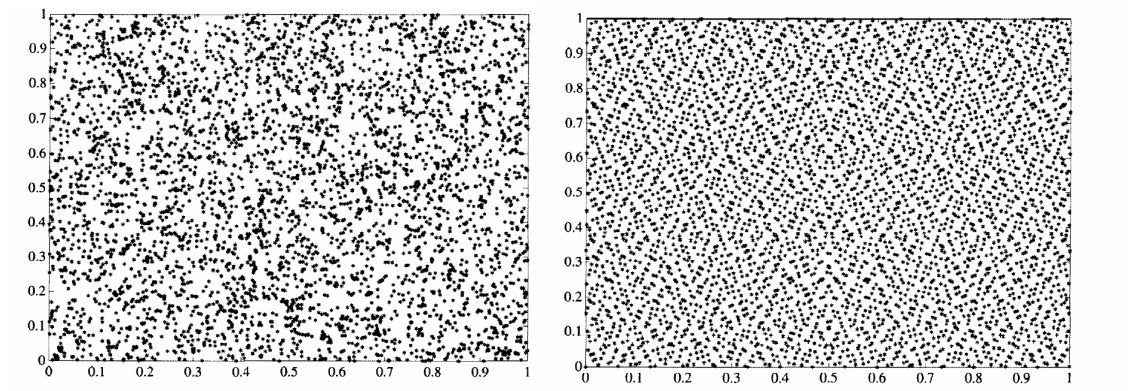

Figure 4.1: 2-dimensional projection of a pseudo-random sequence (left) and a Sobol sequence (right)

4.5 Quasi Monte Carlo Method

Now I give an introduction to quasi Monte Carlo (QMC). More detailed information may be found in Niederreiter (1992). The quasi Monte Carlo method is used in numerical integration. It approximates integrals over the closed s-dimensional unit cube $[0,1]^s$, for $s \in \mathbb{N}$. The quasi Monte Carlo estimation for

$$\theta(f) = \int_{[0,1]^s} f(u)\,du \quad \text{is} \quad \hat{\theta}_n(f) = \frac{1}{n}\sum_{i=1}^{n} f(x_i), \tag{4.5}$$

where $x_1, \ldots, x_n \in [0,1]^s$. Up to now this looks like plain Monte Carlo. The difference of these two methods lies in the sequence of points x_i. These x_i are well-chosen deterministic points such that their distribution is close to the continuous uniform distribution on $[0,1]^s$. To have a measure of how far apart a sequence of deterministic points is from uniformity, the concept of discrepancy is introduced next. It measures the deviation of the points x_i to the uniform distribution.

4.5.1 Discrepancy

Suppose the following situation. Let f be an integrable function. We want to estimate $\theta(f) = \int_{[0,1]^s} f(u)\,du$ with the quasi Monte Carlo estimate $\hat{\theta}_n(f)$ defined in (4.5). We have

$$\theta(f) \;=\; \int_{[0,1]^s} f(u)\,du \;\approx\; \frac{1}{N}\sum_{n=1}^{N} f(x_n) \;=\; \hat{\theta}_N(f). \tag{4.6}$$

with $x_1, \ldots, x_N \in [0,1]^s$. For approximating $\theta(f)$ exactly, we would need an infinite sequence of points $x_1, x_2, \ldots \in [0,1]^s$. Suppose we have such a sequence at hand. Then we can replace the finite sequence $x_1, \ldots, x_N$ by the infinite one and get again that

$$\lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} f(x_n) = \int_{[0,1]^s} f(u)\,du. \tag{4.7}$$

We want this infinite sequence to be such that (4.7) is fulfilled for a whole class of functions f, e.g. all continuous functions. Uniform distributed sequences will accomplish this. So the advisable points $x_1, \ldots, x_N$ for (4.6) should be chosen such that the discrete probability distribution is close to the continuous distribution $U[0,1]^s$. But which sequences are uniformly distributed? To get an answer, have a look at the following definitions of discrepancy, taken from Niederreiter (1992).

Let P be a point set consisting of $x_1, \ldots, x_N \in [0,1]^s$. For an arbitrary subset B of $[0,1]^s$ let $A(B;P)$ be the number of x_n with $1 \le n \le N$ for which $x_n \in B$. So $A(B;P)$ is defined as

$$A(B;P) = \#\{x_n :\; x_n \in B,\, 1 \le n \le N\}$$

Let $\mathcal{B}$ be a nonempty family of Lebesgue-measurable subsets of $[0,1]^s$. A general definition of the discrepancy of the point set P is then given by

$$D_N(\mathcal{B};P) = \sup_{B\in\mathcal{B}} \left| \frac{A(B;P)}{N} - \lambda_s(B) \right|.$$

Definition 17. *Let $\mathcal{J}^*$ be the family of all subintervals of $I^s = [0,1)^s$ of the form $[0,u] = \prod_{i=1}^{s}[0,u_i]$. Then the star discrepancy $D_N^*(P) = D_N^*(x_1,\ldots,x_N)$ of the point set P is defined by $D_N^*(P) = D_N(\mathcal{J}^*;P)$.*

Definition 18. *Let $\mathcal{J}$ be the family of all subintervals of I^s of the form $[u,v] = \prod_{i=1}^{s}[u_i,v_i]$. Then the (extreme) discrepancy $D_N(P) = D_N(x_1,\ldots,x_N)$ of the point set P is defined by $D_N(P) = D_N(\mathcal{J};P)$.*

The star discrepancy D_N^* and the discrepancy D_N are associated with each other in the following way:

$$D_N^*(P) \le D_N(P) \le 2^s D_N^*(P).$$

These discrepancies define an L^∞ norm on the elements in $\mathcal{J}^*$ and $\mathcal{J}$. The next discrepancy is an L^2 norm on the elements in $\mathcal{J}^*$ and is given by Halton and Zaremba (1969). The practical experiences with these discrepancies have shown that the extreme discrepancy D_N and T_N^* are ordinarily of the same size.

Definition 19. *The discrepancy T_N^* of the point set P is defined by*

$$T_N^*(P) = \left(\int_{[0,1]^s} \left(\frac{A([0,u];P)}{N} - \lambda_s\left([0,u]\right) \right)^2 du \right)^{1/2}.$$

Definition 20 (Uniformly distributed sequence). *A* uniformly distributed sequence *is one for which $D_N^* \to 0$ as $N \to \infty$.*

The next theorem is the famous Koksma-Hlawka Theorem and gives an upper bound for the integration error $\epsilon = |\hat{\theta}_n - \theta|$. For that we need the following definition, taken from Caflisch (1998).

Definition 21. *Define the variation (in the Hardy-Krause sense) of f, a differentiable function of a single variable, as*

$$V_{HK} = \int_0^1 \left| \frac{df}{dt} \right| dt.$$

In s dimensions, the variation is defined as

$$V_{HK} = \int_{I^s} \left| \frac{\partial^s f}{\partial t_1 \cdots \partial t_s} \right| dt_1 \cdots dt_s + \sum_{i=1}^{s} V_{HK}\left(f_1^{(i)}\right)$$

in which $f_1^{(i)}$ is the restriction of the function f to the boundary $x_i = 1$.

For properties of V_{HK} and other multidimensional variation measures see Owen (2005).

Theorem 16 (Koksma-Hlawka Theorem). *For any sequence $\{x_N\}$ and any function f with variation $V_{HK}(f) < \infty$, the integration error $\epsilon = |\hat{\theta}(f) - \theta(f)|$ is bounded as*

$$\epsilon \leq D_N^* \, V_{HK}(f). \tag{4.8}$$

The upper bound on the integration error in (4.8) consists of the discrepancy, which measures the quality of the points and the variation which measures the smoothness of

the integrand. As pointed out later, one can construct sequences of points $x_1, \ldots, x_N$ where $D_N^* = \mathcal{O}(N^{-(1-\epsilon)})$ holds for any $\epsilon > 0$. So if the variation is finite the Koksma-Hlawka Theorem indicates that quasi Monte Carlo performs better than plain Monte Carlo. Anyway, computing the star discrepancy is very hard, see Gnewuch et al. (2009), and computing $V_{\mathrm{HK}}(f)$ is harder than integrating f itself and so the upper bound in (4.8) is impractical for estimating the integration error ϵ.

Practical applications have shown that the discrepancy of quasi random sequences is a more significant measure for QMC performance than the variation. The next result by Woźniakowski (1991) is independent of the variation and characterizes the mean square error $E(\epsilon(f)^2)^{1/2}$ of quasi Monte Carlo integration.

Theorem 17. *It holds that*

$$E(\epsilon(f)^2)^{1/2} = T_N^*(f), \tag{4.9}$$

where the expectation is taken with respect to the function f which is distributed according to the Brownian sheet measure.

Caflisch (1998) defines the Brownian sheet measure as a measure on a function space. It is a natural generalization of a Brownian motion in one dimension to multi-dimensional time. A proof of Woźniakowski's identity (4.9) can be found in Caflisch (1998).

After defining the discrepancy and determining some error bounds for QMC we go on with some further properties of low-discrepancy sequences.

4.5.2 Quasi-random numbers

The inventors of quasi random sequences where number theorists who did research on the uniformity properties of numerical sequences, see Kuipers and Niederreiter (1974).

Let $x_1, x_2, \ldots, x_N \in [0,1]^s$ be a sequence of points. For the moment let us consider the one-dimensional case where $s = 1$ and let us fix N. With these restrictions in mind, it is easy to determine the minimum of the discrepancy $D_N(x_1, \ldots, x_N)$ and the star discrepancy $D_N^*(x_1, \ldots, x_N)$ and to give sequences for which these minima are attained, see Niederreiter (1992). For the star discrepancy it holds that

$$D_N^*(x_1, \ldots, x_N) \geq \frac{1}{2N},$$

and equality holds for the sequence $x_n = (2n-1)/2N$ for $1 \leq n \leq N$. An analogous

result can be shown for the discrepancy. It holds that

$$D_N(x_1, \ldots, x_N) \geq \frac{1}{N},$$

and equality holds for the sequence $x_n = (2n-1)/2N$ for $1 \leq n \leq N$.

In words this means that for fixed N it is possible to reach a discrepancy of order of magnitude $\mathcal{O}(N^{-1})$ for point sets $x_1, \ldots, x_N \in [0,1]$. In contrast to this, it can be shown that there exists no sequence of points in $[0,1]$ where $D_N(S) = \mathcal{O}(N^{-1})$ is true for all $N \geq 1$. So loosen the condition of fixing N at the beginning impairs the magnitude of the discrepancy. Schmidt (1972) gives a more precise result which states that for any point-sequence S in $[0,1]$ it holds that

$$D_N(S) \geq cN^{-1} \log N$$

for infinitely many N and an absolute constant c. This implies that the best possible discrepancy for a sequence S has order of magnitude $D_N(S) = \mathcal{O}(N^{-1} \log N)$. Therefore quasi-random sequences are defined in the following way.

Definition 22. *A sequence of points $x_1, x_2, \ldots, x_N \in [0,1]^s$ is called* quasi-random *or a* low-discrepancy sequence *if*

$$D_N \leq c(\log N)^k N^{-1}$$

where c and k are constants independent of N, but may be dependent on the dimension s. As a matter of fact sequences can be constructed in a way such that $k = s$.

The next step is the explicit construction of low-discrepancy sequences. A very clear and popular quasi-random sequence is the van der Corput sequence in one dimension first published by van der Corput (1935). Even though the one-dimensional case isn't of great interest in practice, the van der Corput sequence builds a basis for other sequences in higher dimension, e.g. the Halton sequence, introduced by Halton (1960).

The van der Corput sequence in base $b \geq 2$ is constructed by writing out $n \in \mathbb{N}$ in base b.

$$n = a_m a_{m-1} \ldots a_1 a_0 \qquad \text{(base } b\text{)}.$$

The nth point x_n of the sequence is then obtained by a symmetric reflection of the bits

of n around the decimal point, so

$$x_n = 0.a_0a_1 \dots a_{m-1}a_m \qquad (\text{base } b).$$

Halton sequences generalize the one-dimensional van der Corput sequences. A Halton sequence in a given dimension s is constructed as follows. Let $p_1, \dots, p_s$ be the bases for the sequence where the bases are chosen to be pairwise relatively prime. Write out $n \in \mathbb{N}$ in base p_k for $1 \leq k \leq s$. Let $\psi_p(n)$ be the symmetric reflection of the expansion of n in base p around the decimal point. So, if $n = a_m a_{m-1} \dots a_1 a_0 \ \ (\text{base } p)$, then

$$\psi_p(n) = \sum_{i=0}^{m} a_i(n) p^{-i-1}.$$

Now the nth point x_n of the Halton sequence in dimension s is given by

$$x_n = (\psi_{p_1}(n), \dots, \psi_{p_s}(n)) \quad \in [0,1]^s.$$

Suppose S is a Halton sequence. We already know that the discrepancy is bounded below by

$$D_N(S) \geq cN^{-1} \log^s N.$$

An upper bound on the star discrepancy of a Halton sequence S is given in Niederreiter (1992)

$$D_N^*(S) \leq c_s N^{-1} \log^s N + \mathcal{O}(N^{-1} \log^{s-1} N).$$

for all $N \geq 2$ and a constant c_s dependent on the dimension s and also dependent on the bases $p_1, \dots, p_s$. For dimension s the constant c_s is minimal when choosing $p_1, \dots, p_s$ to be the first s prime numbers. Nevertheless, one can show that c_s grows superexponentially as $s \to \infty$, which means that this bound is practically useless for larger dimension s. For more details and the proofs of these statements see Niederreiter (1992).

Next I will give an introduction into a group of point sets and sequences, which have discrepancy bounds with much smaller constants.

4.5.3 (t, m, s)-Nets and (t, s)-Sequences

The first (t, m, s)-nets and (t, s)-sequences in base 2 where introduced by Sobol'. The general definitions for (t, m, s)-nets and (t, s)-sequences given here are from Niederreiter

(1992) but first introduced in the earlier work Niederreiter (1987).

Let $b \geq 2$ be the basis and $s \geq 1$ be the dimension and fix them.

Definition 23. *Let $E \subset [0,1)^s$ be of the form*

$$E = \prod_{i=1}^{s} \left[a_i b^{-d_i}, (a_i + 1) b^{-d_i}\right)$$

with $a_i, d_i \in \mathbb{N}$ and $0 \leq a_i \leq b^{d_i}$ for $1 \leq i \leq s$. Then E is called an elementary interval in base b.

Definition 24. *Let $0 \leq t \leq m$ be integers. A* (t, m, s)-net in base b *is a point set P consisting of b^m points in $[0,1]^s$ such that $A(E; P) = b^t$ for every elementary interval E in base b with measure b^{t-m}, i.e. the discrepancy $D_{b^m}(E, P) = 0$ for every elementary interval E in base b with measure b^{t-m}.*

Definition 25. *Let $t \in \mathbb{N}$. A sequence $x_1, x_2, \ldots$ of points with $x_i \in [0,1]^s$ is called a* (t, s)-sequence in base b *if, for all integers $k \geq 0$ and $m > t$, the point set consisting of the x_i with $kb^m \leq i < (k+1)b^m$ forms a (t, m, s)-net in base b.*

With this definitions the one-dimensional van der Corput sequence in base b explained before is a $(0, 1)$-sequence in base b.

After doing piles of calculation and proving, which is found in e.g. Niederreiter (1992), an upper bound for the star discrepancy $D_N^*(S)$ for a (t, s)-sequence S eventuates.

Theorem 18. *The star discrepancy $D_N^*(S)$ for a (t, s)-sequence S is base b satisfies*

$$N D_N^*(S) \;\leq\; C(s, b) b^t \log^s N + \mathcal{O}(b^t \log^{s-1} N) \tag{4.10}$$

for all $N \geq 2$ and a constant $C(s, b)$ depending on the dimension s and the basis b. Explicitly

$$C(s, b) = \frac{1}{s} \left(\frac{b-1}{2 \log b}\right)^s$$

if either $s = 2$ or $s = 3, 4$ and $b = 2$. Otherwise

$$C(s, b) = \frac{1}{s!} \frac{b-1}{2 \lfloor b/2 \rfloor} \left(\frac{\lfloor b/2 \rfloor}{\log b}\right)^s.$$

The bound in (4.10) shows that the (t, s)-sequences and therefore also the (t, m, s)-nets are most evenly distributed when choosing $t = 0$. The construction of those special nets

and sequences and also the general construction principles of (t, m, s)-nets and (t, s)-sequences is too time consuming to be demonstrated here and so the interested reader is referred to Niederreiter (1992).

For the moment we have collected enough information about the quasi Monte Carlo method and low discrepancy sequences. Some classic books for a deeper understanding of uniform distribution of sequences are by Kuipers and Niederreiter (1974), Hlawka (1979), Niederreiter (1992), Drmota and Tichy (1997). In the next chapter, we shall proceed by combining Markov chain Monte Carlo methods introduced in Chapter 3 with quasi random sequences. This hybrid of MCMC and quasi Monte Carlo seems obvious and reasonable; nevertheless, except for a few papers, the combination of these two methods has not become popular until recently.

5 Markov Chain Quasi Monte Carlo

5.1 Introduction

The MCMC method has an extended field of application compared to the simple Monte Carlo method. Sampling from sophisticated distributions or distribution functions which are difficult to calculate is implemented via Markov chains. The desired distributions are modeled by Markov chains which converge to their stationary distribution. After sufficiently many steps of the Markov chain the stationary distribution is approximately reached and statements about the original problem can be made. To move from one state of the Markov chain to the next the MCMC method uses random variables as help. In more detail, this means that when at state X_n to generate a proposal Y_{n+1} and to accept or reject this proposal a sequence of uniformly independent distributed random variables are called in action, see Chapter 3.

As we have seen in the last Chapter 4, quasi Monte Carlo methods can also have an enormous benefit towards plain Monte Carlo in some fields of application. Furthermore, quasi Monte Carlo gains a higher accuracy, since the sampling points used are elements of a quasi-random sequence which are more uniformly spread than ordinary uniformly and independent pseudo-random numbers. Therefore our interest lies now to combine the advantages of MCMC and quasi Monte Carlo and apply quasi-random sequences to an MCMC sampler. Our hope is that the replacement of the uniformly IID random numbers with a quasi-random sequence also brings a benefit and moreover, that the MCMC sampler remains consistent. As we will see in a little while this is true for some, though not all low-discrepancy sequences. The appropriate sequences have to satisfy a further property. In particular they have to be (weakly) completely uniformly distributed (CUD), which is defined a little further below. The elements of a (weakly) CUD sequence will be even more balanced, as its name implies.

When the IID random numbers for selecting a proposal and acceptance or rejection of the proposal are replaced by a (weakly) CUD sequence, we get a quasi Monte Carlo version of the Markov Chain Monte Carlo method. The application of these quasi-

random sequences is similar to using the whole period of a moderately sized random number generator. Therefore the random number generators should be chosen in an adequate way, such that the produced random numbers are well balanced. Further they need to be small enough such that the whole period can be used.

Just a few years ago Owen and Tribble (2005) proved that the consistency of a Metropolis-Hastings sampler is preserved, when the IID random numbers which drive the process are replaced by a CUD or a weakly CUD sequence. The proof of this consistency statement is built on a result by Čencov (1967). It gives conditions on the (weakly) CUD sequence used for the sampling algorithm, with the result that the consistency is preserved.

The replacement of IID random numbers by a quasi-random sequence for driving the sampling process of the MCMC method was titled *Markov chain quasi Monte Carlo (MCQMC)* by Owen and Tribble (2005). A crucial fact, which I want to point out at the beginning, is that the use of a quasi-random sequence in MCMC simulation is not done to accelerate the convergence to equilibrium. It's rather a question of obtaining more balanced samples by using low-discrepancy sequences and as a result of this to improve the accuracy of the simulation, similarly as with using quasi Monte Carlo instead of plain Monte Carlo.

Hereafter a listing of all past efforts of combining quasi Monte Carlo with Markov chain Monte Carlo is given. If one starts searching for literature which applies quasi MC sampling to MCMC problems it may take a while before the effort is rewarded. Although I did a lot of literature search, I want to cite the well structured literature review by Chen et al. (2009).

5.1.1 Literature review

The probably fist article which deals with this topic appears in the 1960s by Čencov (1967), followed by a paper by Sobol (1974). Both papers assume that the Markov chain has a discrete state space and that the transitions are sampled by inversion. Unfortunately, QMC methods usually bring no large performance improvements on such unsmooth problems and inversion is not a very convenient method. Čencov replaces IID samples by one long CUD sequence. Sobol uses what is conceptually an $n \times \infty$ matrix of values from the unit interval. Each row is used to make transitions until the chain returns to its starting state. Then the sampling starts using the next row. It is like deterministic regenerative sampling.

These methods were not widely cited and, until recently, were almost forgotten, prob-

ably due to the difficulty of gaining large improvements in discrete problems, and the computational awkwardness of inversion as a transition mechanism for discrete state spaces.

The next attempt is that of Liao (1998). Liao takes a set of QMC points in $[0,1]^d$ shuffles them in random order, and uses them to drive an MCMC samper. He reports 4− to 25−fold efficiency improvements, but gives no theory. An analysis of Liao's method is given in Tribble and Owen (2008) and Tribble (2007). Later Chaudhary (2004) tried a different strategy using QMC to generate balanced proposals for Metropolis-Hastings sampling, but found only small improvements and did not publish the work. Lemieux and Sidorsky (2006) report variance reduction factors ranging from about 1.5 to about 18 in some work using QMC in conjunction with the perfect sampling method of Propp and Wilson (1996). Craiu and Lemieux (2007) consider the so-called multipletry Metropolis, which is a modified Metropolis algorithm that allows larger step sizes, and find variance reductions of up to 30%, which is still not that much.

Really large benefits from combining quasi Monte Carlo and Markov chain Monte Carlo have first arise a few years ago by Owen and Tribble (2005) and later in the dissertation by Tribble (2007). Tribble's best results come from Gibbs sampling problems computing posterior means.

There is another line of research in which large improvements have been obtained by combining QMC with MCMC. This is the so-called array-RQMC method described in Lécot and Tuffin (2004), L'Ecuyer et al. (2008) and L'Ecuyer and Sanvido (2010). That method simulates numerous chains in parallel using quasi-Monte Carlo to update all the chains. It requires a complicated method to match the update variables for each step to the various evolving chains. This method has achieved variance reductions of many thousand fold on some problems from queuing and finance. Very few properties have been established for it, beyond the case of heat particles in one dimension that was considered by Morokoff and Caflisch (1993). Finally, Jim Propp's rotor-router method is a form of deterministic Markov chain sampling. It has brought large efficiency improvements for some problems on a discrete state space and has been shown to converge at better than the Monte Carlo rate on some problems. See for example Doerr and Friedrich (2009).

First I will present the method by Owen and Tribble (2005) who uses weakly CUD sequences to drive the Metropolis-Hastings sampler. This method has a huge practical advantage compared to e.g. the rotor-router model or the array-RQMC method. It only replaces the IID random numbers used as driving sequence in a typical MCMC sampler

by another sequence of numbers, namely a (weakly) CUD sequence. After outlining the MCQMC method I will explain briefly the array-RQMC method by L'Ecuyer et al. (2008).

5.1.2 Preparatory Work

The use of quasi-random sequences instead of pseudo-random numbers in the Monte Carlo method appears as an obvious idea from the beginning. Each element of the quasi-random sequence is used as a sample point. When using quasi-random points instead of IID random numbers in the Metropolis-Hastings algorithm, this may seem less reasonable. So here is how we will handle this. Recall the Metropolis-Hastings algorithm defined in Section 3.4. Random variables are needed for moving from state X_n to the next sample point X_{n+1}. When at state X_n, a proposal Y_{n+1} is generated at random. Assume that each generation of a proposal needs at most $(d-1)$ IID random variables uniformly distributed in $[0,1]$. This proposal Y_{n+1} is either accepted or rejected. This can be implemented by a simple Bernoulli experiment using another random variable in $[0,1]$. So altogether at most d random variables are needed to generate a sample from a Markov chain. Therefore this is called a d-dimensional MCMC algorithm. When sampling with another MCMC method, the way of proceeding is quite similar and one can also assume that there is a bounding number d of IID random variables, uniformly distributed in $[0,1]$, needed for moving forward one step from X_n to X_{n+1}.

Now suppose we want N samples. Then we need to run the Markov chain for N steps, and Nd uniformly distributed random variables $u_1, u_2, \ldots, u_{Nd} \in [0,1]$ are required. These random variables are the *driving sequence* of the MCMC algorithm and are stored in the *variable matrix*, in the following way:

$$\begin{pmatrix} u_1 & u_2 & \cdots & u_d \\ u_{d+1} & u_{d+2} & \cdots & u_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ u_{d(N-1)+1} & u_{d(N-1)+2} & \cdots & u_{Nd} \end{pmatrix} \tag{5.1}$$

Hereafter we will look at the rows of the variable matrix. Therefore we will define blocks of size d of the driving sequence as d-dimensional random variables. Let $1 \leq n < N$ and denote by $z_n = (u_{d(n-1)+1}, u_{d(n-1)+2}, \ldots, u_{dn})$ the d-dimensional point in $[0,1]^d$. Then z_n gives the nth row of the variable matrix. Let $\{X_n\}$ be a Markov chain. Then the nth step of the Markov chain can be constructed by the nth row of the variable

matrix and the update function ϕ as

$$X_n = \phi(X_{n-1}, z_n). \tag{5.2}$$

5.2 CUD and weakly CUD sequences

As we have learned from the previous Chapter 4 quasi-random sequences are those with discrepancy $D_n = 0$ for $n \to \infty$. If we want to use points from quasi-random sequences instead of IID points for MCMC sampling this condition will be unrewarding and a sharper condition is needed. This leads us to the definition of completely uniformly distributed sequences and weakly completely uniformly distributed sequences. The property of completely uniformly distribution of a sequence is an important property for quasi-random sequences and is a definition of randomness by Knuth (1981). The idea behind completely uniformly distributed sequences originated in Korobov (1948). I took the following definitions from Tribble (2007).

Definition 26. *A sequence $u_1, u_2, \ldots \in [0,1]$ is* completely uniformly distributed *(CUD) if for every integer $d \geq 1$, the sequence $z_1, z_2, \ldots$ of d-blocks $z_i = (u_i, \ldots, u_{i+d-1}) \in [0,1]^d$ satisfies*

$$D_n^*(z_1, \ldots, z_n) \longrightarrow 0 \quad \text{as } n \to \infty. \tag{5.3}$$

This definition holds for deterministic sequences. A similar definition can be given for random sequences.

Definition 27. *A sequence of random variables $u_1, u_2, \ldots \in [0,1]$ is* weakly completely uniformly distributed *(WCUD), if for every integer $d \geq 1$, the sequence $z_1, z_2, \ldots$ of d-blocks $z_i = (u_i, \ldots, u_{i+d-1}) \in [0,1]^d$ satisfies*

$$P\big(D_n^*(z_1, \ldots, z_n) > \epsilon\big) = 0 \quad \text{as} \quad n \to \infty \tag{5.4}$$

for every $\epsilon > 0$.

CUD sequences are defined for overlapping blocks $z_i = (u_i, \ldots, u_{i+d-1})$ of length $d \geq 1$. When sampling with MCMC for each step in the Markov chain (5.2) we consider one row of the variable matrix (5.1). Each row is a non-overlapping d-block $\hat{z}_i = (u_{d(i-1)+1}, \ldots, u_{di})$ for $i \geq 1$ of the sequence $u_1, u_2, \ldots \in [0,1]$. Čencov (1967) proved the following result:

$$
\begin{aligned}
& D_n^*(z_1, \ldots, z_n) \to 0, \\
\Longleftrightarrow & \\
& D_n^*(\hat{z}_1, \ldots, \hat{z}_n) \to 0.
\end{aligned}
\tag{5.5}
$$

Knuth (1981) proved the generalization of this result.

Lemma 8. *The sequence $u_1, u_2, \ldots \in [0,1]$ is CUD if and only if for arbitrary integers $1 \leq k \leq d$, the sequence z_i of d-tuples defined by $z_i = (u_{id-k+1}, u_{id-k+2}, \ldots, u_{id-k+d})$ satisfies*

$$
D_n^*(z_1, \ldots, z_n) \to 0 \quad \textit{for} \quad n \to \infty. \tag{5.6}
$$

An analogous result holds for WCUD sequences.

This means that if a sequence has the CUD property it is uniformly in its overlapping blocks as well as in its non-overlapping blocks.

As already mentioned the use of CUD sequences in MCMC is comparable to the use of the whole period of a random number generator, which shouldn't be too large. For more details see Niederreiter (1986). Tribble (2007) uses small versions of multiplicative congruential generators and linear feedback shift register generators. I will give an example of this later. But first we will see that using a CUD sequence in a Metropolis-Hastings algorithm leads to consistency.

5.2.1 The Consistency Theorem

For proving consistency of the Metropolis-Hastings sampler driven by a CUD sequence we assume a finite state space S and the following regularity conditions on the proposal mechanism. The next definition and the following lemma are from Tribble (2007).

Definition 28. *The proposals of a d-dimensional Metropolis-Hastings algorithm are* regular *if and only if for any two states $x_k, x_l \in S$ and time i, the set of proposals defined as*

$$
\mathcal{A}_i^{kl} := \left\{ (u_{id+1}, \ldots, u_{id+d-1}) \in [0,1] \mid Y_{i+1} = x_l \quad \textit{when} \quad X_i = x_k \right\} \tag{5.7}
$$

is Jordan measurable.

Remember that a Jordan measurable set is one whose indicator function is Riemann integrable. When at state x_k at time i, then the hypercube $[0,1]^{d-1}$ can be divided

into regions $\mathcal{A}_i^{kl}$ of $(d-1)$-dimensional random variables which will choose x_l to be the next proposal. As a result from the regularity condition, each of these $\mathcal{A}_i^{kl}$ is Jordan measurable. If the proposals are homogenous, which we assume most of the time, then the $\mathcal{A}_i^{kl}$ are the same for all i. The regularity condition can be extended from the set of proposals to the set of transitions, since unions, complements and tensor products of Jordan measurable sets are again Jordan measurable.

Lemma 9. *If regularity of proposals holds, then for any two states $x_k, x_l \in S$ and time i, the set of transitions defined as*

$$\mathcal{E}_i^{kl} := \left\{(u_{id+1}, \ldots, u_{id+d}) \in [0,1] \mid X_{i+1} = x_l \quad \textit{when} \quad X_i = x_k\right\} \tag{5.8}$$

is Jordan measurable.

Note that these regularity conditions aren't a restriction since they are satisfied by all feasible Markov chain Monte Carlo methods anyway.

The main theorem of this section provides information about when CUD sequences preserve the consistency of a Metropolis-Hastings sampler. Remember that consistency holds if for any state x in the state space S and for any starting state $X_0 = x_0$

$$\pi_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{X_i = x} \;\to\; \pi(x) \quad \text{for} \quad n \to \infty, \tag{5.9}$$

where π is the stationary limit distribution of the Markov chain $\{X_n\}$. Weak consistency holds if for any state $x \in S$ and for any starting state $X_0 = x_0$

$$P(|\pi_n(x) - \pi(x)| > \epsilon : \; X_0 = x_0) \to 0 \quad \text{for} \quad n \to \infty. \tag{5.10}$$

holds for any arbitrary but fix $\epsilon > 0$.

The following theorem is a generalization of a result of Čencov. This generalization and also the proof of it is given in Owen and Tribble (2005) and Tribble (2007).

Theorem 19. *Suppose $S = \{x_1, \ldots, x_K\}$ is a finite state space and a sequence $u_1, u_2, \ldots \in [0,1]$ is used to run a Metropolis-Hastings sampler with regular homogenous proposals. Assume the resulting sample is weakly consistent if the u_i are IID $U[0,1]$, such that (5.10) holds. If the u_i form a CUD sequence, consistency holds.*

If the u_i are replaced by a weakly CUD sequence, weak consistency holds.

Proof. Let $u_1, u_2, \ldots \in [0,1]$ be a completely uniformly distributed sequence. Given $X_0 = x_0$ the empirical measure $\pi_n(x)$ is a function of $u_1, \ldots, u_{nd}$. Next we look at bad regions in $[0,1]^{nd}$ where the empirical measure of a state x is not close to its equilibrium distribution $\pi(x)$. Therefore we define for each starting state $x_k \in S$ and target state $x_l \in S$ sets of consecutive u_i for which this bad convergence is seen.

$$\tau_{kl}^n(\epsilon) = \big\{(u_1, \ldots, u_{nd}) : |\pi_n(x_l) - \pi(x_l)| > \epsilon \quad \text{when} \quad X_0 = x_k\big\} \tag{5.11}$$

where $\epsilon > 0$ is a given tolerance limit. Since the proposals are Jordan measurable, these regions are also Jordan measurable as they are the finite unions of Jordan measurable sets.

The volume of $\tau_{kl}^n(\epsilon)$ is the probability under IID sampling that $|\pi_n(x_l) - \pi(x_l)| > \epsilon$ when $X_0 = x_k$. Since we have assumed weak consistency when using independent and uniformly distributed $u_i \in [0,1]$ it follows that for any $x_k, x_l \in S$

$$\text{Vol}(\tau_{kl}^n(\epsilon)) \longrightarrow 0 \quad \text{as} \quad n \to \infty.$$

Suppose m large enough such that for all $x_k, x_l \in S$

$$\text{Vol}(\tau_{kl}^m(\epsilon)) < \frac{\epsilon}{K}$$

where K is the finite number of states.

Define

$$\tau_l^m(\epsilon) = \bigcup_{x_k \in S} \tau_{kl}^m(\epsilon)$$

to be the region which samples x_l poorly for at least one starting state x_k. This set is also Jordan measurable and $\text{Vol}(\tau_j^m(\epsilon)) < \epsilon$. For $x_l \in S$, $i = 1, \ldots, n$ and an integer m define an indicator $Z_i \in 0, 1$ in the following way

$$Z_i = \mathbb{1}_{\tau_l^m(\epsilon)}(u_{(i-1)d+1}, \ldots, u_{(i-1)d+md}).$$

Next we define the empirical measure $\pi_i^m(x_l)$, where m steps of sampling are done and which corresponds to $(u_{(i-1)d+1}, \ldots, u_{(i-1)d+md})$, as

$$\pi_i^m(x_l) = \frac{1}{m} \sum_{j=0}^{m-1} \mathbb{1}_{X_{i+j} = x_l}.$$

Note that $|\pi_i^m(x_l) - \pi(x_l)| < \epsilon$ whenever $Z_i = 0$. One can show that for the empirical

volume $\mathrm{Vol}_n(\tau_l^m(\epsilon))$ it follows that

$$\mathrm{Vol}_n(\tau_l^m(\epsilon)) \;=\; \frac{1}{n}\sum_{i=1}^{n} Z_i \longrightarrow \mathrm{Vol}(\tau_l^m(\epsilon)), \tag{5.12}$$

which we will need a little bit later. First we split the overall empirical measure $\pi_n(x_l)$ on n points in the following more complicated way

$$\pi_n(x_l) = \frac{1}{n}\sum_{i=1}^{n} \pi_i^m(x_l) + \frac{1}{n}\sum_{j=1}^{m-1}\left(\mathbb{1}_{X_{m-j}=x_l} - \mathbb{1}_{X_{(n+m-j)}=x_l}\right). \tag{5.13}$$

Note that the second term from (5.13) is smaller than m/n. Therefore

$$\begin{aligned}
|\pi_n(x_l) - \pi(x_l)| &< \frac{1}{n}\sum_{i=1}^{n} |\pi_i^m(x_l) - \pi(x_l)| + \frac{m}{n} \\
&\leq \frac{1}{n}\sum_{i=1}^{n} Z_i |\pi_i^m(x_l) - \pi(x_l)| \\
&\quad + \frac{1}{n}\sum_{i=1}^{n} (1 - Z_i)|\pi_i^m(x_l) - \pi(x_l)| + \frac{m}{n} \\
&\leq \frac{1}{n}\sum_{i=1}^{n} Z_i + \epsilon + \frac{m}{n} \\
&\to \mathrm{Vol}(\tau_l^m(\epsilon)) + \epsilon \quad \text{as} \quad n \to \infty \\
&\leq 2\epsilon.
\end{aligned} \tag{5.14}$$

As ϵ is arbitrary, (5.9) proves our result for completely uniformly distributed u_i. If $u_1, u_2, \ldots$ is a WCUD sequence, the third inequality in (5.14) still holds in probability. Also the result (5.12) about the empirical volume still holds for u_i WCUD, but the convergence is in probability, so that

$$\frac{1}{n}\sum_{i=1}^{n} Z_i \xrightarrow{P} \mathrm{Vol}(\tau_l^m(\epsilon)). \tag{5.15}$$

Now let $n > m/\epsilon$. Then it holds that

$$P(|\pi_n(x_l) - \pi(x_l)| > 3\epsilon) < P(\frac{1}{n}\sum_{i=1}^{n} Z_i > \epsilon) \to 0$$

and so (5.10) is true for weakly CUD sequences u_i. □

Maybe you have noticed that the consistency theorem, Theorem 19, does not explicitly make assumptions about the aperiodicity or the irreducibility of the Markov chain and you wonder if those properties wouldn't be necessary for the Metropolis-Hastings sampler. They are indeed very important, but they are already embedded in the assumption that the samples are weakly consistent (5.10). So if the used Metropolis-Hastings algorithm produces consistent samples for an uniformly IID driving sequence, the use of a completely uniformly distributed sequence or a weakly CUD sequence in the Metropolis-Hastings sampler won't change anything about the consistency. The task after choosing a suitable proposal mechanism for our Metropolis-Hastings sampler is to find a completely uniformly distributed sequence or a weakly CUD sequence to run the MCQMC sampler smoothly. Therefore the next section discusses CUD and WCUD sequences more deeply and gives some broader definitions for these sequences, so-called triangular arrays.

5.2.2 Weakly CUD Triangular Arrays

In this section I will define extensions of (weakly) CUD sequences and give some basic characteristics of them which will have a relevance when doing MCMC sampling with them. Those extensions will be triangular arrays which contain (weakly) CUD sequences. In Theorem 19 we assume an infinite driving sequence $u_1, u_2, \ldots$. But actually, when sampling on a computer, the driving sequence will be of finite length N.

We assume that the simulation of a Markov chain consumes d 1-dimensional random variables per transition. If we want to run the simulation for n_1 times we therefore need a driving sequence of length $N_1 = dn_1$. For getting more than n_1 samples we need to renew the driving sequence. Suppose the new sequence has length $N_2 > N_1$. In general it cannot be assumed that the shorter driving sequence is nested in the longer one. Therefore we need to consider a concept that handles this issue in the right way.

Let $N \in \mathcal{N}$ where $\mathcal{N}$ is a non-random set of infinite size containing positive integers. We define a triangular array with elements $u_{N,1}, \ldots, u_{N,N} \in [0,1]$ in the Nth row. The points in row $N = dn$ will generate π_n. The (weak) consistency conditions now look as follows. For all $x_i \in E$, the $\pi_n(x_i)$ (weakly) converges to $\pi(x_i)$ for $n = \lfloor N/d \rfloor$ and N going through the elements $N \in \mathcal{N}$ (write $N \to \infty$).

The following definition of (weakly) completely uniformly distributed triangular array is taken from Tribble and Owen (2008).

Definition 29. *Let $u_{N,1}, \ldots, u_{N,N} \in [0,1]$ for an infinite set of positive integers $N \in \mathcal{N}$.*

Suppose that $N \to \infty$ through the values in $\mathcal{N}$, such that

$$D^*_{N-d+1}((u_{N,1},\ldots,u_{N,d}),\ldots,(u_{N,N-d+1},\ldots,u_{N,N})) \to 0 \tag{5.16}$$

holds for any $d \geq 1$. Then the triangular array $(u_{N,i})$ is said to be completely uniformly distributed. If the $u_{N,i}$ are random and

$$P(D^*_{N-d+1}((u_{N,1},\ldots,u_{N,d}),\ldots,(u_{N,N-d+1},\ldots,u_{N,N})) > \epsilon) \to 0 \tag{5.17}$$

as $N \to \infty$ holds for all integers $d \geq 1$ and all $\epsilon > 0$, then the triangular array $(u_{N,i})$ is called weakly CUD.

Note that this definition of a (weakly) CUD triangular array broadens the general definition of (weakly) CUD sequences. Let $u_1, u_2, \ldots \in [0,1]$ be a (weakly) CUD sequence. Define a triangular array by taking $u_{N,i} = u_i$, for all rows $N \geq 1$. Then this triangular array is also (weakly) CUD.

The next theorem is the generalization of the consistency theorem, Theorem 19, for (weakly) CUD triangular arrays and is taken from Tribble and Owen (2008).

Theorem 20. *Let S be a finite state space for the Markov chain $\{X_n\}$. For $n \geq 0$ let Y_{n+1} be a regular proposal generated from X_n by $(u_{dn+1},\ldots,u_{dn+d-1})$. X_{n+1} is determined by acceptance or rejection of the proposal Y_{n+1} decided by u_{dn}. Suppose that these transitions are weakly consistent for uniformly IID random variables u_i. Each transition consumes d elements of the u_i. If the u_i are replaced by elements $u_{N,i}$ of a CUD triangular array then*

$$\lim_{n\to\infty} \pi_n(x_i) = \pi(x_i) \quad \textit{for all} \quad x_i \in S, \tag{5.18}$$

for $n = \lfloor N/d \rfloor$. If the u_i are replaced by elements $u_{N,i}$ of a WCUD triangular array then

$$\lim_{n\to\infty} P(|\pi_n(x_i) - \pi(x_i)| > \epsilon) = 0 \quad \textit{for all} \quad x_i \in S. \tag{5.19}$$

The proof of this theorem is very similar to the proof of Theorem 19. A sketch of it is given in Tribble and Owen (2008).

To investigate the behavior of a sequence in terms of discrepancy, it may be useful to study the local behavior first. Therefore we define the local discrepancy of a sequence.

Definition 30. *Given n points $z_1, \ldots, z_n \in [0,1]^d$ the* local discrepancy function *is defined as*

$$\delta_n(z) = \delta_n(z; z_1, \ldots, z_n) = \Big| \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{z_i \in [0,z]} - \lambda_s([0,z]) \Big|, \tag{5.20}$$

for each $z \in [0,1]^d$.

With this definition in mind we go on and define a pointwise (weakly) CUD property for triangular arrays.

Definition 31. *The triangular array $u_{N,i} \in [0,1]$ is* pointwise CUD*, if*

$$\lim_{N \to \infty} \delta_{N-d+1}(z; (u_{N,1}, \ldots, u_{N,d}), \ldots, (u_{N,N-d+1}, \ldots, u_{N,N})) = 0 \tag{5.21}$$

holds for every integer $d \geq 1$ and every $z \in [0,1]^d$. The triangular array of random variables $u_{N,i} \in [0,1]$ is pointwise weakly CUD*, if*

$$\lim_{N \to \infty} P(\delta_{N-d+1}(z; (u_{N,1}, \ldots, u_{N,d}), \ldots, (u_{N,N-d+1}, \ldots, u_{N,N})) > \epsilon) = 0 \tag{5.22}$$

holds for every integer $d \geq 1$ and every $z \in [0,1]^d$ and every $\epsilon > 0$.

Fortunately, as we will see next, it turns out that the pointwise (weak) CUD property implies the (weak) CUD property.

Lemma 10. *If $u_{N,i}$ are pointwise CUD then they are CUD. If the $u_{N,i}$ are pointwise WCUD then they are WCUD.*

A proof of this result appears in Tribble and Owen (2008).

Up to now the given (weakly) CUD characteristics hold for consecutive blocks of size d which overlap for $d > 1$. For our circumstances it would now be helpful if these characteristics remained true for non-overlapping blocks, since e.g. when doing MCMC sampling such non-overlapping blocks are of interest. The next theorem, which might not seem reasonable at first, shows this property for triangular arrays.

Theorem 21. *Let $\mathcal{N}$ and $\mathcal{D}$ be infinite sets of nonnegative integers. Let $N \in \mathcal{N}$ be the length of the sequence u_i and $d \in \mathcal{D}$ the dimension of the MCMC sampler. Suppose that a triangular array satisfies for any dimension d, number of samples $n = \lfloor N/d \rfloor$ and $\epsilon > 0$ that*

$$P\Big(D_n^*\big((u_{N,1}, \ldots, u_{N,d}), (u_{N,d+1}, \ldots, u_{N,2d}), \ldots, (u_{N,(n-1)d+1}, \ldots, u_{N,nd})\big) > \epsilon \Big) \to 0, \tag{5.23}$$

where $N \to \infty$. Then the triangular array is WCUD.

This result is taken from Tribble (2007). A proof of it is given in Tribble and Owen (2008).

In the sequel I will give a construction rule for triangular arrays by using lattices. Furthermore an upper bound for the discrepancy of those lattice points is given. For appropriately chosen input parameters this bound won't contain constants which grow exponentially in the dimension of the points, as it was the case for Halton sequences in Chapter 4.

5.2.3 Lattice Constructions

The following construction of completely uniformly distributed triangular arrays goes back to a result by Niederreiter (1977) and uses lattice points.

For a prime number N let $u_0 = 1/N$ and u_i for $i > 0$ be $u_i = au_{i-1}/N \mod 1$ where a is a primitive root modulo N. Denote $x_i = (u_i, \ldots, u_{i+s-1})$ for $s \geq 2$ and $1 \leq i < N-1$. This lattice construction is also known as Korobov lattice. With this setup Niederreiter (1977) proved the following upper bound on the discrepancy.

Lemma 11. *For any prime N and any given dimension $s \geq 2$, there exists a primitive root $a \mod N$ such that the discrepancy D_N of the associated sequence $x_1, \ldots, x_{N-1} \in [0,1]^s$ satisfies*

$$D_{N-1}(x_1, \ldots, x_{N-1}) < \frac{1}{N-1}\left(1 + \frac{(N-2)(s-1)}{\phi(N-1)}\right)\left(\frac{2}{\pi}\log(N) + \frac{7}{5}\right)^s. \tag{5.24}$$

Note that $\phi(N)$ denotes the Euler totient function which counts the number of positive integers less than or equal to N that are relatively prime to N. One can show that for an $N_0 < \infty$ and $N \geq N_0$

$$\phi(N) >> \frac{N}{\log\log N}. \tag{5.25}$$

Therefore there exist a constant $A < \infty$ which do not have to grow exponentially in s and an $N_0 < \infty$ such that for $N \geq N_0$

$$D_{N-1} = (x_1, \ldots, x_{N-1}) < \frac{A}{N} s \log\log N \log^s N \tag{5.26}$$

holds uniformly in $s \geq 1$.

The next theorem gives a construction rule for completely uniformly distributed triangular arrays and appears in Tribble and Owen (2008).

Theorem 22. *Let $\mathcal{N}$ be an infinite set of prime numbers. Let $s(N)$ be a nondecreasing integer function of $N \in \mathcal{N}$ satisfying $s(N) = o((\log N/\log\log N)^\alpha)$ for some positive $\alpha < 1$. For each $N \in \mathcal{N}$ let $a(N)$be a primitive root modulo N for which (5.24) holds. Form a triangular array via $u_{N,1} = a(N)/N \mod 1$ and $u_{N,i} = a(N)u_{N,i-1}/N \mod 1$ for $i = 2, \ldots, N-1$. Then the triangular array $(u_{N,i})$ is CUD.*

One can show that if an s-tuple of points is prepended to an s-dimensional CUD sequence of overlapping s-tuples the CUD property remains valid. Also, when shifting a CUD sequence by a finite amount the CUD property remains valid, see Čencov (1967). These features are called into action in practice, as you will see later, when examples are presented in Section 5.3.

As already mentioned, the use of a CUD sequence as driving sequence for a MCMC sampler can be compared with the usage of a whole period of a pseudo-random number generator. The generator which fulfills the above conditions of the Korobov lattice is also known as multiplicative congruential generator (MCG). To randomize the lattice points, generally a Cranley-Patterson rotation by Cranley and Patterson (1976) is implemented.

Definition 32. *Let $a \in [0,1]^s$ and U uniformly distributed by $U[0,1]^s$. A Cranley-Patterson rotation of a is defined as $x = a + U \mod 1$ (which is done componentwise).*

So no matter what a was, after rotating a the result x is uniformly distributed on $[0,1]^s$.

Each row of the variable matrix (5.1) is randomized by a Cranley-Patterson rotation. Note that when doing a Cranley-Patterson rotation the lattice points are randomized but the spacing between the points is preserved. This is because doing a Cranley-Patterson rotation on each row of the variable matrix is equivalent to rotating each column independently of the others by a one-dimensional uniformly distributed random variable.

Furthermore it can be shown that doing a Cranley-Patterson rotation on a (weakly) CUD triangular array, the outcome is again a (weakly) CUD triangular array, see Tribble and Owen (2008).

Tribble and Owen (2008) and Tribble (2007) study Liao's proposal, see Liao (1998), which is another method for producing a randomized weakly CUD driving sequence for the MCMC computation.

5.2.4 Liao's Method

Suppose $a_1, \ldots, a_n \in [0,1]^s$ to be s-dimensional quasi Monte Carlo points. Let τ be a random permutation of the integers $\{1, 2, \ldots, n\}$. Denote by w_i the permuted low-

discrepancy points $a_{\tau(i)}$. The driving sequence for the MCMC simulation is built by a vector $u = u_1, \ldots, u_{ns}$, containing all the w_i, in a way that the one-dimensional point u_i comes from w_k where $k = \lceil i/s \rceil$, specifically it is the $i - s(k-1)$th component of w_k, so

$$u_i = w_k^{(i-s(k-1))}.$$

Suppose the MCMC sampler needs d points of the driving sequence for one transition of the Markov chain and note that it is not necessary to use $s = d$. Regroup the u_i into d-dimensional blocks $z_1, \ldots, z_m \in [0,1]^d$ where $m = \lfloor sn/d \rfloor$, specifically

$$z^{(i)} = (u_{(i-1)d+1}, \ldots, u_{id}. \tag{5.27}$$

The next result seems obvious and is essential for our reasons of finding weakly CUD sequences. It is cited from Tribble (2007) and its proof appears in Tribble and Owen (2008).

Theorem 23. *Suppose D_n^* is the s-dimensional star discrepancy of the sequence $a_1, \ldots, a_n \in [0,1]^s$. Then for arbitrary dimension $d \geq 1$, the sequence $z_1, \ldots, z_m \in [0,1]^d$ which comes from the permuted and regrouped sequence $a_1, \ldots, a_n \in [0,1]^s$ satisfies the following. For arbitrary $z \in [0,1]^d$ and $\epsilon > 0$ it holds*

$$P\Big(\delta_m(z; z^{(1)}, \ldots, z^{(m)}) > \epsilon\Big) = \mathcal{O}(n^{-1} + D_n^*). \tag{5.28}$$

The conclusion of Theorem 23 is the following.

Corollary 2. *Weak consistency for Markov chain quasi Monte Carlo method holds when the driving sequence is generated by Liao's method.*

Proof. Apply that weakly CUD follows from pointwise weakly CUD, Lemma 10, to Theorem 23. So Liao's proposal generates weakly CUD points and with Theorem 20 weak consistency holds. □

5.3 Example of a Bayesian Model of a Pump System

This example already originated in Liao (1998) and its data come from Gelfand and Smith (1990). Owen and Tribble (2005) and Tribble (2007) took this example up again.

It contains 10 pumps and records the number of failures s_k of each pump k in $t_k \times 1000$ hours. These failures occur according to independent Poisson processes with failure rates $\lambda_1, \ldots, \lambda_{10}$, so

$$P_{\lambda_k t_k}(X = m) = \frac{(\lambda_k t_k)^m}{m!} \exp(-\lambda_k t_k).$$

The λ_k have a Gamma(α, β) distribution where $\alpha = 1.802$ is the shape parameter and β the scale parameter with Gamma(γ, δ) prior distribution where $\gamma = 0.1$ is the shape parameter and $\delta = 1$ is the scale parameter. A table of the recorded data of the number of failures s_k at time t_k for pump k is given in Gelfand and Smith (1990) and Tribble (2007).

They stated that for a rate λ, the number of failures at time t is Poisson(λt) distributed. So the distribution of λ_k given β and the data are independent of the other λ values and is Gamma$(\alpha + s_k, \beta + t_k)$ distributed. The distribution of β given all the λ values is independent of the data and has a Gamma$(\gamma + 10\alpha, \delta + \sum \lambda_k)$ distribution. These conditional distributions are used to run a Gibbs sampler whose values converge to the joint posterior distribution. For the Bayesian model, which we consider here, we are interested in the posterior distributions of the λ_k and β and moreover in the construction of the estimates of these parameters. For the joint posterior distribution π of the λ_k and β, the Monte Carlo estimates of $E_\pi(\lambda_k)$ and $E_\pi(\beta)$ will be the parameter estimates we are interested in.

We want to determine the square error of the sample means from the Gibbs sampler to estimate the true average values of the parameters. The simulations were done by Tribble (2007) in the programming language R.

They did 100 replications of the simulations of size approximately 2^{10} and 2^{14} using pseudorandom numbers from the Mersenne Twister, lattice points randomly permuted via Liao's method and the multiplicative congruential generator. The MCG are of size $N = 1021$ with primitive root 65 and $N = 16381$ with primitive root 665. A table of these generators of different sizes with a good lattice structure is given in L'Ecuyer (1999). For randomization a Cranley-Patterson rotation was applied to all 11-tupel of the driving sequence $u_{11i-10}, \ldots, u_{11i}$ for $1 \leq i \leq N$.

To estimate the variance for the MCQMC method, a number r of replications of the sampling procedure are done. Producing r independent copies of the IID sampler gives the empirical variance which is used as the unbiased variance estimator. The following tables show the sample variances of these 100 posterior mean estimates.

$N \approx 2^{10}$			
	IID	Liao	MCG
λ_1	6.21e-07	3.72e-09	3.79e-09
λ_2	9.21e-06	4.88e-08	4.86e-08
λ_3	1.89e-06	8.23e-09	7.86e-09
λ_4	1.22e-06	4.13e-09	5.52e-09
λ_5	9.00e-05	9.02e-07	7.69e-07
λ_6	1.63e-05	1.05e-07	7.76e-08
λ_7	3.19e-04	1.17e-05	9.34e-06
λ_8	4.14e-04	1.37e-05	1.99e-05
λ_9	3.74e-04	9.34e-06	3.92e-06
λ_{10}	1.61e-04	1.35e-06	9.99e-07
β	9.00e-04	1.37e-05	1.04e-05

$N \approx 2^{14}$			
	IID	Liao	MCG
λ_1	3.96e-08	2.48e-11	2.20e-11
λ_2	4.62e-07	1.01e-09	1.37e-09
λ_3	8.46e-08	5.81e-11	4.67e-11
λ_4	6.95e-08	2.30e-11	2.67e-11
λ_5	5.44e-06	3.34e-08	6.35e-09
λ_6	1.02e-06	1.02e-09	6.96e-10
λ_7	2.18e-05	7.57e-07	3.73e-08
λ_8	2.65e-05	7.46e-07	5.33e-08
λ_9	3.13e-05	4.63e-07	2.89e-08
λ_{10}	1.07e-05	2.52e-08	1.09e-08
β	7.04e-05	8.58e-07	5.79e-07

The next table gives the minimum and the maximum variance reduction factors (VRF) of each MCQMC sampler over the IID sampler, where the extremes are taken over all 11 parameters.

	$N \approx 2^{10}$		$N \approx 2^{14}$	
Method	min VRF	max VRF	min VRF	max VRF
Liao	27	296	29	3016
MCG	21	241	121	2603

As one may notice, the variance reduction of the MCQMC methods with respect to the IID method gets quite large and gets even larger when the sample size increases. The accurate bias of the MCQMC methods is unknown, but if one expects the true mean to be near the mean of the 100 unbiased IID samples, then it can be shown that the bias is much lower than the variance of the estimates by IID sampling and has therefore no effect on the output.

5.4 A Randomized QMC Simulation Method for Markov Chains

Once more our aim is to estimate expectations of the form

$$\theta = \int_\Omega c(x)\, d\pi(x), \tag{5.29}$$

where π is a probability measure over some measurable space $(\Omega, \mathcal{F})$, and $c : \Omega \to \mathbb{R}$ is a measurable function, often interpreted as cost function. For the estimation of the expected average cost θ by a Markov chain $\{X_i\}$ we use an estimate

$$Y = \frac{1}{\tau} \sum_{i=0}^{\tau} c_i(X_i) \quad \text{for a stopping time } \tau, \tag{5.30}$$

where c_i is the cost function at step i.

When simulating θ by a Markov chain where at each move state-dependent costs $c_i(X_i)$ are paid, the forthcoming technique gives a low-variance unbiased estimator for the expectation θ. Furthermore it turns out that this method is even more effective if there is a natural order on the states which comes along with the cost function.

The previous randomized QMC (RQMC) method for Markov chains presented at the beginning of this chapter uses randomized low-discrepancy point sets as driving sequence with the consequence of variance reductions compared to the classical IID version. The new RQMC method presented here, first appeared in Lécot and Tuffin (2004). It is called array RQMC and operates roughly in the following way. The method simulates n copies of the Markov chain in parallel. To generate the ith sample of each copy, a $d+1$-dimensional randomized low-discrepancy point set P' of cardinality n is used, where d

is the dimension of the Markov chain. P'_{nj} is defined in L'Ecuyer et al. (2008) as

$$P'_{nj} = \left\{ u'_{ij} = \left(\frac{i + 0.5}{n}, u_{ij} \right), \quad \text{for} \quad 1 \leq i \leq n \right\} \tag{5.31}$$

with the properties that

- $P_{nj} = \{u_{1j}, \ldots, u_{nj}\}$ is a RQMC point set in $[0,1)^d$,
- u_{ij} is a random vector uniformly distributed over $[0,1)^d$ for each i,
- P'_{nj} is highly uniform in $[0,1)^{d+1}$, in a sense that is left open for the moment.

Note that a point set $P = (u_1, \ldots, u_n)$ is called RQMC point set if the $u_i \in [0,1)^d$ are uniformly distributed and the point set P covers $[0,1)^d$ more uniformly that a set of independent random points.

The points in the set P'_{nj} are randomized independently at each step i, such that for each of the n copies of the Markov chain the probability distribution of the generated samples corresponds to the distribution in the classical MCMC method. Consequently an unbiased estimator

$$Y_n = \frac{1}{n\tau} \sum_{i=1}^{n} \sum_{j=0}^{\tau} c_j(X_{i,j}) \quad \text{with stopping time } \tau \tag{5.32}$$

for the average costs θ (5.29) results. In order that the empirical distribution of the n realizations of X_i at step i provide a better approximation of the distribution of the random variables X_i, a dependence over the n copies is imposed. This is done by resorting the n copies of the Markov chain after each step.

Now here is how the method works in more detail: Let $\{X_n\}$ be a Markov chain with state space $S \subseteq \mathbb{R}^l \cup \{\infty\}$ for some integer $l \geq 1$ and let each transition of the chain consume d random variables. Furthermore let there be a sorting function $h : S \to \mathbb{R} \cup \{\infty\}$ on the state space, which assigns a real number to each state in S with $h(\infty) = \infty$. A state x_i is said to be smaller than a state x_j if $h(x_i) < h(x_j)$ and two states are said to be h-equivalent if $h(x_i) = h(x_j)$. When choosing h appropriately, the following array RQMC algorithm will perform well.

Algorithm (Array-RQMC Algorithm by L'Ecuyer et al. (2008))
Let $\tilde{P}_n = (\tilde{u}_1, \ldots, \tilde{u}_n)$ be a d-dimensional QMC point set. Select a randomization method for $\tilde{P}_n$ such that the randomized version P_n defines a new RQMC point set P'_n as defined

in (5.31). Simulate n copies, numbered $1, \ldots, n$, of the Markov chain in parallel as follows. Set $X_{1,0} = x_0, \ldots, X_{n,0} = x_0$ and $j = 1$.
While $X_{1,j-1} < \infty$ do {

- *Randomize $\tilde{P}_n$ to attain $P_{nj} = (u_{1j}, \ldots, u_{nj})$*

  Note that P_{nj} is a new randomization of $\tilde{P}_n$, independent of all the previous randomizations.

- *Set i=1.*

- *While $i \leq n$ and $X_{i,j-1} < \infty$ do {*

  - *Update the Markov chains $X_{i,j}$ according to an update function $\phi : S \times [0,1)^d \to S$. So $X_{i,j} = \phi(X_{i,j-1}, u_{ij})$.*
  - *Set $i = i + 1$.*

  }

- *Sort the states $X_{1,j}, \ldots, X_{n,j}$ by increasing order of their values of $h(X_{i,j})$ and renumber them in this order such that $h(X_{1,j}) \leq \ldots \leq h(X_{n,j})$.*

- *Set $j = j + 1$.*

}
Return the average Y_n of the n estimators Y to get an unbiased estimator for the expected cost θ.

L'Ecuyer et al. (2008) showed that for a finite state space S the upper bound on the converge rate in the worst case is $\mathcal{O}(n^{-1} \log n)$. Under some assumptions on the state space S and the random variables u_i a better convergence rate of order of magnitude $\mathcal{O}(n^{-3/2})$ holds. This is a better convergence rate for the variance than the rate $\mathcal{O}(n^{-1})$ for the classical MCMC method, see Section 3.5.1. The elaborately proofs for the bounds of the convergence rate of the array RQMC method are given in L'Ecuyer et al. (2008). The results of the empirical experiments by L'Ecuyer et al. (2008) showed that for many examples the variance reduction is even much faster than the theoretical bound.

5.4.1 Example

The numerical examples by L'Ecuyer et al. (2008) illustrate that the variance for the array RQMC method decreases faster than in the classical RQMC methods and the standard Monte Carlo sampling. In their illustrations they compare standard MC, classical RQMC, and RQMC methods to each other. I will exhibit only one of these examples to illustrate that the new array RQMC method can reach variance reductions of several thousands in some cases.

The considered sample-sizes n are approximately $2^{10}, 2^{12}, 2^{14}, 2^{16}, 2^{18}$ and 2^{20}. For the classical RQMC methods, they use Korobov lattice rules (see Section 5.2.3) on the one hand and Sobol' nets (see Section 4.5.3) on the other hand. When taking a set P_n of Korobov lattice points, P_n is randomized by a Cranley-Patterson rotation. For the exact values of n and the associated primitive roots see L'Ecuyer et al. (2008). They also considered Korobov lattice points after applying the Baker's transformation, which transforms each coordinate u to $2u$ if $u < 1/2$ and to $2(1-u)$ if $u \geq 1/2$, after the randomization.

For the second type of driving sequence, they use a Sobol' net, which is randomized by a left (upper-triangular) matrix scrambling followed by a random digital shift. For more details about that see L'Ecuyer et al. (2008) and the references given there.

For the upcoming example the following classical random QMC were applied: a randomly shifted Korobov rule (Classical Korobov), a randomly shifted Korobov rule with Baker's transformation (Classical Korobov-Baker) and a randomized Sobol' point set (Classical Sobol).

For the new array RQMC method they use the following types of driving sequences: a $(d+1)$-dimensional Korobov lattice rule with its first coordinate skipped and randomly shifted (Array Korobov), the same Korobov rule with the random shift followed by a Baker's transformation (Array Korobov-Baker), the first n points of a randomized Sobol' sequence where the points are enumerated by order of their Gray code (Array Sobol) and the same randomized Sobol' point set but with the points enumerated in their natural order (Array Sobol no Gray). For the exact construction of these sequences see L'Ecuyer et al. (2008) and the references given there.

In the upcoming example the variance reduction factors (VRF) in comparison with standard Monte Carlo are illustrated. Those were estimated by $\mathrm{Var}Y/(n\mathrm{Var}Y_n)$, where $\mathrm{Var}Y_n$ is estimated by $m = 100$ independent copies of Y_n and $\mathrm{Var}Y$ is estimated by the empirical variance of a large number of independent copies of Y.

An $M/M/1$ Queue with $d = 1$

Assume a single-server queue. The exponential interarrival times A_j of this queue are assumed to be IID with mean 1 and the exponential service times S_j are assumed to be IID with mean $\rho < 1$. This ρ is said to be the utilization factor of the server. Our aim is the estimation of the expected average waiting time, say θ, of the first t customers. Let W_j be the waiting time of customer j, where the first customer who arrives to the empty queue has number 0. These W_js satisfy the Lindley recurrence where $W_0 = 0$ and $W_j = \max\{0, W_{j-1} + S_{j-1} - A_j\}$ for $j \geq 1$. θ is estimated by

$$Y = \frac{W_0 + \ldots + W_{t-1}}{t}. \tag{5.33}$$

Therefore we need to generate $2t$ random variables $S_0, \ldots, S_{t-1}, A_1, \ldots, A_t$. The simulation is done by a $d = 1$ dimensional Markov chain which moves one step forward each time one of these random variables is generated. So, $X_0 = W_0$, $X_1 = W_0$, $X_2 = W_1$, $X_3 = W_1 + S_1$, and so on. The dimension of the low-discrepancy points is $s = 2t$.

L'Ecuyer et al. (2008) give results for $t = 100$ customers with utilization factors $\rho = 0.2$, 0.5, and 0.8. The variance per run σ^2 for the IID case was estimated by simulating 100×2^{18} independent runs. The best estimates of θ were achieved by the array RQMC method with $n \approx 2^{20}$. These estimates are

$$\begin{aligned} &\theta = 0.04922 \quad \text{and } \sigma^2 = 0.0005393 \quad \text{for } \rho = 0.2 \\ &\theta = 0.48000 \quad \text{and } \sigma^2 = 0.06307 \quad \text{for } \rho = 0.5 \\ &\theta = 2.48004 \quad \text{and } \sigma^2 = 3.1544 \quad \text{for } \rho = 0.8. \end{aligned} \tag{5.34}$$

The table below gives the empirical VRFs of the different classical RQMC simulations and the array RQMC simulations in comparison to an IID driving sequence for the average waiting time of 100 customers in a single-server queue. The utilization factor is ρ and approximately $n \approx 2^k$ sample points were considered.

ρ		$k = 10$	$k = 12$	$k = 14$	$k = 16$	$k = 18$	$k = 20$
0.2	Classical Korobov-Baker	5	8	15	16	59	117
	Classical Sobol	1	1	3	1	13	28
	Array Korobov	18	55	49	292	850	2,169
	Array Korobov-Baker	43	159	306	991	3,168	10,590
	Array Sobol	87	282	836	3,705	10,640	47,850
	Array Sobol no Gray	46	112	276	874	2,914	7,429
0.5	Classical Korobov-Baker	10	7	13	6	14	14
	Classical Sobol	2	1	4	5	9	10
	Array Korobov	14	46	33	231	686	2,034
	Array Korobov-Baker	44	200	241	1,155	3,540	15,650
	Array Sobol	123	504	1,083	5,651	13,830	55,160
	Array Sobol no Gray	55	130	302	1,188	3,507	11,260
0.8	Classical Korobov-Baker	11	2	15	17	21	26
	Classical Sobol	3	2	4	6	10	11
	Array Korobov	15	85	33	337	727	5,119
	Array Korobov-Baker	70	463	287	2,225	10,080	75,920
	Array Sobol	370	1,281	3,240	19,730	57,290	233,100
	Array Sobol no Gray	117	288	996	4,580	13,210	48,660

As a result from this table one clearly sees that array RQMC sampling obtains conspicuous improvements not only in comparison to Monte Carlo sampling but also in comparison with classical RQMC sampling. Note that the VRF increases when the utilization factor ρ increases. For this example of a $M/M/1$ queue the VRF in classical RQMC is higher when using a Korobov sequence with Baker's transformation as driving sequence instead of a Sobol' net. Conversely, when performing array RQMC sampling the highest VRF is reached for driving sequences using a Sobol net where the points are sorted in their Gray code order.

6 Conclusion

After the introductory chapters about different Monte Carlo simulation methods, including standard Monte Carlo, quasi MC and Markov chain Monte Carlo, I presented two hybrid versions of QMC and MCMC. Those relatively new methods showed promising results in comparison to classical MCMC methods. The first result I specified was by Owen and Tribble (2005). They showed that when using quasi-random points which have the (weakly) CUD property instead of pseudo-random points as driving sequence for the Metropolis-Hastings sampler, the consistency of the sampler remains. An example illustrates that the variance of this hybrid method is always lower than for classical MCMC. The variance reduction factor reaches from about 14 to more than 200.

The second, more sophisticated method was originated by Lécot and Tuffin (2004) and revisited by L'Ecuyer et al. (2008). The method presented there is denoted by array RQMC. To perform this array RQMC method, numerous copies of a Markov chain are run in parallel, where the driving sequence is built by randomized low-discrepancy points. After each step those copies were resorted according to a sorting function which operates on the state space. An example makes it obvious that this array RQMC sampler allows variance reduction factors up to many thousands.

It should be mentioned that the performance of this method decreases if the integrand has a high variance, or the dimension of the state space gets too large and there exists no natural sorting function for the states. Nevertheless, also in those cases, the variance reductions reached for this method compared with standard MCMC are still conspicuous.

Another hybrid is presented by L'Ecuyer and Sanvido (2010), who take the basic ingredients of the array RQMC method and adapt it for the coupling from the past algorithm by Propp and Wilson (1996). They also reported performance improvements in comparison to conventional MCMC methods.

Bibliography

Albert, J. H. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *J. Amer. Statist. Assoc.*, 88(422):669–679.

Aldous, D. and Diaconis, P. (1987). Strong uniform times and finite random walks. *Adv. in Appl. Math.*, 8(1):69–97.

Barker, A. (1965). Monte carlo calculations of the radial distribution functions for a proton-electron plasma. *Australian Journal of Physics*, pages 18–119.

Behrends, E. (2000). *Introduction to Markov chains*. Advanced Lectures in Mathematics. Friedr. Vieweg & Sohn, Braunschweig. With special emphasis on rapid mixing.

Caflisch, R. E. (1998). Monte Carlo and quasi-Monte Carlo methods. In *Acta numerica, 1998*, volume 7 of *Acta Numer.*, pages 1–49. Cambridge Univ. Press, Cambridge.

Casella, G., Lavine, M., and Robert, C. P. (2001). Explaining the perfect sampler. *Amer. Statist.*, 55(4):299–305.

Čencov, N. N. (1967). Pseudo-random numbers for the simulation of Markov chains. *Ž. Vyčisl. Mat. i Mat. Fiz.*, 7:632–643.

Chaudhary, S. K. (2004). *Acceleration of Monte Carlo methods using low discrepancy sequences*. ProQuest LLC, Ann Arbor, MI. Thesis (Ph.D.)–University of California, Los Angeles.

Chen, S., Dick, J., and Owen, A. B. (2009). Consistency of markov chain quasi-monte carlo on continuous state.

Corcoran, J. N. and Tweedie, R. L. (2002). Perfect sampling from independent Metropolis-Hastings chains. *J. Statist. Plann. Inference*, 104(2):297–314.

Craiu, R. V. and Lemieux, C. (2007). Acceleration of the Multiple-Try Metropolis algorithm using antithetic and stratified sampling. *Stat. Comput.*, 17(2):109–120.

Cranley, R. and Patterson, T. N. L. (1976). Randomization of number theoretic methods for multiple integration. *SIAM J. Numer. Anal.*, 13(6):904–914.

Dick, J., Larcher, G., Pillichshammer, F., and Woźniakowski, H. (2011). Exponential convergence and tractability of multivariate integration for Korobov spaces. *Math. Comp.*, 80(274):905–930.

Doerr, B. and Friedrich, T. (2009). Deterministic random walks on the two-dimensional grid. *Combin. Probab. Comput.*, 18(1-2):123–144.

Drmota, M. and Tichy, R. F. (1997). *Sequences, discrepancies and applications*, volume 1651 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin.

Feller, W. (1968). *An introduction to probability theory and its applications. Vol. I*. Third edition. John Wiley & Sons Inc., New York.

Fill, J. A. (1998). An interruptible algorithm for perfect sampling via Markov chains. *Ann. Appl. Probab.*, 8(1):131–162.

Fill, J. A., Machida, M., Murdoch, D. J., and Rosenthal, J. S. (2000). Extension of Fill's perfect rejection sampling algorithm to general chains (extended abstract). In *Monte Carlo methods (Toronto, ON, 1998)*, volume 26 of *Fields Inst. Commun.*, pages 37–52. Amer. Math. Soc., Providence, RI.

Gaver, D. P. and O' Muircheartaigh, I. G. (1987). Robust empirical Bayes analyses of event rates. *Technometrics*, 29(1):1–15.

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *J. Amer. Statist. Assoc.*, 85(410):398–409.

Gilks, W., Richardson, S., and Spiegelhalter, D. (1996). *Markov chain Monte Carlo in practice*. Chapman & Hall/CRC.

Gnewuch, M., Srivastav, A., and Winzen, C. (2009). Finding optimal volume subintervals with k-points and calculating the star discrepancy are NP-hard problems. *J. Complexity*, 25(2):115–127.

Häggström, O. (2002). *Finite Markov chains and algorithmic applications*, volume 52 of *London Mathematical Society Student Texts*. Cambridge University Press, Cambridge.

Häggström, O. and Nelander, K. (1999). On exact simulation of Markov random fields using coupling from the past. *Scand. J. Statist.*, 26(3):395–411.

Halton, J. H. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numer. Math.*, 2:84–90.

Halton, J. H. and Zaremba, S. C. (1969). The extreme and L^2 discrepancies of some plane sets. *Monatsh. Math.*, 73:316–328.

Hastings, W. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109.

Hlawka, E. (1979). *Theorie der Gleichverteilung*. Bibliographisches Institut, Mannheim.

Hohendorff, J. M. (2005). An introduction to markov chain monte carlo. Lecture notes, University of Toronto. `http://probability.ca/jeff/ftpdir/johannes.pdf`.

Hunter, J. J. (2009). Coupling and mixing times in a Markov chain. *Linear Algebra Appl.*, 430(10):2607–2621.

König, W. (Sommersemester 2003). Stochastische algorithmen. Vorlesungsskript, Universität zu Köln. `http://www.wias-berlin.de/people/koenig/www/AlgStoch.pdf`.

Knuth, D. E. (1981). *The art of computer programming. Vol. 2*. Addison-Wesley Publishing Co., Reading, Mass., second edition. Seminumerical algorithms, Addison-Wesley Series in Computer Science and Information Processing.

Korobov, N. M. (1948). On functions with uniformly distributed fractional parts. *Doklady Akad. Nauk SSSR (N. S.)*, 62:21–22.

Kuipers, L. and Niederreiter, H. (1974). *Uniform distribution of sequences*. Wiley-Interscience [John Wiley & Sons], New York. Pure and Applied Mathematics.

Lécot, C. and Tuffin, B. (2004). Quasi-Monte Carlo methods for estimating transient measures of discrete time Markov chains. In *Monte Carlo and quasi-Monte Carlo methods 2002*, pages 329–343. Springer, Berlin.

L'Ecuyer, P. (1999). Tables of linear congruential generators of different sizes and good lattice structure. *Math. Comp.*, 68(225):249–260.

L'Ecuyer, P., Lécot, C., and Tuffin, B. (2008). A randomized quasi-monte carlo simulation method for markov chains. *Operations Research*, 56:958–975.

L'Ecuyer, P. and Sanvido, C. (2010). Coupling from the past with randomized quasi-Monte Carlo. *Math. Comput. Simulation*, 81(3):476–489.

Lemieux, C. and Sidorsky, P. (2006). Exact sampling with highly uniform point sets. *Math. Comput. Modelling*, 43(3-4):339–349.

Levin, D. A., Peres, Y., and Wilmer, E. L. (2009). *Markov chains and mixing times*. American Mathematical Society, Providence, RI. With a chapter by James G. Propp and David B. Wilson.

Liao, L. G. (1998). Variance reduction in gibbs sampler using quasi random numbers. *Journal of Computational and Graphical Statistics*, 7:253–266.

MacKay, D. J. C. (2003). *Information theory, inference and learning algorithms*. Cambridge University Press, New York.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087.

Morokoff, W. J. and Caflisch, R. E. (1993). A quasi-Monte Carlo approach to particle simulation of the heat equation. *SIAM J. Numer. Anal.*, 30(6):1558–1573.

Niederreiter, H. (1977). Pseudo-random numbers and optimal coefficients. *Advances in Math.*, 26(2):99–181.

Niederreiter, H. (1986). Multidimensional numerical integration using pseudorandom numbers. *Math. Programming Stud.*, (27):17–38. Stochastic programming 84. I.

Niederreiter, H. (1987). Point sets and sequences with small discrepancy. *Monatsh. Math.*, 104(4):273–337.

Niederreiter, H. (1992). *Random number generation and quasi-Monte Carlo methods*, volume 63 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.

Owen, A. B. (2005). Multidimensional variation for quasi-Monte Carlo. In *Contemporary multivariate analysis and design of experiments*, volume 2 of *Ser. Biostat.*, pages 49–74. World Sci. Publ., Hackensack, NJ.

Owen, A. B. (2009). Monte Carlo and quasi-Monte Carlo for statistics. In *Monte Carlo and quasi-Monte Carlo methods 2008*, pages 3–18. Springer, Berlin.

Owen, A. B. and Tribble, S. D. (2005). A quasi-Monte Carlo Metropolis algorithm. *Proc. Natl. Acad. Sci. USA*, 102(25):8844–8849 (electronic).

Propp, J. and Wilson, D. (1998). Coupling from the past: a user's guide. In *Microsurveys in discrete probability (Princeton, NJ, 1997)*, volume 41 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 181–192. Amer. Math. Soc., Providence, RI.

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

Roberts, G. O. and Rosenthal, J. S. (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statist. Sci.*, 16(4):351–367.

Roberts, G. O. and Rosenthal, J. S. (2004). General state space Markov chains and MCMC algorithms. *Probab. Surv.*, 1:20–71 (electronic).

Rosenthal, J. S. (2003). Asymptotic variance and convergence rates of nearly-periodic Markov chain Monte Carlo algorithms. *J. Amer. Statist. Assoc.*, 98(461):169–177.

Schmidt, V. (2010). Markov chains and monte-carlo simulation. Lecture notes, Ulm University. `http://www.uni-ulm.de/fileadmin/website_uni_ulm/mawi.inst.110/lehre/ss10/MCMC/Schmidt_MCMC_2010.pdf`.

Schmidt, W. M. (1972). Irregularities of distribution. VII. *Acta Arith.*, 21:45–50.

Sobol, I. M. (1974). Pseudorandom numbers for the construction of discrete Markov chains by a Monte Carlo method. *Ž. Vyčisl. Mat. i Mat. Fiz.*, 14:36–44, 266.

Tribble, S. D. (2007). *Markov chain Monte Carlo algorithms using completely uniformly distributed driving sequences*. PhD thesis, Stanford University.

Tribble, S. D. and Owen, A. B. (2008). Construction of weakly CUD sequences for MCMC sampling. *Electron. J. Stat.*, 2:634–660.

van der Corput, J. G. (1935). Verteilungsfunkionen. *Nederl. Akad. Wetensch. Proc. Ser. B*, 38:813–821.

Wilson, D. B. (2000). Layered multishift coupling for use in perfect sampling algorithms (with a primer on CFTP). In *Monte Carlo methods (Toronto, ON, 1998)*, volume 26 of *Fields Inst. Commun.*, pages 143–179. Amer. Math. Soc., Providence, RI.

Woźniakowski, H. (1991). Average case complexity of multivariate integration. *Bull. Amer. Math. Soc. (N.S.)*, 24(1):185–194.