EXACT SIMULATION USING MARKOV CHAINS

MORTEN FISMEN *

DECEMBER 1997

ABSTRACT

This reports gives a review of the new *exact simulation* algorithms using Markov chains. The first part covers the discrete case. We consider two different algorithms, Propp and Wilsons "coupling from the past" (CFTP) technique and Fills rejection sampler. The algorithms are tested on the Ising model, with and without an external field. The second part covers continuous state spaces. We present several algorithms developed by Murdoch and Green, all based on "coupling from the past". We discuss the applicability of these methods on a Bayesian analysis problem of surgical failure rates.

* ADDRESS FOR CORRESPONDENCE: Håvard Rue, Department for Mathematical Sciences, The Norwegian University of Technology and Science, 7034 Trondheim, Norway. EMAIL: havard.rue@math.ntnu.no

A note to the reader

This report is a diploma-thesis (∼master-thesis) in statistics written by Morten Fismen, during the fall 1998, at Department of Mathematical Sciences, The Norwegian University of Science and Technology, Trondheim, Norway. There have been some public interest in this work, so I therefore makes it public available as a technical report. I hope the report will be of use for the reader with interest, but less background, in exact/perfect simulations techniques using Markov chains.

Please keep in mind that our diploma-thesis consists of 4 month of work with an *absolute* time-limit and that the candidate had only basic knowledge of MCMC before he started on this work. Not all topics in exact sampling are covered/cited and the thesis should be read with this in mind.

March 10, 1998
Håvard Rue

PREFACE

This report presents my diploma thesis performed the autumn 1997 at The Norwegian University of Science and Technology (NTNU). I have explored the new methods for perfect sampling via Markov chains, and have also implemented and tested some selected algorithms. The work has been divided into two parts, considering the discrete and continuous case respectively.

The first months were devoted to the discrete case, and theoretical literature studies were followed by implementation and testing of the algorithms. The last months I concentrated on continuous state spaces, and for this part with more focus on the theoretical side.

I have had a great time working with this thesis, and really feel I have gained a lot of new knowledge. Not only do I think of the particular subject of stochastic simulation, but also of programming techniques and more general statistics. From advice I started writing this report early, a recommendation I support with great conviction.

I would like to thank James Allan Fill at the The Johns Hopkins University, Baltimore,US, for helpful comments. A special thanks to Håvard Rue, my supervisor at The Department of Mathematical Sciences, for excellent cooperation. He has been helpful and always eager to introduce me to new and interesting aspects with the problems.

Trondheim, December 17, 1997.
Morten Fismen

1 INTRODUCTION

Exploring the probability distribution π using Markov Chain Monte Carlo (MCMC) methods is a widely used technique with growing popularity. It gives a way to deal with complex and multivariate distributions which can not be handled otherwise. The main idea is to run a Markov chain whose stationary distribution equals π. The states the chain possess after an initial "burn-in" period will then approximate the desired distribution. However, it is difficult to decide exactly how many steps we should perform in this "burn-in" period to achieve a desired accuracy with respect to the distribution. Further, the samples we receive will be dependent. Great interest has therefore been directed towards some new *perfect sampling* algorithms. These methods obtain samples which are both independent and distributed exactly according to the desired distribution π. Algorithms for both the discrete and continuous have been developed, and in this report we present the most important of these new methods.

The contents of this diploma thesis are as follows.

SECTION 2 gives a review of general theory of Markov chain Monte Carlo simulation. We first present the theory of Markov chains, and give the conditions for stationary distribution to exist. Then we show how to perform sampling from a distribution by simulating a Markov chain using the Metropolis-Hastings algorithm. A special case, the Gibbs sampler is described. Finally we describe the the main problem using Markov chain simulation, the question of convergence.

SECTION 3 gives a review of the theory of Markov random fields. The Ising model is presented, and we show how to apply a Gibbs sampler to simulate such fields.

SECTION 4 describes algorithms for perfect sampling from the desired distribution. The coupling algorithm of Propp and Wilson is described in 4.1, while the rejection sampler of Fill is described in 4.3. We also describe how we can get perfect samples of the Ising model by using the algorithms of Propp-Wilson and Fill.

SECTION 5 gives the results from empirical tests on the perfect sampling algorithms. When applied to an Ising model, the coupling time of the algorithm of Propp and Wilson is expected to increase as a function the grid size, and we present the results from a regression analysis. Also, the validity of Fills algorithm is verified through a χ^2 test.

SECTION 6 gives general theory of stochastic image analysis, and we describe how stochastic simulation on an Ising model can be used for noise reduction. We give the results from applying the perfect sampling methods in an image noise reduction algorithm. Image samples are compared with the original and synthetically noisy image.

IN SECTION 7 we describe how Propp and Wilson's idea of coupling can be extended to the continuous case. Several algorithms developed by Murdoch and Green are presented.

SECTION 8 contains a case study where we apply perfect sampling algorithms in the Bayesian analysis.

2 MARKOV CHAIN MONTE CARLO SIMULATION

Sampling from a probability distribution π with the aid of a computer is denoted *stochastic simulation*, and the purpose is to base statistical inference upon the samples. Suppose for example that $X \sim \pi(x)$ and we want to calculate $E[h(X)]$ for some function h. By sampling $X_1, \ldots, X_n$ from $\pi(x)$ we have the estimate

$$\hat{E}[h(X)] = \frac{1}{n}\sum_{i=1}^{n} h(x_i) \xrightarrow{a.s.} E[h(X)] \tag{1}$$

It is often difficult to sample directly from from the distribution π, for example if the normalisation constant to π is unknown. In such circumstances we can use MARKOV CHAIN MONTE CARLO SIMULATION (MCMC). This method is based on the theory of Markov chains, and we first review some basics facts (see for example Taylor and Karlin [15]). We consider the discrete case, but the same results can with slight modifications be extended to the continuous case.

2.1 MARKOV CHAINS

Consider the stochastic process $\{X_{(t)}\} = X_{(0)}, X_{(1)}, X_{(2)}, \ldots$ with $X_{(t)} \in \mathcal{S}$, $\mathcal{S}$ discrete and finite.

DEFINITION 2.1 *A stochastic process* $\{X_{(t)}\}$ *is a* MARKOV CHAIN *if*

$$\textit{Prob}(X_{(s)} = y | X_{(0)} = x_0, \ldots, X_{(s-1)} = x) = \textit{Prob}(X_{(s)} = y | X_{(s-1)} = x) \tag{2}$$

for all steps s and all states $x_0, .., x, y$.

In other words, the state of a Markov chain only depends on the state of the chain in the previous step.

Let P_t denote the transition matrix $P_t(x, y) = \text{Prob}(X_{(t)} = y | X_{(t-1)} = x)$, where $P_t(x, y)$ is the probability of going from state *x* to state *y* at step *t*. We say we have a STATIONARY Markov chain if $P_t(x, y) = P_s(x, y) = P(x, y)\ \forall s, t \in \mathbb{N}$. This means that the transition probabilities are invariant in time. Similarly, let P^m denote the *m-step* transition matrix $P^m(x, y) = \text{Prob}(X_{(t+m)} = y | X_{(t)} = x)$.

Two states, *x* and *y*, in a Markov chain COMMUNICATE if there exist finite *n,m* such that $P^m(x, y) > 0$ and $P^n(y, x) > 0$.

A Markov chain is IRREDUCIBLE if, for all *x,y*, there exists a number $m < \infty$ such that $P^m(x, y) > 0$. In an irreducible Markov chain there is a positive probability of going from every state to any other state in a finite number of steps.

A Markov chain is APERIODIC if the greatest common divisor for *m* where $P^m(x, x) > 0$, $\forall x$, equals 1. The complementary is called a PERIODIC Markov chain, and for such processes there exist states that cannot communicate for certain time steps. A Markov chain which is both aperiodic and irreducible is called REGULAR.

A regular Markov chain has an important property, which is stated below.

THEOREM 2.1 *Consider a Markov chain $\{X_{(t)}\}$ on the state space $\mathcal{S}$, with transition probabilities $P(x, y)$. If the Markov chain is regular then there exits a unique limiting probability distribution $\pi(x)$, $x \in \mathcal{S}$, such that*

$$\lim_{m\to\infty} \textit{Prob}(X_{(m)} = y | X_{(0)} = x) = \pi(y) > 0,\ y \in \mathcal{S} \tag{3}$$

or, in terms of the m-step probability matrix P^m

$$\lim_{m\to\infty} P^m(x, y) = \pi(y) > 0,\ y \in \mathcal{S} \tag{4}$$

This limiting distribution π is called the STATIONARY distribution of the Markov chain, and $\pi(x)$ may be interpreted as the long run probability of being in the state *x*. The stationary distribution may be found by solving the equations

$$\pi(x) = \sum_y \pi(y)P(y, x) \quad \text{and} \quad \sum_y \pi(y) = 1 \tag{5}$$

Some Markov chains exhibits the property of *monotonicity*, which depends on the existence of a partial ordering of the state space $\mathcal{S}$ (see [2]). Suppose $\mathcal{S}$ has dimension n, such that $x = (x_1, \ldots, x_n)$ for all $x \in \mathcal{S}$. We say that the state space $\mathcal{S}$ is a PARTIALLY ORDERED SET (POSET) if the configurations in $\mathcal{S}$ is ordered such that $x \preceq y$ when $x_i \preceq y_i$ for $i = 1, \ldots, n$. A subset $\mathcal{I}$ of a poset $\mathcal{S}$ is called an ORDER IDEAL if $x \in \mathcal{I}$ and $y \preceq x$ implies $y \in \mathcal{I}$. The set of all order ideals is denoted $J(\mathcal{S})$. Given two probability densities μ and ν on $\mathcal{S}$, we say that $\mu \preceq \nu$ STOCHASTICALLY if $\sum_{x\in\mathcal{I}} \mu(x) \geq \sum_{x\in\mathcal{I}} \nu(x)$ for all $\mathcal{I} \in J(\mathcal{S})$.

DEFINITION 2.2 *Let P be a transition matrix on a partially ordered state space $(\mathcal{S}, \preceq)$. P is* MONOTONE *if P preserves the partial order, i.e. if $P(x,\cdot) \preceq P(y,\cdot)$ stochastically whenever $x \preceq y$.*

2.2 MARKOV CHAIN SIMULATION

Now suppose we want to draw a sample from a distribution π. The main idea behind MCMC is to run a Markov chain, whose stationary distribution equals π, for a long time and then output the current state. If the Markov chain has reached equilibrium the output is distributed according to the stationary distribution, as desired.

Consider a regular Markov chain $\{X_{(t)}\}$ with transition matrix *P*. From theorem 2.1 we have the existence of a stationary distribution π, and we may write $\lim_{m\to\infty} ||P^m - \pi|| = 0$, where $||\mu|| = \sup_A |\mu(A)|$. During the simulation we run the Markov chain for M steps and our output is the state at step M. The number *M* has to be so large that $||P^t - \pi|| < \epsilon$, $\forall t \geq M$ for a desired accuracy ϵ. Until further, we leave the problem of deciding the size of *M*.

The task is now to construct a Markov chain with transition matrix *P* which has π as its stationary distribution. Hastings [5] show that reversible Markov chains, i.e. chains with

$$\pi(x)P(x, y) = \pi(y)P(y, x),\ \forall x, y \in \mathcal{S}, \tag{6}$$

fulfil the equation system (5). For reversible chains it is impossible to distinguish between forward and backward running of the chain. Now, to construct the transition probabilities $P(x, y)$ we set

$$P(x, y) = q(x, y)\alpha_{xy} \tag{7}$$

where the $q(x, y)$'s are the transition probabilities for another Markov chain fulfilling

$$q(x, y) > 0 \Rightarrow q(y, x) > 0 \quad \forall x, y \in \mathcal{S} \tag{8}$$

The algorithm is then

- From the current state x, propose a new state y with probability $q(x, y)$.
- Accept the new state with probability α_{xy}, else stay in the current state x.

Hastings [5] show that by choosing the acceptance probabilities to be

$$\alpha_{xy} = \min\left(1, \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)}\right) \tag{9}$$

the stationary distribution π is obtained. This algorithm is called the METROPOLIS-HASTINGS algorithm, and a simplification is the METROPOLIS-algorithm with $q(x, y) = q(y, x)$. Note that there is no need to know the normalisation factor for the distribution π, since the distribution appears as a fraction in the acceptance probability. This is one of the main strengths of Markov chain simulation. However, since there is no such things as free lunch, there are implicit costs which we will return to in section 2.4.

In the typical implementation of Monte Carlo simulation the samples are some transformation of the output from a seeded pseudo-random number generator. When dealing with simulation of Markov chains it is therefore often practical to consider the updating of the chain via a function $\phi : \mathcal{S} \times \mathcal{U} \to \mathcal{S}$ such that

$$X_{t+1} = \phi(x_t, U_{t+1}) \quad \text{and} \quad \text{Prob}(\phi(x, U) = y) = P(x, y) \tag{10}$$

where the random variable U takes on values in an arbitrary probability space $\mathcal{U}$. The function ϕ is called the UPDATING FUNCTION.

2.3 GIBBS SAMPLING

Gibbs sampling, also known as the HEATH BATH ALGORITHM for the discrete and binary case, is a method to draw samples from a multivariate distribution. It is a special case of the Metropolis-Hastings algorithm, and it makes use of the conditional distributions. Let $X = (X_1, ..., X_n) \sim \pi(x)$, and let $\pi(x_i|x_{-i})$ denote the distribution of X_i conditional on the other elements of X. To draw samples from $\pi(x)$, start with an initial state $x = (x_1, ..., x_n)$. Pick a site i to be updated, this can be done both randomly or predetermined sequentially. If the site i is to be picked randomly, we say have a RANDOM SITE UPDATING (RSU) rule. From the present state x, propose a new state $y = (x_1, .., x_{i-1}, y_i, x_{i+1}, .., x_n)$ such that y_i is drawn form the conditional $q(x, y) = \pi(x_i|x_{-i})$. The accept probability is

$$\alpha(x, y) = \min\left(1, \frac{\pi(y)\pi(x_i|x_{-i})}{\pi(x)\pi(y_i|x_{-i})}\right) = 1 \tag{11}$$

This means that using Gibbs sampling the proposal is always accepted. But note that the conditional distributions have to be known. We will later in section 3.2 see examples where the conditionals have a particular tractable form.

2.4 Convergence and independence in Markov chain simulation

As more and more iterations are performed we will receive samples whose distribution approaches the stationary distribution π. However, two problems immediately arises. First, how many steps should we iterate until the samples approximate π with desired accuracy? And second, since the values in a Markov chain are dependent, how should we proceed if inference should be based on a number of independent samples?

To decide the number of iterations various convergence tests have been proposed (see Brooks [1] for further references). Suppose we can determine a *convergence rate* or a bound on a such rate. As an example a Markov chain with finite and discrete state space converges geometrically such that

$$||P^t - \pi|| < M\rho^t,\ 0 < \rho < 1 \tag{12}$$

We can then stop the iteration once $||P^t - \pi|| < \epsilon$ for some desired accuracy $\epsilon > 0$, giving the length of the chain

$$t^* = \frac{\log(\epsilon/M)}{\log(\rho)} \tag{13}$$

These convergence rates are generally difficult to find, and upper bounds are often too weak to be of practical interest. A lot of work has therefore been performed in the development of *convergence diagnostics* methods ([1]). These methods uses eigenvalue estimation, plots of the sample paths, autocorrelation and quantiles and other techniques to *a posteriori* or *during run time* decide if the Markov chain has converged or not. But these methods all suffer from the possibility that the Markov chain reaches a *metastable* state. Recall from equation (9) that the normalisation constant is not known. Hence we can not measure the absolute probabilities, but only the relative probabilities between the states. So even if the chain seems to be stable, there may exist domains in the state space which not yet is visited, and where the states have much higher probability than in the visited domains. The only way to ensure we have convergence is to check that we have visited every state, but if we could do this in a practical way there generally would be no need to do the simulation at all.

After convergence the samples will approximate the desired distribution, and by saving a number of samples we can estimate $E[h(X)]$. However, since the states in a single run of a Markov chain are dependent, we can not control the variance of this estimate. To receive approximative independent samples there are two possible solutions. We can simulate a long Markov chain and save only each k'th sample. By this we will have larger belief that the Markov chain possesses the correct distribution, however it can be difficult do decide the size of k. Or we can run several Markov chains and sample the value at the end of the iteration. The initial values of the chains should then cover as much of the state space as possible.

Even if these methods in practise has shown to be useful, it would be desirable to have an algorithm returning samples both exactly distributed according to π and independent. Such algorithms will be presented in section 4.

3 The Ising Model

We will now introduce some areas where MCMC has shown useful. The Ising model arose in statistical physics, but has been widely used in image processing and often serve as a first introductory non-trivial example in the Markov chain society. The statistical foundation of the Ising model is the theory of Markov random fields, and for clarity we now review some background material on such fields.

3.1 Random fields

Let $\mathcal{L}_\mathcal{D}$ be a grid over some domain $\mathcal{D}$, and let $X = \{X_i; i \in \mathcal{L}_\mathcal{D}\}$ be a a random vector with state space $\mathcal{S}$ defined on the grid. If we have

$$\text{Prob}(X = x) = \pi(x) > 0 \; \forall \; x \in \mathcal{S} \tag{14}$$

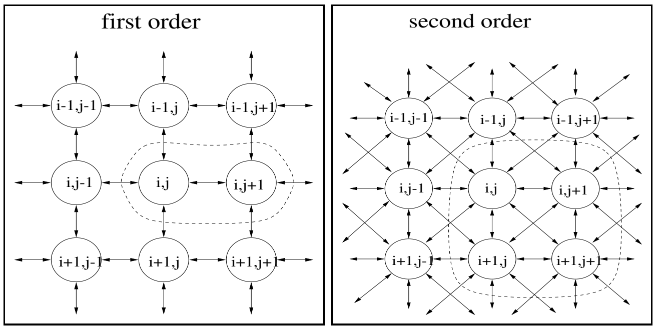
then we say we have a STOCHASTIC or RANDOM FIELD.

A collection $\delta = \{\delta(s); s \in \mathcal{L}_\mathcal{D}\}$ of subsets of $\mathcal{L}_\mathcal{D}$ is denoted a NEIGHBOURHOOD SYSTEM if :

- $s \notin \delta(s)$
- $s \in \delta(t)$ if and only if $t \in \delta(s)$

The sites $s \in \delta(t)$ is denoted the NEIGHBOURS of t, and we write $\langle s, t \rangle$ if s and t are neighbours. A subset $\mathcal{C} \subseteq \mathcal{L}_\mathcal{D}$ is called a CLIQUE if $s \in \delta(t) \; \forall \; s, t \in \mathcal{C}$.

EXAMPLE 3.1 *Consider a 2D rectangular grid $\mathcal{L}_\mathcal{D}$. We have a* FIRST-ORDER *neighbourhood system if the neighbours of an element are the elements above, under, to the left and to the right of the element. This is also denoted a* FOUR-NEIGHBOUR *system. If, in addition, the diagonal elements are neighbours then we have a* SECOND-ORDER *system, or an* EIGHT-NEIGHBOUR *system. Figure 1 pictures the two models.*



a b

Figure 1: Illustration over the *first-order* (a) and *second-order* (b) neighbourhood system. The arrows indicate neighbours, while the dashed line surrounds the elements of the largest possible cliques for the two models

Now let $\pi(x_s|x_r, r \neq s) = \pi(x_s|x_{-s})$ denote the conditional distribution of $\{X_s, s \in \mathcal{A}\}$ conditioned on $\{X_r, r \in \mathcal{B}\}$, where $\mathcal{A} \cap \mathcal{B} = \emptyset$ and $\mathcal{A} \cup \mathcal{B} = \mathcal{S}$. A random field is a MARKOV FIELD if for all $r, s \in \mathcal{L}_{\mathcal{D}}$

$$\text{Prob}\left(X_s = x_s|X_r = x_r, r \neq s\right) = \text{Prob}\left(X_s = x_s|X_r = x_r, r \in \delta(s)\right) \tag{15}$$

i.e. the conditional distribution of a site only depends on its neighbours.

A GIBBS FIELD with regard to a neighbourhood system $\delta = \{\delta(s), s \in \mathcal{L}_{\mathcal{D}}\}$ is a random field of the form

$$\pi(x) = c \times \exp\left(-H(x)\right), \quad \pi(x) > 0 \quad \forall x \tag{16}$$

where $c = \frac{1}{\sum_{x'} \exp(-H(x'))}$ is the normalisation constant, $H(x) = \sum_{\mathcal{C}'} U_{\mathcal{C}'}(x)$ (that is, sum over all possible cliques $\mathcal{C}'$), and $U_{\mathcal{C}}(x)$ is a function only of the values of the sites lying within the clique $\mathcal{C}$.

The origin of this model is in statistical physics, and $H(x)$ is denoted the ENERGY FUNCTION, $c^{-1} = Z$ is denoted the PARTITION FUNCTION, and $U_{\mathcal{C}}(x)$ is denoted the POTENTIAL FUNCTION.

There is a strong connection between Gibbs fields and Markov fields, and the following theorem show that they are equivalent.

THEOREM 3.1 (HAMMERSLEY-CLIFFORD) *Let δ be a neighbourhood system. A random field is a Markov field with respect to δ if and only if it is a Gibbs field with respect to δ.*

HALF PROOF: We give the first trivial part of the proof, that is to prove that a Gibbs field is Markov field with respect to δ. Consider the conditional distribution of X_s conditioned on X_r

$$\begin{aligned}
\pi(x_s|x_{-s}) &= \text{Prob}(X_s = x_s|X_r = x_r; r \neq s) \\
&= \frac{\pi(x_s, x_{-s})}{\pi(x_{-s})} \\
&= \frac{\pi(x_s, x_{-s})}{\sum_{x'_s} \pi(x_{s'}, x_{-s})} \\
&= \frac{c \times \exp\left(\sum_{\mathcal{C}} -U_{\mathcal{C}}(x_s, x_{-s})\right)}{\sum_{x_{s'}} c \times \exp\left(\sum_{\mathcal{C}} -U_{\mathcal{C}}(x'_s, x_{-s})\right)} \\
&= \frac{\exp\left(-\sum_{\mathcal{C}:s\in\mathcal{C}} U_{\mathcal{C}}(x_s, x_{-s}) - \sum_{\mathcal{C}:s\notin\mathcal{C}} U_{\mathcal{C}}(x_s, x_{-s})\right)}{\sum_{x'_s} \exp\left(-\sum_{\mathcal{C}:s\in\mathcal{C}} U_{\mathcal{C}}(x'_s, x_{-s}) - \sum_{\mathcal{C}:s\notin\mathcal{C}} U_{\mathcal{C}}(x'_s, x_{-s})\right)}
\end{aligned} \tag{17}$$

Now, since the potential function $U_{\mathcal{C}}(x)$ is a function only of sites lying within the clique $\mathcal{C}$, we find that $U_{\mathcal{C}}(x'_s, x_{-s}) = U_{\mathcal{C}}(x_s, x_{-s})$ for all cliques $\mathcal{C}$; $s \notin \mathcal{C}$. This gives

$$\begin{aligned}
\pi(x_s|x_{-s}) &= \frac{\exp\left(-\sum_{\mathcal{C}:s\in\mathcal{C}} U_{\mathcal{C}}(x_s, x_{-s})\right)}{\sum_{x'_s} \exp\left(-\sum_{\mathcal{C}:s\in\mathcal{C}} U_{\mathcal{C}}(x'_s, x_{-s})\right)} \\
&= c_s \times \exp\left(-\sum_{\mathcal{C}:s\in\mathcal{C}} U_{\mathcal{C}}(x_s, x_{-s})\right)
\end{aligned} \tag{18}$$

which is independent of all $X_r; r \notin \delta(s)$, and hence a Gibbs field is a Markov field with respect to δ. $\square$

3.2 The Ising Model

The Ising model is a particular simple and tractable form of a Gibbs field. The model was introduced in the study of ferromagnetic substances, and has been widely used in areas like statistical mechanics and image processing. An Ising system consists of a two-dimensional (2D) or three-dimensional (3D) collection of elements, each element possessing the property of a *spin*. The spin may be aligned in two directions, either *up* or *down*. The direction of the spin is a random quantity, depending on the directions of the spin of the other elements and, if it exists, on an external field. Preferably, the spin is aligned in the same direction as the spin of neighbouring elements, and all elements prefer to have spin aligned with the external field. Consider a 2D Ising system with elements on the grid $\mathcal{L}_\mathcal{D}$ in the domain $\mathcal{D}$. The system is specified with the random vector $X = \{X_i; i \in \mathcal{L}_\mathcal{D}\}$, where each X_i can take two possible values ± 1 according to as the spin of the element *i* is up or down. Note that we can impose a partial ordering on an Ising field, as $x \preceq y$ if $x_i = -1$ whenever $y_i = -1$. The Ising model has a first-order neighbourhood, and the energy function is given by

$$H(x) = -\sum_{<i,j>} \beta_{i,j} x_i x_j - \sum_i \alpha_i x_i \tag{19}$$

where the parameter $\beta_{i,j}$ represents the degree of interaction between site *i* and *j*, and α_i is the strength of the external field measured at site *i* . In case of $\alpha_i = 0$ and $\beta_{i,j} = \beta$ for all *i,j* we have

$$\begin{aligned} H(x) &= -\sum_{<i,j>} \beta x_i x_j \\ &= -\beta \times (\#\text{equal neighbours} - \#\text{unequal neighbours}) \end{aligned} \tag{20}$$

From this we find

$$\pi(x) = c \times \exp\left(\beta \times (\#\text{equal neighbours} - \#\text{unequal neighbours})\right) \tag{21}$$

which states that more equal neighbours give higher probability. Sampling from the Ising model is difficult due the large number of possible configurations. Typically, $\mathcal{L}_\mathcal{D}$ is a rectangular grid with for example $n = 64 \times 64$ sites. The partition function is then a sum over $2^{64\times 64}$ terms and hence infeasible to calculate. However, sampling can be done effectively via Gibbs sampling, with no need of calculating the partition function. Recall that Gibbs sampling method updates one site at a time, and that the conditional distribution is used as proposal. From the proof of Theorem 3.1 we find the conditional distribution of site i to be

$$\begin{aligned} \pi(x_i|x_{-i}) &= c_i \times \exp\left(-\sum_{\mathcal{C}:i\in\mathcal{C}} U_\mathcal{C}(x_i, x_{-i})\right) \\ &= c_i \times \exp\left(\sum_{j:j\in\delta(i)} \beta_{i,j} x_i x_j + \alpha_i x_i\right) \end{aligned} \tag{22}$$

Now, for the Ising model each site can possess only two possible values (spin up or down). If the site

i is to be updated, the spin at this site is therefore set to $+1$ with probability

$$
\begin{aligned}
\text{Prob}(X_i = +1 | x_{-i}) &= \pi(x_i = 1 | x_{-i}) \\
&= \frac{\exp\left(\sum_{j:j\in\delta(i)} \beta_{i,j} x_j + \alpha_i\right)}{\exp\left(\sum_{j:j\in\delta(i)} \beta_{i,j} x_j + \alpha_i\right) + \exp\left(-\sum_{j:j\in\delta(i)} \beta_{i,j} x_j - \alpha_i\right)} \\
&= \left[1 + \exp\left(-2 \sum_{j:j\in\delta(i)} \beta_{i,j} x_j - 2\alpha_i\right)\right]^{-1} \qquad (23)
\end{aligned}
$$

and to -1 with the complementary probability. It is clear that $\pi(x_i|x_{-i})$ is an increasing function of x_{-i} in the partial order, and distributions possessing this behaviour are called ATTRACTIVE. Also, $\pi(x_i|x_{-i})$ is quickly calculated being a sum of only five elements. Now, by using the random site updating rule, we find that the updating from X to X' at each iteration may be written in terms of a updating function ϕ such that $X' = \phi(x, U_1, U_2)$. U_1 is drawn uniformly on $(1, \ldots, n)$ and decides the site i to be updated. U_2 is a uniform $(0, 1)$ variable, and we have

$$
X' = \phi(x, U_1, U_2) = \begin{cases} (x; x_i \leftarrow 1) & : \quad U_2 \leq \pi(x_i = 1 | x_{-i}) \\ (x; x_i \leftarrow -1) & : \quad U_2 > \pi(x_i = 1 | x_{-i}) \end{cases} \qquad (24)
$$

where $(x; x_i \leftarrow \sigma)$ denotes the configuration which has spin σ at site i and else has the same spin as x. The transition matrix P is then $P(x, (x; x_i \leftarrow \sigma)) = 1/n \times \pi(x_i = \sigma | x_{-i})$.

4 Exact simulation for discrete state spaces

As described in 2.4 it is a problem to decide how big M should be so that $||P^M - \pi|| < \epsilon$ for a fixed ϵ. This is a difficult task, and the choice of the initial state can greatly affect the time to equilibrium.

Recently Propp and Wilson [11] came up with an algorithm that eliminates the question about the size of M, by introducing a method that receives samples which are distributed exactly according to the target distribution π. This is denoted EXACT or PERFECT SAMPLING, and their pioneer work has attracted considerable interest. Their idea is based on coupling, or coalescence, of Markov chains. Later Fill [2] has devised another perfect sampling algorithm based on rejection sampling. Additionally, both these methods return independent samples.

4.1 The Propp-Wilson algorithm

We now will introduce the algorithm of Propp and Wilson, which is based on an intuitive idea. First, consider to simultaneously run $N = |\mathcal{S}|$ copies of the Markov chain from every possible initial state. Let the same updating function and sequence of random numbers be used for all the chains. We say that two chains have COALESCED if the two chains at one step reach the same state. From that step and onward the two chains will follow the same sample path, due to the fact that the same updating function is used for all chains. Now, suppose all *N* chains have coalesced such that the effect of the initial state has worn off. A simple MCMC algorithm would then be to start *N* chains in every possible initial state and output the current state at once all chains had coalesced. However, this will not yield a correct algorithm. To see this, consider Markov chains with states with a unique predecessor. If the above algorithm were run, we would obtain samples that were biased.

EXAMPLE 4.1 This example is collected from [2]. Consider a Markov chain with state space *S* = $\{0,1\}$, with the following transition matrix

	0	1
0	0.5	0.5
1	1.0	0.0

From state 0 the chain moves to 0 or 1 with probability 1/2 each, and from state 1 the chain moves deterministically to state 0. By solving the equation system (5) we find the stationary distribution to be $\pi(0) = 2/3, \pi(1) = 1/3$. The naive algorithm would then be to run two Markov chains, one from state 0 and the other from state 1, and output the state of the chains once they had coalesced. However, since the chain moves deterministically to state 0 from state 1, the two chains will with probability one coalesce in state 0.

4.1.1 The algorithm in a general setting

The solution to the biased sampling problem is quite simple and still remarkable. Instead of running the chains from the present into the future, the simulation is done from the past to the present. To see why this gives a correct sampling scheme, we introduce some useful notation and give the algorithm in a more general setting.

Consider first a situation where the number of iterations M is fixed. Our goal is to run a Markov chain, starting in an arbitrary state x^*, from time $-M$ to time 0. The updating at each step is done via the updating function $\phi : \mathcal{S} \times \mathcal{U} \to \mathcal{S}$ such that Prob $(\phi(x, U) = y) = P(x, y)$. The output from a chain with initial state x^* is now determined by the mapping $F^0_{-M}(x^*, U_0, \ldots, U_{-M+1})$, where F^0_t is defined recursively by

$$F^0_0(x) = x \tag{25}$$

$$F^0_{-1}(x, U_0) = \phi(x, U_0) \tag{26}$$

$$F^0_t(x, U_0, U_{-1}, \ldots, U_{t+2}, U_{t+1}) = F^0_{t+1}(\phi(x, U_{t+1}), U_0, U_{-1}, \ldots, U_{t+2}) \tag{27}$$

Note that if $F^0_{t'}$ is constant map for some $t' \in [-M, 0\rangle$, with $F^0_{t'}(x, U_0, U_{-1}, \ldots, U_{t'+1}) = F^0_{t'}(x', U_0, U_{-1}, \ldots, U_{t'+1})$ for all x, x', then this will be the situation for all $t \le t'$. Particularly we would have the identity $F^0_{-M} = F^0_{t'}$. Now, by setting M equal to infinity it is clear that the output values of $F^0_{-\infty}$ is distributed exactly according to the stationary distribution π of the Markov chain. Since this output equals the value of F^0_t for all t's where F^0_t is a constant map, it will be sufficient to dynamically determine a value of t where F^0_t is a constant map. Suppose we were running N Markov chains, starting in each of the N possible initial states, from time t to time 0. It is clear that a constant map F^0_t is equivalent with coalescence of all chains with different initial states. However, the value of the mapping F^0_t is not necessarily the same as the state of the chains at the time of coalescence. That is, the chains may coalescence *before* time 0. This explains why backward simulation obtains unbiased samples. On the opposite, if we were doing forward simulation from time 0, then the time t at which the map F^t_0 became constant would coincide with the time of coalescence. Also, the value of F^t_0 is not necessarily the same as the value F^∞_0, the latter being a correct sample from the stationary distribution π.

An algorithm could then be stated as follows

> $t \leftarrow 0$
> repeat
> $t \leftarrow t - 1$
>   for all $x \in \mathcal{S}$
> $F^0_t(x, U_0, U_{-1}, \ldots, U_{t+2}, U_{t+1}) \leftarrow F^0_{t+1}(\phi(x, U_{t+1}), U_0, U_{-1}, \ldots, U_{t+2})$
> until F^0_t is a constant map
> return the unique value $F^0_t(\cdot, U_0, U_{-1}, \ldots, U_{t+2}, U_{t+1})$

In practise this algorithm is an iterative procedure. N copies of the chain, starting in each of the possible initial states, are run from time $t = -1$ to time 0. If all the chains have coalesced, then output the current value of the chains, else set $t = t - 1$ and start all over. It is important that the variables U_t are the same in each iteration and also for each of the N chains. If not, biased samples would again be obtained (see [11]).

The algorithm presented will not be feasible if the state space $\mathcal{S}$ is large . Efficient simulation is possible only if we can reduce the number of chains to run. Recall the definition 2.2 of a monotone transition matrix. Suppose we can find a updating function $\phi : \mathcal{S} \times \mathcal{U} \to \mathcal{S}$ satisfying (10) such that $\phi(x, u) \preceq \phi(y, u)$ for all $u \in \mathcal{U}$ whenever $x \preceq y$. If, in addition, we can find minimal and maximal elements $\hat{0}, \hat{1}$ such that $\hat{0} \preceq x \preceq \hat{1}$ for all $x \in \mathcal{S}$, we have the MONOTONE CASE. Obviously, if the monotone case is present, then it is sufficient to start the Markov chain in the two extreme states $\hat{0}, \hat{1}$. Indeed, if these two chains coalesce, then so will all the others.

4.2 Propp-Wilson's algorithm applied to the Ising model

The implementation of Propp-Wilson's coupling from the past technique to the Gibbs sampler is straightforward as described in section 4.1. Two Markov chains X and Y are at time $-M$ started in the two extreme states $\hat{0}, \hat{1}$, with all spin up or all spin down. The updating of the Markov chains is done via the updating function ϕ defined in equation (24), such that $X_{t+1} = \phi(x_t, U_{1,t}, U_{2,t})$ and $Y_{t+1} = \phi(y_t, U_{1,t}, U_{2,t})$. If $X_0 = Y_0$ we set this value as our sample. Recall the importance that the value of $(U_{1,t}, U_{2,t})$ is the same in each iteration. This could be achieved by storing $(U_{1,t}, U_{2,t})$, however this demands a large amount of memory. A better solution is to use a seeded pseudo-number generator, and set the seed as a function of t. If the algorithm is run more than once to obtain several independent samples, then the seed should be a function of both t and the sample number k. Otherwise we would receive biased samples. Some samples generated with this model are shown in Figure 2 and Figure 3. Figure 2 shows the effect of varying β in systems with no external field, while Figure 3 shows the effect of applying an external field.

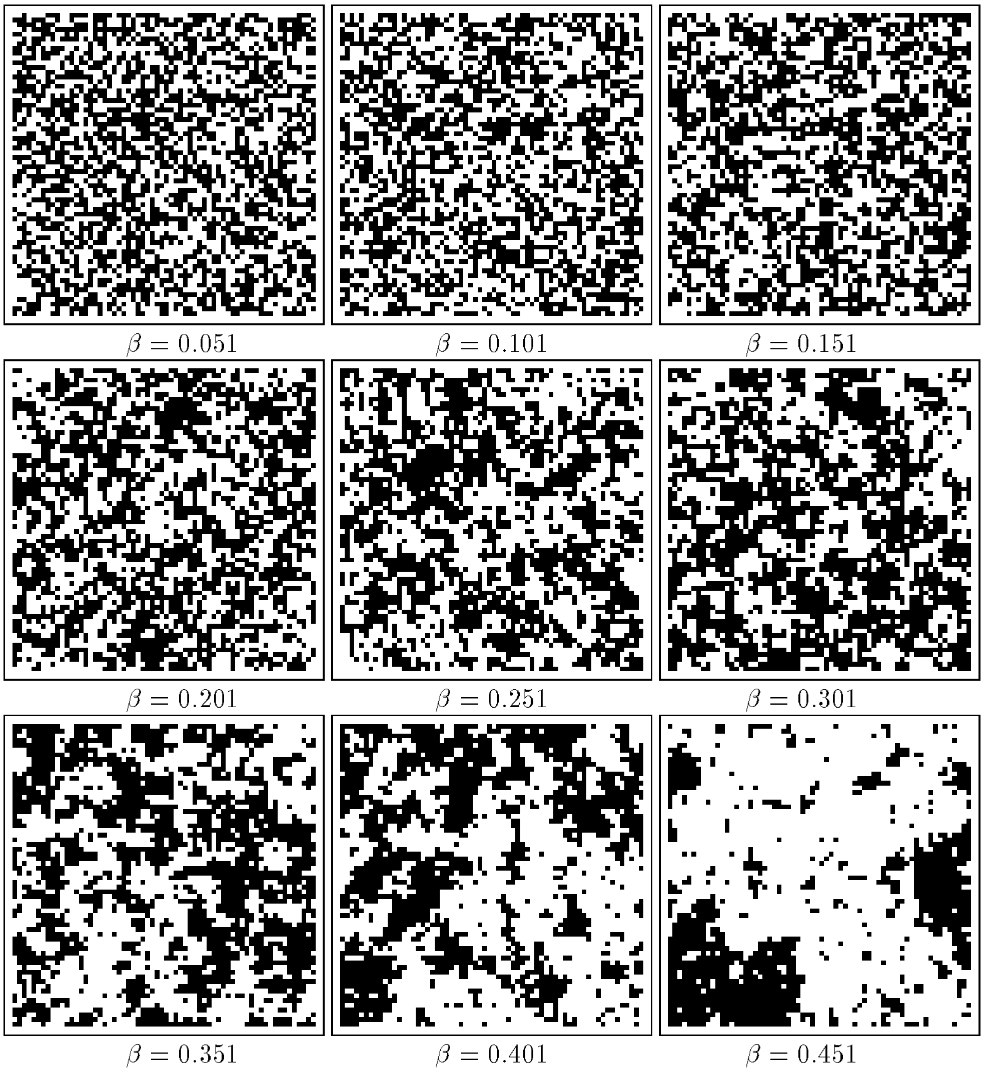


Figure 2: Samples from an Ising field, grid size 64×64, as function of the attractiveness parameter β. The samples are produced with Propp-Wilson's algorithm.

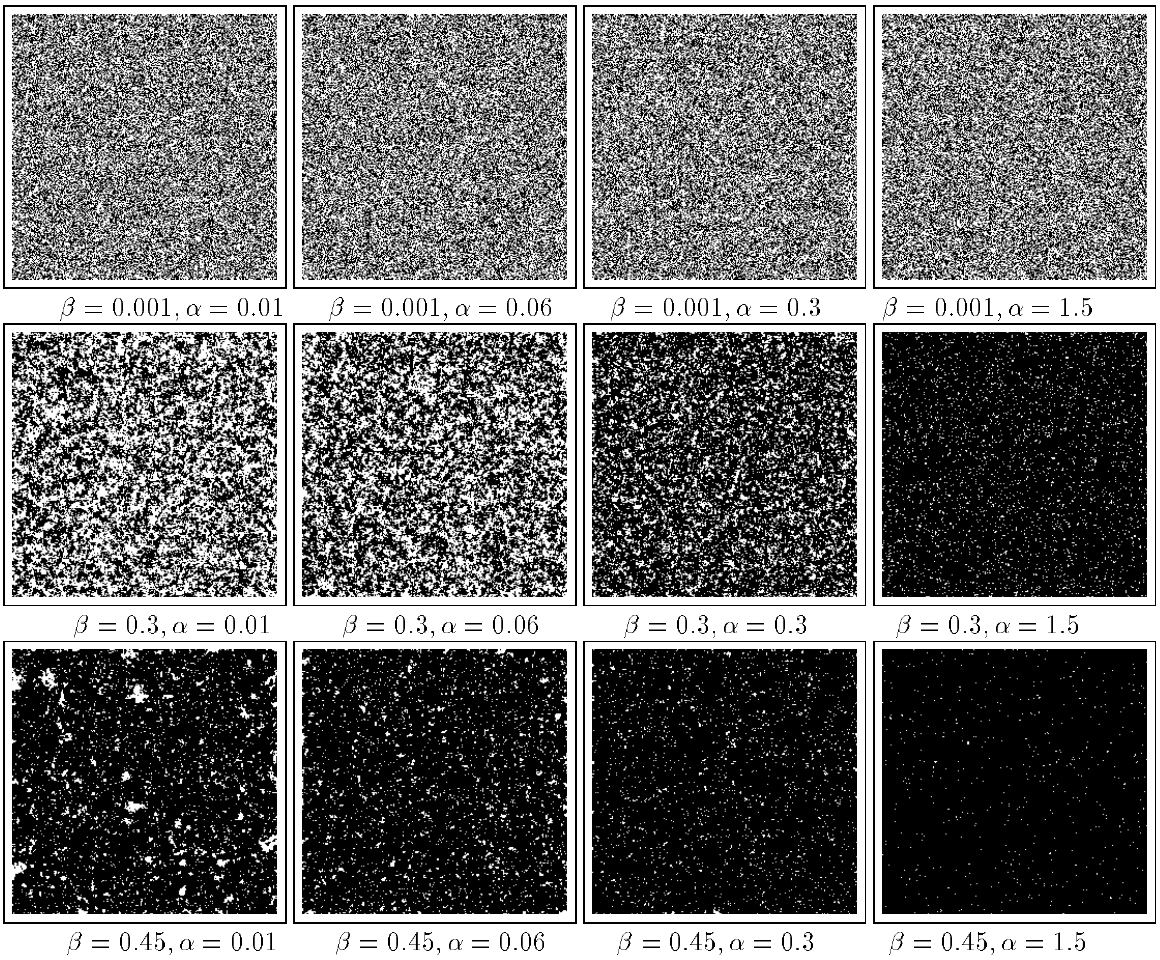


Figure 3: Samples from an Ising field, grid size 256×256, as function of the attractiveness parameter β and the external field α. The samples are produced with the Propp-Wilson algorithm.

4.3 FILLS INTERRUPTIBLE ALGORITHM

Propp-Wilson's algorithm returns independent and exact samples from the stationary distribution. However, it is crucial that the simulation is allowed to iterate until output is received. If a simulation is interrupted if no output is received during a fixed number of iterations, then the samples will be biased in favour of configurations that take short time to generate.

EXAMPLE 4.2 Fill [2] sketch an example of why interrupting the simulation leads to biased samples, and we give here the details. Consider a Markov chain on the state space $S = \{0, 1, 2\}$, with the following transition matrix

	0	1	2
0	0.5	0.5	0.0
1	0.5	0.0	0.5
2	0.0	0.5	0.5

By solving the equation system (5) we find the stationary distribution to be $\pi(0) = 1/3, \pi(1) = 1/3, \pi(2) = 1/3$. Monotonicity is obtained if we have the partial ordering $\hat{0} = 0 \preceq 1 \preceq 2 = \hat{1}$, and we may then run the Propp-Wilson algorithm with the two chains *X* and *Y* with initial states $x_0 = 0, y_0 = 2$. To secure the monotone case obtains the chains evolves according a transition rule using a uniform random variable U in the following way: $X_t = \max(X_{t-1} - 1, 0)$ if $U \leq 0.5$ and $X_t = \min(X_{t-1} + 1, 2)$ if $U > 0.5$

Suppose we decide to abort runs not completed in M_0 iterations, and set $M_0 = 2$. Let I be a categorical random variable taking value 1 if the two chains have coalesced within M_0 iterations, and taking the value -1 if no coalescence have occurred. Let Z denote the output of the simulation conditionally given completion. By investigating the transition matrix we find that the two chains cannot coalesce in just one step, and with two steps we have the four following possible outcomes:

- $U_1 \leq 0.5, U_2 \leq 0.5 : I = 1, Z = 0$
- $U_1 \leq 0.5, U_2 > 0.5 : I = -1, Z$ not defined
- $U_1 > 0.5, U_2 \leq 0.5 : I = -1, Z$ not defined
- $U_1 > 0.5, U_2 > 0.5 : I = 1, Z = 2$

Thus we see that this algorithm will yield output Z governed by the distribution $\pi'(0) = 1/2, \pi'(1) = 0, \pi'(2) = 1/2$, whereas the desired distribution was $\pi(0) = 1/3, \pi(1) = 1/3, \pi(2) = 1/3$.

To overcome this problem Fill [2] has devised an interruptible exact algorithm which gives both independent and unbiased samples, even if long runs are aborted. Fill consider running a monotone Markov chain from the minimal (or maximal) state $\hat{0}$ for t steps, with probability distribution of the output z governed by the t-step transition matrix $P^t(\hat{0}, z)$. The idea is to find an upper bound *L* on the ratio $\pi(z)/P^t(\hat{0}, z)$ and accept the output z as an observation from π with probability $L^{-1}\pi(z)/P^t(\hat{0}, z)$. This can be seen as a form of rejection sampling [12].

4.3.1 BACKGROUND

Fill's method requires knowledge to some special results and definitions ([2]). Suppose we have a monotone transition matrix P such that $P(x, \cdot) \preceq P(y, \cdot)$ stochastically for all $x \preceq y$. Then there exist an UPWARD KERNEL $K_{x,y}(\cdot,\cdot)$ such that (1) for all $x' \in \mathcal{S}$, the measure $K_{x,y}(x',\cdot)$ is supported on $\{y' \in \mathcal{S} : x' \preceq y'\}$ and (2) we have for all $y' \in \mathcal{S}$ the identity

$$P(y,y') = \sum_{x'} P(x,x')K_{x,y}(x',y') \tag{28}$$

Further, let $\tilde{P}(x,y) = \pi(y)P(y,x)/\pi(x)$ denote the TIME-REVERSAL of a Markov transition matrix P with stationary distribution π.

4.3.2 THE ALGORITHM

Let P be a Markov transition matrix with monotone time-reversal $\tilde{P}$, and let the partially ordered state space S possess both a unique minimum element $\hat{0}$ and a unique maximum element $\hat{1}$. Let $K_{x,y}$ be the upward kernel for $\tilde{P}$.

First, run the Markov chain X with transition matrix P for t steps, record the trajectory $(X_0,\ldots,X_t)$. The chain is started from state $\hat{0}$, and let $x_t = z$. Next, consider $(X_t,\ldots,X_0)$ as the time-reversed Markov chain $\tilde{X} = (\tilde{X}_0,\ldots,\tilde{X}_t)$ with transition matrix $\tilde{P}$. Then run a second Markov chain $\tilde{Y}$, starting from state $\hat{1}$. From state y this chain choose its new state y' with transition probability $K_{x,y}(x',y')$, where x to x' is the observed transition of $\tilde{X}$. If $\tilde{Y}_t = \hat{0}$ the simulation is finished and the sample is z. If $\tilde{Y}_t \neq \hat{0}$ we have two possibilities. We can either proceed the current run with an increased value of t, or we can abort the current run and start a completely new iteration.

The correctness of this algorithm is verified by considering the two-phase routine as an implementation of rejection sampling. The idea behind rejection sampling is to sample an observation x from a distribution f, and accept x as a sample from another distribution g with probability $\frac{g(x)}{Lf(x)}$, where $L \geq \sup_x \frac{g(x)}{f(x)}$. Our goal is to simulate an observation from π, and our proposal is an observation from $P^t(\hat{0},z)$. We accept z if $\tilde{Y}_t = \hat{0}$, and must therefore show that this gives a valid acceptance probability. The proposal should be accepted with probability $L^{-1}\pi(z)/P^t(\hat{0},z)$, where L is an upper bound on the ratio $\pi(z)/P^t(\hat{0},z)$. The existence of the monotone time reversal $\tilde{P}$ implies

$$\frac{\pi(z)}{P^t(\hat{0},z)} = \frac{\pi(\hat{0})}{\tilde{P}^t(z,\hat{0})} \leq \frac{\pi(\hat{0})}{\tilde{P}^t(1,\hat{0})} = L \tag{29}$$

and hence we should accept z with probability

$$\begin{aligned}\frac{1}{L} \times \frac{\pi(z)}{P^t(\hat{0},z)} &= \frac{\tilde{P}^t(1,\hat{0})}{\pi(\hat{0})} \times \frac{\pi(z)}{P^t(\hat{0},z)} \\ &= \frac{\tilde{P}^t(1,\hat{0})}{\pi(\hat{0})} \times \frac{\pi(\hat{0})}{\tilde{P}^t(z,\hat{0})} \\ &= \frac{\tilde{P}^t(1,\hat{0})}{\tilde{P}^t(z,\hat{0})}\end{aligned} \tag{30}$$

The following lemma shows that the probability of observing $\tilde{Y}_t = \hat{0}$ in fact is $\tilde{P}^t(1,\hat{0})/\tilde{P}^t(z,\hat{0})$.

LEMMA 4.1

$$\textit{Prob}\left(\tilde{Y}_t = \hat{0} | \tilde{X}_0 = z, \tilde{X}_t = \hat{0}, \tilde{Y}_0 = \hat{1}\right) = \frac{\tilde{P}^t\left(\hat{1}, \hat{0}\right)}{\tilde{P}^t\left(z, \hat{0}\right)} \tag{31}$$

PROOF: We fill in some of the details sketched in Fill [2].

Consider first the stochastic process $\left\{(\tilde{X}_t, \tilde{Y}_t)\right\} = (\tilde{X}_0, \tilde{Y}_0), (\tilde{X}_1, \tilde{Y}_1), \ldots$. From the nature of the algorithm it is clear this is a Markov chain with transition probability

$$\text{Prob}(\tilde{X}_{t+1} = x', \tilde{Y}_{t+1} = y' | \tilde{X}_t = x, \tilde{Y}_t = y) = \tilde{P}(x, x') K_{x,y}(x', y') \tag{32}$$

Marginally we have Prob$(\tilde{X}_{t+1} = x' | \tilde{X}_t = x, \tilde{Y}_t = y) = \tilde{P}(x, x')$, and from (28) we get

$$\begin{aligned} \text{Prob}(\tilde{Y}_{t+1} = y' | \tilde{X}_t = x, \tilde{Y}_t = y) &= \sum_{x'} \tilde{P}(x, x') K_{x,y}(x', y') \\ &= \tilde{P}(y, y') \end{aligned} \tag{33}$$

and hence Y_t is marginally a Markov chain with transition matrix $\tilde{P}$. Also, Y_t is independent of X_{t-1}. Now

$$\text{Prob}\left(\tilde{Y}_t = \hat{0} | \tilde{X}_0 = z, \tilde{X}_t = \hat{0}, \tilde{Y}_0 = \hat{1}\right) = \frac{\text{Prob}\left(\tilde{Y}_t = \hat{0}, \tilde{X}_t = \hat{0} | \tilde{X}_0 = z, \tilde{Y}_0 = \hat{1}\right)}{\text{Prob}\left(\tilde{X}_t = \hat{0} | \tilde{X}_0 = z, \tilde{Y}_0 = \hat{1}\right)} \tag{34}$$

Since $\tilde{Y}_t = \hat{0}$ entails $\tilde{X}_t = \hat{0}$, this equals

$$\frac{\text{Prob}\left(\tilde{Y}_t = \hat{0} | \tilde{X}_0 = z, \tilde{Y}_0 = \hat{1}\right)}{\text{Prob}\left(\tilde{X}_t = \hat{0} | \tilde{X}_0 = z, \tilde{Y}_0 = \hat{1}\right)} = \frac{P^t\left(\hat{1}, \hat{0}\right)}{P^t\left(z, \hat{0}\right)} \tag{35}$$

as desired. □

4.4 FILL'S ALGORITHM APPLIED TO THE ISING MODEL

The implementation of Fill's method requires some more investigation, and Fill [2] describes how this may be done. Recall from section 4.3 that we accept an output $x_t = z$ from $P^t(\hat{0}, \cdot)$ as a sample from π with probability $\tilde{P}^t(\hat{1}, \hat{0})/\tilde{P}^t(z, \hat{0})$, or if $\tilde{Y}_t = \hat{0}$. Also, $\tilde{X}$ is updated through the updating function $\phi(x, U_1, U_2)$ such that $\tilde{P}(x, x') = \text{Prob}(\phi(x, U_1, U_2) = x')$. $\tilde{Y}$ gets a new y' value with probability $K_{x,y}(x', y')$, and Fill [2] suggest we may use $K_{x,y}(x', y') = \text{Prob}(\phi(y, U_1, U_2) = y' | \phi(x, U_1, U_2) =$

x'). This yields a correct choice since

$$\begin{aligned}
\sum_{x'} \tilde{P}(x,x')K_{x,y}(x',y') &= \sum_{x'} \text{Prob}(\phi(x,U_1,U_2)=x') \\
&\quad \times \text{Prob}(\phi(y,U_1,U_2)=y'|\phi(x,U_1,U_2)=x') \\
&= \sum_{x'} \text{Prob}(\phi(x,U_1,U_2)=x') \\
&\quad \times \frac{\text{Prob}(\phi(y,U_1,U_2)=y',\phi(x,U_1,U_2)=x')}{\text{Prob}(\phi(x,U_1,U_2)=x')} \\
&= \sum_{x'} \text{Prob}(\phi(y,U_1,U_2)=y',\phi(x,U_1,U_2)=x') \\
&= \text{Prob}(\phi(y,U_1,U_2)=y') \\
&= \tilde{P}(y,y')
\end{aligned} \tag{36}$$

as required in equation (28). We can generate a sample from $K_{x,y}(x',\cdot)$ by setting $Y'=\phi(y,\hat{U}_1,\hat{U}_2)$ such that $\left(\hat{U}_1,\hat{U}_2\right) \sim \lambda\left(U_1,U_2|\phi(x,U_1,U_2)=x'\right)$. That is, λ is the conditional distribution of (U_1,U_2) given $\phi(x,U_1,U_2)=x'$, and where $x'=(x;x_\nu \leftarrow \sigma)$. It is clear that we have $\hat{U}_1=U_1=\nu$. Further, the nature of the Gibbs sampler and monotonicity that suggest we can treat three possible situations of the development of $\tilde{X}$ differently.

1. $x' \neq x$ with $x'=(x;x_\nu \leftarrow 1)$. Since $\tilde{X}_t \preceq \tilde{Y}_t$ for all t is it not necessary to calculate the distribution λ. We set deterministically $Y'=(y;y_\nu \leftarrow 1)$.

2. $x' \neq x$ with $x'=(x;x_\nu \leftarrow -1)$. In this case we use the fact that $U_2 \leq \pi(x_\nu=-1|x_{-\nu})$, which gives that $\hat{U}_2$ is distributed Uniform$(0,\pi(x_\nu=-1|x_{-\nu}))$. A sample from this distribution is generated by sampling a Uniform(0,1) random variable $\hat{U}_2^*$ and set $\hat{U}_2=\hat{U}_2^* \times \pi(x_\nu=-1|x_{-\nu})$. From this we find that we set $Y'=(y;y_\nu \leftarrow -1)$ if $\hat{U}_2^* \leq \frac{\pi(y_\nu=-1|y_{-i})}{\pi(x_\nu=-1|x_{-\nu})}$, and $Y'=(y;y_\nu \leftarrow +1)$ otherwise.

3. We observe a hold, i.e $x=x'$. Here we may use that the hold was produced in the forward iteration of X by drawing (U_1',U_2'), such that $\phi(x,U_1',U_2')=x$. If the observed numbers (u_1',u_2') from the forward iteration are stored, we may set $\left(\hat{U}_1,\hat{U}_2\right)=(U_1',U_2')$ and thus obtain the sample $y'=\phi(y,u_1',u_2')$. To prove this we show that this gives a correct distribution for $(\hat{U}_1,\hat{U}_2)$ (this proof is only mentioned in Fill [2]). Consider the case where $U_1'=\nu$ and $U_2' \leq \pi(x_\nu=-1|x_{-\nu})$ such that $x=x'$ with $x_\nu=-1$. From the forward iteration of X we get the following conditional distribution for (U_1',U_2')

$$\begin{aligned}
\lambda\left(U_1',U_2'|\phi(x,U_1',U_2')=x\right) &= \lambda\left(\nu,U_2'|\phi(x,\nu,U_2')=x\right) \\
&= \lambda\left(\nu,U_2'|U_2' \leq \pi(x_\nu=-1|x_{-\nu})\right) \\
&= \lambda\left(\nu,U_2'\pi(x_\nu=-1|x_{-\nu})\right)
\end{aligned} \tag{37}$$

Similarly, from the backward iteration of $\tilde{X}$ we get

$$\begin{aligned}
\left(\hat{U}_1,\hat{U}_2\right) &\sim \lambda\left(U_1,U_2|\phi(x,U_1,U_2)=x\right) \\
&= \lambda\left(\nu,U_2\pi(x_\nu=-1|x_{-\nu})\right)
\end{aligned} \tag{38}$$

By assigning $U_2 = U_2'$ we get the correct distribution, and since we have observed u_2' we set $\hat{u_2} = u_2'$. We proceed similarly for the case $x_\nu = 1$.

5 Experiments with the Ising model

We have implemented both Propp-Wilson's and Fill's algorithms, and have done different tests for the algorithms. In all experiments we used the Gibbs sampler and the random site updating rule.

5.1 Empirical testing of Fill's algorithms

The algorithm of Fill has, according to Fill himself, never been implemented and tested, and it is therefore important verify the output from the implementation. We have considered the correctness of the algorithms through empirical testing, and have applied the χ^2 test for goodness of fit (see Appendix A). To perform such testing it is necessary to know the normalisation factor, and therefore testing in practice is restricted to small grids. Our test case is the Ising model with 3×3 cyclic grid. The sample space S consist of $2^{3\times 3} = 512$ possible configurations, so a substantive amount of simulations is necessary.

5.1.1 Empirical results

We performed the χ^2 test for different values of the parameter β. For each value of the parameters 400000 simulations were carried out, to ensure that the expected frequency for each outcome was greater than 5. The results is presented in Figure 4 and Table 1. The figure show the estimated point densities from the observed data, compared with the theoretical point densities. The outcomes are sorted according to the number of equal neighbours, configurations with more equal neighbours having higher index. The table shows the observed test statistics, p-values and conclusion. We find the results to be satisfactory, strongly supporting our claim of a correct implementation.

Table 1: Results from χ^2-test, Fill

β	Test statistic χ^2	Test statistic N(0,1)	pvalue	reject H_0
0.01	513.612	0.082	0.935	NO
0.11	500.131	-0.340	0.734	NO
0.21	524.474	0.421	0.673	NO
0.31	517.203	0.194	0.846	NO

5.1.2 A note on the necessity of testing computer programs

The purpose of performing the χ^2 test was to investigate whether our implementation of the algorithm truly returned correct samples. Indeed, a minor error in the implementation of Fill's algorithm was first revealed after performing the test. Figure 5a shows the results from the incorrect implementation compared with the theoretical frequencies. At first sight the results seems correct, however the χ^2 test rejected our null hypothesis (see Table 2). By sorting the configurations by both number of equal pixels *and* the number of white pixels, we found the samples to be minor biased in favour of configurations with more black pixels (see Figure 5b,c). A thorough investigation of the code then showed we had cancelled two unequal normalisations factors against each other, thereby leading to

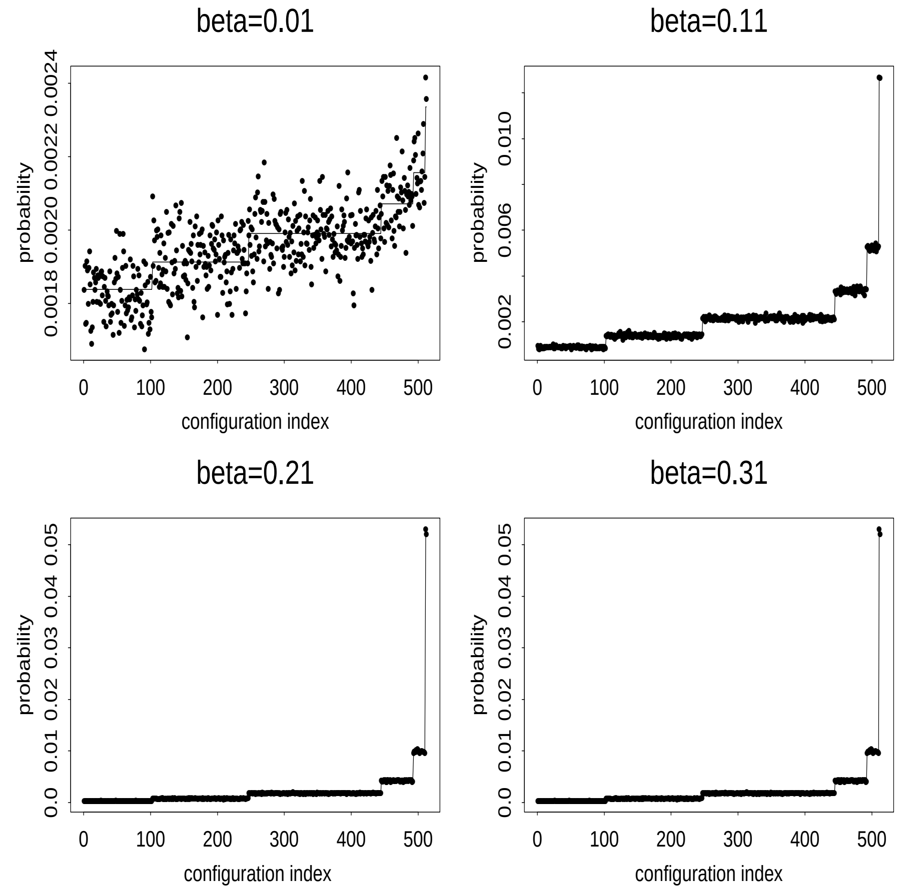


Figure 4: Results from testing Fill's algorithm with external field, 3×3 torroidial grid. Estimated point densities from observations (dots), compared with theoretical point densities (line).

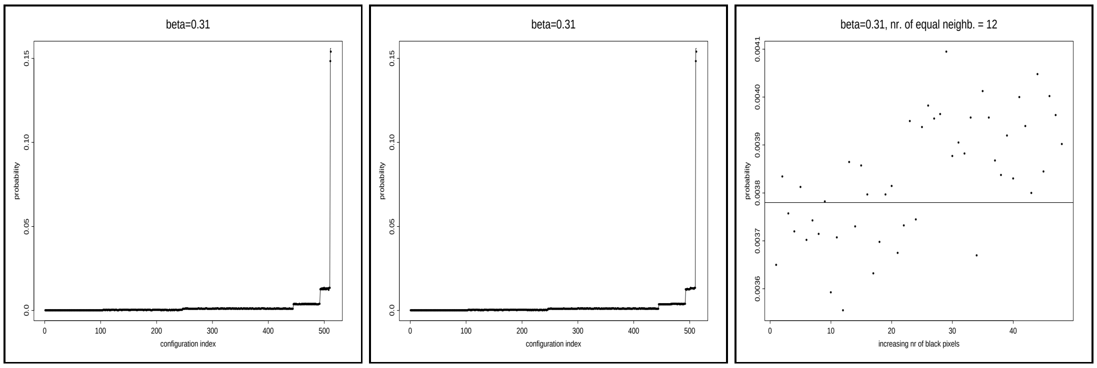
the observed results. This error was also more significant for greater β, explaining the acceptance of the null hypothesis for $\beta < 0.21$ (see Table 2).

5.2 Investigation of distribution of coupling time as function of grid size for Propp-Wilson algorithm

In this section we have done an empirical investigation of the coupling time T as a function of the size of the state space. We have considered an Ising model with no external field. The state space $\mathcal{S}$ has $2^{n_x \times n_y}$ possible configurations, where $n_x \times n_y$ is the number of elements in the grid $\mathcal{L}_\mathcal{D}$. We therefore expect the coupling time to increase heavily as a function of the grid size, and our hypothesis is that the $T = \mathcal{O}(n^\alpha)$ for some constant α. Taking logarithms gives

$$\log(T) = \alpha \log(n) + C \tag{39}$$

C being a constant. We have done testing on a quadratic grid with $n = n_y = n_x$. The observed coupling times, measured in number of iterations, is plotted i Figure 6. Note that it is the exact coupling times that we have observed. We have at every iteration checked if the two chains have coalesced.



a b c

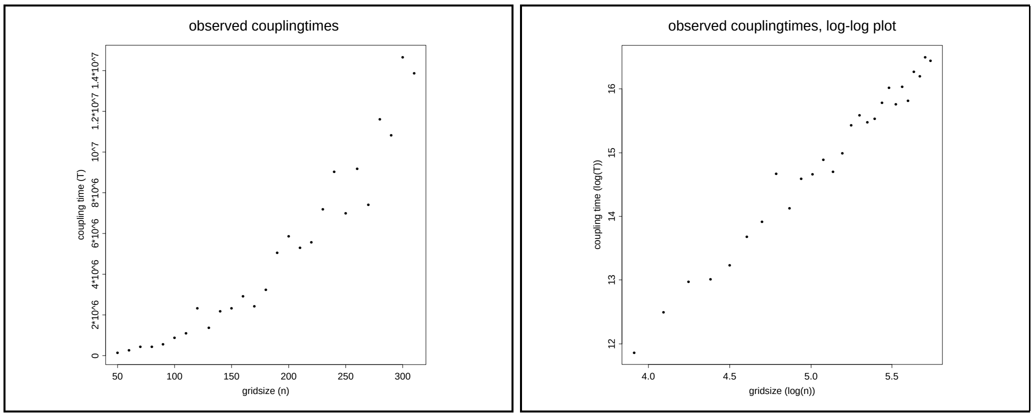
Figure 5: Estimated point density (dots) compared with theoretical point density (line) for the *incorrect* implementation of Fill's algorithm with $\beta = 0.31$: (*a*) configurations sorted just by number of equal neighbours, (*b*) configurations sorted by both number of equal neighbours and number of white pixels, (*c*) configurations with the same number of equal neighbours (in this example 12) sorted by number of white pixels.

Table 2: RESULTS FROM χ^2-TEST, INCORRECT IMPLEMENTATION.*When testing our first implementation of Fill's algorithm these results were obtained, with the rejection of the null hypothesis for* $\beta > 0.21$ *as conclusion.*

β	Test statistic χ^2	Test statistic N(0,1)	pvalue	reject H_0
0.01	506.085	-0.154	0.878	NO
0.11	545.613	1.083	0.279	NO
0.21	722.884	6.628	3.406e-11	YES !
0.31	907.820	12.413	2.229e-35	YES !

As Figure 6.b clearly shows it seems natural to assume a linear relationship between the logarithms. We performed a regression analysis with the statistical package *Splus*, and the Table 3 shows the results. The parameter α is estimated to 2.44, and the results also show that a linear model is a satisfactory approximation to the observed data. Figure 7 shows the predicted coupling times compared with the observations. We performed the testing for values of n up to 310, and we therefore have to be careful to make these conclusions yield for larger n.

It is clear that the coupling time increases with increasing grid size, and our model is $T = \mathcal{O}(n^\alpha)$ where α is estimated to 2.44. It is natural to suspect that also the variance of the coupling time increases with n. We performed therefore several tests for each grid size to compute the sample mean and variance. Figure 8 shows the results, and even if we only have data for $n \leq 100$, it seems clear that also the variance is increasing.

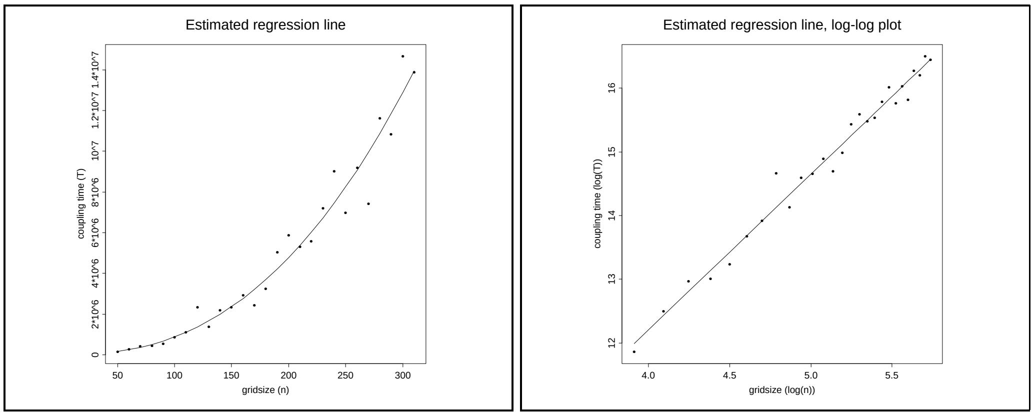


a b

Figure 6: (a) Observed coupling times as a function of the grid size, and (b) the observations on log scale.

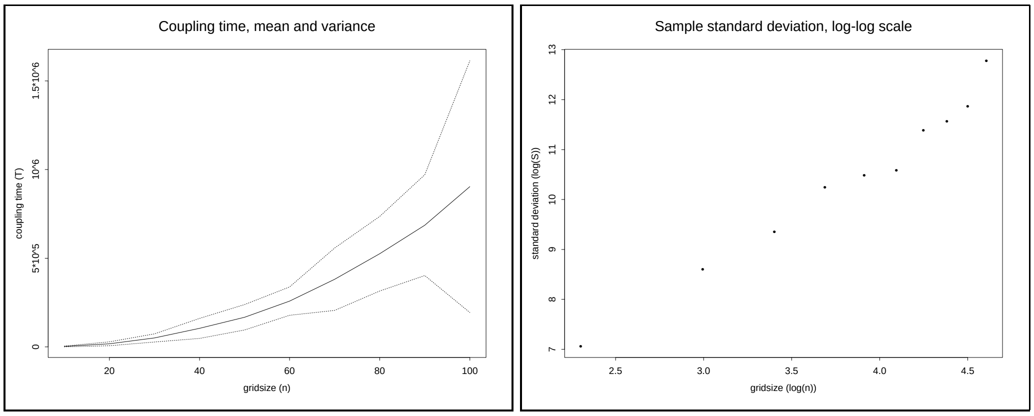
Table 3: REGRESSION ANALYSIS. *The output after the linear regression analysis for the coupling time T as function of grid size n, our model being $\log(T) = \alpha \log(n) + C$. The analysis is performed in Splus, and the $t - values$ is computed for the zero-hypothesis $H_0: \quad \alpha = 0, C = 0$.*

```
 Residuals:
     Min       1Q     Median       3Q    Max
 -0.2947 -0.1309 -0.006599 0.07663 0.5337

Coefficients:
              Value  Std. Error  t value  Pr(>|t|)
          C   2.4309   0.3422      7.1046   0.0000
      alpha   2.4440   0.0671     36.4499   0.0000

Residual standard error: 0.1786 on 25 degrees of freedom
Multiple R-Squared: 0.9815
F-statistic: 1329 on 1 and 25 degrees of freedom, the p-value is 0

Correlation of Coefficients:
           C
alpha   -0.9949
```



a b

Figure 7: Predicted coupling times in (a) natural scale, and (b) in logarithmic scale. The observations are also plotted.



a b

Figure 8: (a) The sample mean of 60 observed coupling times (solid line), and plus-minus two times the sample standard deviation (dotted lines). (b) The sample standard deviation in log-log scale.

6 PERFECT SAMPLING APPLIED IN IMAGE NOISE REDUCTION ALGORITHMS

Stochastic simulation and the Ising model may also be used for image noise reduction. A digital image may be represented as a Markov random field, and we first review some theory on stochastic image analysis.

6.1 STOCHASTIC IMAGE ANALYSIS

We consider binary images, i.e. images with pixels that are either black or white. This corresponds to spin systems with up or down spin. Such an image will typically be degraded during a transmission, e.g through a cable. Then each pixel, independent of all other pixels, will have a false value with some probability ϵ. This kind of degradation is called CHANNEL NOISE.

Stochastic image processing will typically adapt the Bayesian paradigm (see Winkler [16]). A priori, we state that an image is distributed according to a Markov random field, and we will see that the Ising model has shown to be useful. The major assumption in the choice of this prior is that we expect the image to be partitioned into a collection of domains, the pixel value being constant within each domain. As an example, consider images of objects on a background. Now, the main idea is that we simulate images from the posteriori distribution, conditioned on the noisy image. This will lead to samples upon which we can base inference, i.e. the maximum a posteriori (MAP) estimate, marginal posterior mode (MPM) estimate or minimum mean squares estimate, and corresponds to smoothening of the image. The main drawback of this method is that we may erase small existing objects or smooth fuzzy boundaries.

First, we introduce useful notation and give some definitions.

Let y be some noisy image, defined on the grid $\mathcal{L}_{\mathcal{D}}$ and with state space $S = \{y_i \in \{-1, 1\}; i \in \mathcal{L}_{\mathcal{D}}\}$. Let x be the true unknown image which we want to estimate. A priori, x is distributed as the Ising model with distribution function

$$\pi(x) = c \times \exp\left(\beta \sum_{<i,j>} x_i x_j\right) \tag{40}$$

The probability of observing y if x is the correct picture is denoted the LIKELIHOOD, and we write Prob$(Y = y|X = x) = L(y|x)$. In case of channel noise, with some probability ϵ a pixel y_i will have a false value, and with probability $(1 - \epsilon)$ the pixel value is correct. Since the noise at one pixel is independent of the all the other pixels, the y_i's may be considered as independent Bernoulli random variables with parameter ϵ and distribution

$$L(y_i|x_i) = \epsilon^{I_{y_i \neq x_i}} (1 - \epsilon)^{I_{y_i = x_i}} \tag{41}$$

where I is the indicator function. The likelihood function equals the joint distribution of the y_i's, and is given by

$$L(y|x) = \prod_{i \in \mathcal{L}_D} \epsilon^{I_{y_i \neq x_i}} (1 - \epsilon)^{I_{y_i = x_i}} = \epsilon^{\sum_i (1 - I_{y_i = x_i})} (1 - \epsilon)^{\sum_i I_{y_i = x_i}} \tag{42}$$

From Bayes formula we get the posterior distribution of x, given the observed but noisy image

$$\pi(x|y) \propto \pi(x)L(y|x) \tag{43}$$

By writing

$$\begin{aligned}
\log L(y|x) &= \log\left(\epsilon^{\sum_i (1-I_{y_i=x_i})}(1-\epsilon)^{\sum_i I_{y_i=x_i}}\right) \\
&= \sum_i \log\epsilon - \sum_i \log\epsilon I_{y_i=x_i} + \sum_i \log(1-\epsilon)I_{y_i=x_i} \\
&= \sum_i \log\epsilon + \sum_i \log\frac{1-\epsilon}{\epsilon}I_{y_i=x_i}
\end{aligned} \tag{44}$$

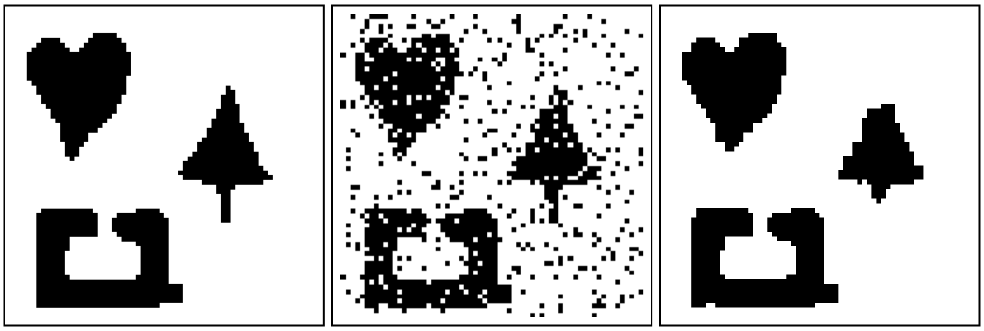
we get

$$\begin{aligned}
\pi(x|y) &\propto \exp\left(\beta\sum_{<i,j>} x_i x_j + \sum_i \log\epsilon + \sum_i \log\frac{1-\epsilon}{\epsilon}I_{y_i=x_i}\right) \\
&\propto \exp\left(\beta\sum_{<i,j>} x_i x_j + \sum_i \log\frac{1-\epsilon}{\epsilon}I_{y_i=x_i}\right)
\end{aligned} \tag{45}$$

and hence the posterior distribution may be thought of as an Ising distribution with an external field. Propp-Wilson's and Fill's methods may then be applied to the Gibbs sampler to simulate images from the posterior distribution.

There exists a lot of other types of noise, and the methods of Propp and Wilson may be used as long as $y_i = \psi(x_i, U_i)$ for some iid random variables U_i and function ψ. As an example, we investigate WHITE GAUSSIAN NOISE. We assume that each pixel value y_i in the observed picture equals the true pixel value x_i plus some Gaussian noise variable η_i with expectation 0 and variance σ^2. This gives the posteriori distribution of the image proportional to

$$\pi(x|y) \propto \exp\left(\beta\sum_{<i,j>} x_i x_j - \frac{1}{2\sigma^2}\sum_i (y_i - x_i)^2\right) \tag{46}$$



a: Original image b: noisy image c: MPM estimate

Figure 9: Here is an example of how an Ising model may be used for noise reduction. The noise level is ten percent, which equals $\epsilon = 0.1$ The samples are produced with Propp-Wilson's algorithm. The β parameter is set to 1.1, which means a high degree of attractiveness. The MPM estimate (c) is computed from 70 independent samples.

6.2 Experiments with image noise reduction

We have implemented the algorithms of Propp-Wilson and Fill in an image noise reduction program. The program is tested on a binary image of 64×64 pixels, and synthetic channel noise is laid upon the picture. Each pixel is independently flipped with probability $\epsilon = 0.1$.

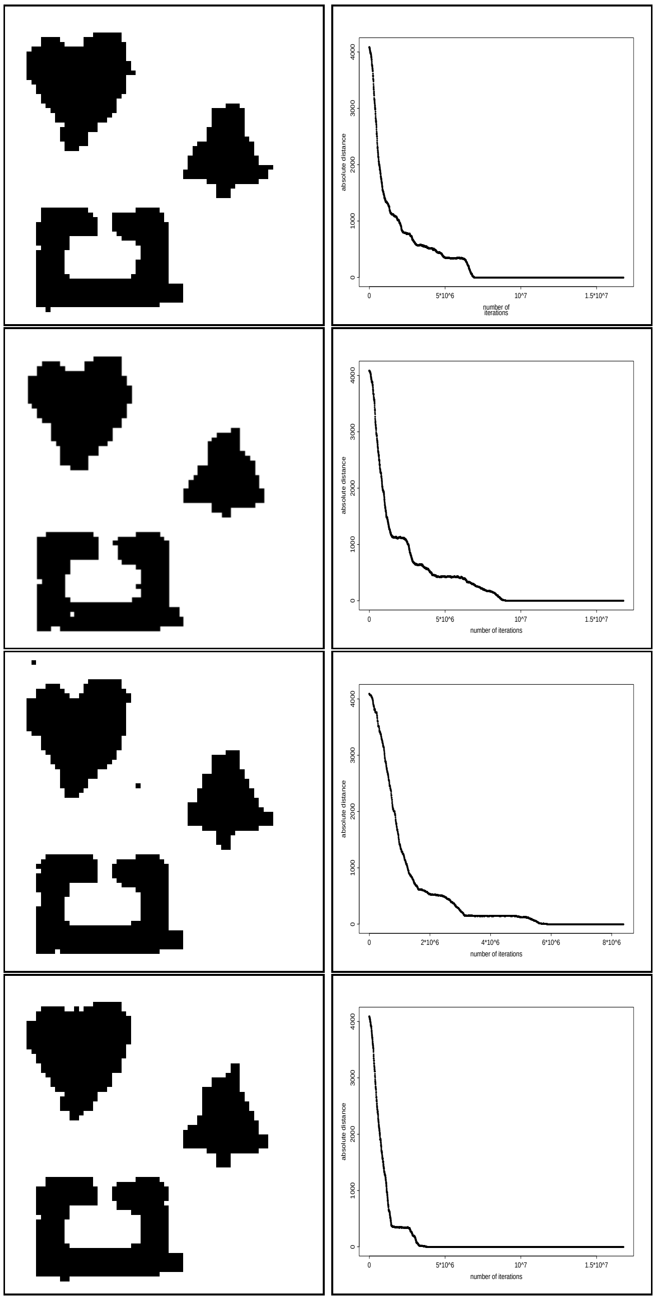
6.2.1 The Propp-Wilson algorithm

The original and noisy image is shown in Figure 9a,b. Some independent samples from the posteriori distribution is shown in Figure 10, together with the difference $D(upper, lower)$ between the two Markov chains $upper$ and $lower$, starting in respectively all pixels white and black. The difference at each step is defined as the number of unequal pixels in the two pictures, i.e

$$D(upper, lower) = \sum_{i=1}^{n} I_{upper_i \neq lower_i} \tag{47}$$

6.2.2 Fill's algorithm

The noisy image and some samples from the posteriori distribution is shown in Figure 11. The simulation of the pictures with Fill's algorithm seemed to take longer time than for the Propp-Wilson simulation.



Sampled image Absolute distance between chains

Figure 10: Sampled images using the Propp-Wilson algorithm, together with the absolute distance between the two chains, starting at all pixels white and all pixels black respectively, as a function of number of iterations performed.

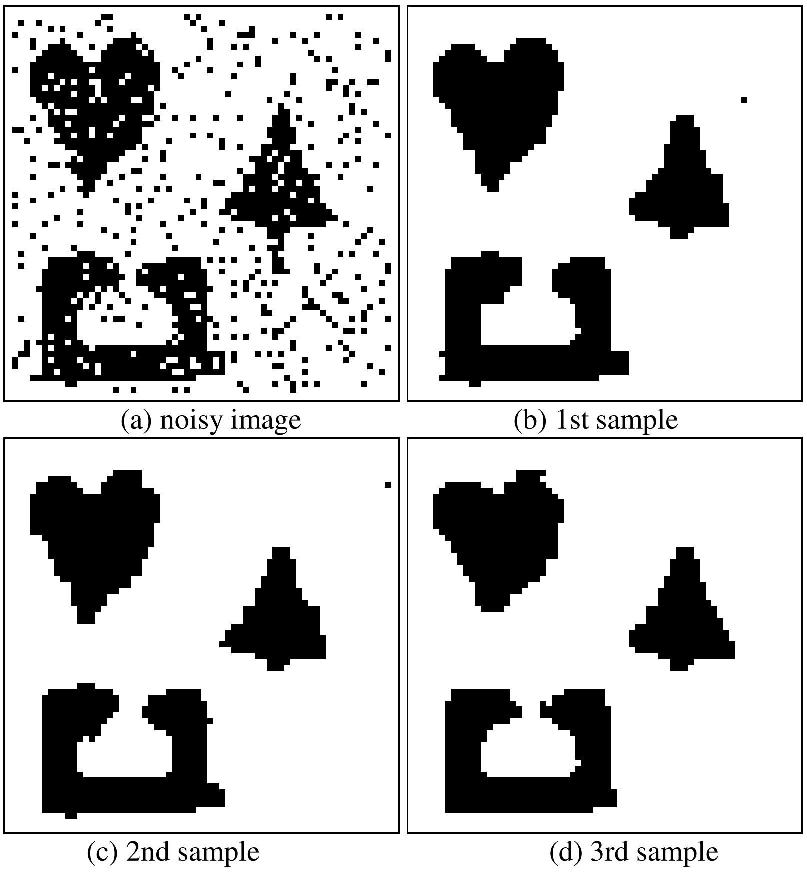


Figure 11: Results from the implementation of Fill's algorithm. (a): The noisy image, (b),(c) and (d): some samples from the posteriori distribution.

7 PERFECT ALGORITHMS FOR THE CONTINUOUS CASE

Markov chain simulation may also be used for sampling from a continuous probability density π on the state space $\mathcal{S}$. The transition matrix P is then replaced with a TRANSITION KERNEL f such that

$$\int \pi(x)f(y|x)dx = \pi(y) \tag{48}$$

and where the transition kernel has to be irreducible and aperiodic. Let $\{X_t\}$ denote the Markov chain. As with the discrete case, the updating at each step of the Markov chain is done via an updating function $\phi : \mathcal{S} \times \mathcal{U} \to \mathcal{S}$ such that

$$X_{t+1} = \phi(x_t, U_{t+1}) \tag{49}$$

and

$$\text{Prob}\,(\phi(x, U) \leq y) = \int_{-\infty}^{y} f(y'|x)dy' \tag{50}$$

To achieve perfect samples from π one use the same idea of coupling from the past as Propp-Wilson has used with the discrete case. However, one problem immediately arises since the state space consists of an infinite number of possible states, and chains with different starting values will therefore not coalesce in finite time. Murdoch and Green [9] has therefore developed several exact sampling algorithms which introduces a discreteness into the state space. Their idea is to construct algorithms that update sets of states into one single state. Let $\{X_t\}$ and $\left\{\tilde{X}_t\right\}$ be two Markov chains with same transition kernel f, and let $X_{t-1} = x_1, \tilde{X}_{t-1} = x_2$ where $x_1 \neq x_2$. Suppose these two chains coalesce in step t, such that $X_t = \tilde{X}_t = y$. A necessary and sufficient condition for this to occur is that $\phi(x_1, U_t) = \phi(x_2, U_t) = y$. It is clear that if we can construct an algorithm such that with probability 1 the updating at a random step in the iteration is done through an updating function ψ such that $\phi(x, U_t) = \psi(U_t)$ for all x, then all chains will simultaneously coalesce at this step.

7.1 THE ALGORITHMS

We will now introduce the algorithms of Murdoch and Green [9]. Assume we are starting a Markov chain in every possible state at time $-M$. Since we have a continuous state space, the set of current states which the Markov chains possess at this point will be uncountable. Generally we denote the set of states the Markov chains possess at the current step t by $\mathcal{R}_t$, and we set $\mathcal{R}_{-M} = \mathcal{S}$. The algorithms will reduce $\mathcal{R}_t$ to a countable set, and if $\mathcal{R}_0$ consists of a single element ($\#\mathcal{R}_0 = 1$) then this will be a sample from the target density.

7.1.1 THE MULTI-GAMMA COUPLER

The multi-gamma coupler may be used if the transition kernel f is explicitly known. Let $r(y) \geq 0$ be a lower bound on the transition kernel $f(y|x)$ such that

$$f(y|x) \geq r(y) \quad \forall x, y \tag{51}$$

Define $\rho = \int r(y')dy'$ and $q(y|x) = f(y|x) - r(y)$. Further, define

$$R(y) = \rho^{-1} \int_{-\infty}^{y} r(y')dy', \tag{52}$$

$$Q(y|x) = (1-\rho)^{-1} \int_{-\infty}^{y} q(y'|x)dy', \tag{53}$$

and the updating functions $\psi(U)$ and $\kappa(x, U)$ such that

$$\text{Prob}(\psi(U) \leq y) = R(y), \tag{54}$$

$$\text{Prob}(\kappa(x, U) \leq y) = Q(y|x) \tag{55}$$

The transition kernel $f(y|y)$ can be seen as a mixture of the distributions $r(y)/\rho$ and $(1-\rho)^{-1}q(y|x)$, where the updating function is

$$\phi(x, U) = \begin{cases} \psi(U^{(2)}) & : \quad U^{(1)} \leq \rho \\ \kappa(x, U^{(2)}) & : \quad U^{(1)} > \rho \end{cases} \tag{56}$$

To prove this, compute

$$\text{Prob}(\phi(x, U) \leq y) = \rho R(y) + (1-\rho)Q(y|x) = \int_{-\infty}^{y} f(y'|x)dy' \tag{57}$$

The multi-gamma coupler algorithm is best illustrated through the set $\mathcal{R}_t$ of states the Markov chains possess at time t. The iteration is started at time $-M$, with $\mathcal{R}_{-M} = \mathcal{S}$. In each step t the updating is with probability ρ done via the function $\psi(U_t)$, and the first time this occurs $\mathcal{R}_t$ will shrink to a single state. In all other steps the updating is done via $\kappa(\cdot, U_t)$, but before $\mathcal{R}_{t-1}$ has become countable we just set $\mathcal{R}_t = \mathcal{S}$. If $\mathcal{R}_0$ do not consists of a single state we restart the iteration at time $-2M$. This is similar to the "coupling from the past" algorithm of Propp and Wilson for the discrete case. A pseudo-code for this algorithm is as follows.

Multigamma(M):

```
t ← −M
R_t ← S
while R_t = S and t < 0
    t ← t + 1
    if U_t^(1) < ρ
        R_t ← ψ(U_t^(2))
    else
        R_t ← S
while t < 0
    t ← t + 1
    x ← R_{t−1}
    if U_t^(1) < ρ
        R_t ← ψ(U_t^(2))
    else
        R_t ← κ(x, U_t^(2))
if #R_0 = 1
    return R_0
else
    return Multigamma(2M)
```

The search of M can in fact be avoided by considering the number T between each time the Markov chains are updated via $\psi(U)$ as an random variable, where $T+1$ is geometrically distributed with success probability ρ.

$$\text{Prob}(T=t) = \rho(1-\rho)^t; \quad t = 0, 1, \ldots \tag{58}$$

Consider the Markov chain starting at time $-\infty$ and ending at time 0. We know that the distribution of X_0 equals the target distribution. Also, seen from time 0 and backwards, the number T of steps to the first time the Markov chains are updated via $\psi(U)$ has distribution (58). We may sample T from this distribution, and thus we know that $X_{-T} = \psi(U)$. The Markov chain is then propagated according to $X_t = \kappa(x_{t-1}, U_t)$ to time 0. The output X_0 will then be a perfect sample from π. Note that the expected time to coupling is $1/\rho$, and therefore will a small $r(\cdot)$ obstruct this algorithm of being of any practical use. Further, if $r(\cdot) \equiv 0$ the algorithm cannot be used at all.

7.1.2 The Partitioned Multi-gamma Coupler

Even if it is difficult to find a lower bound on the transition kernel f the state space $\mathcal{S}$ can sometimes be partitioned into m disjoint sets $\mathcal{A}_1, \ldots, \mathcal{A}_m$, where a bound can be placed on the transition kernel in each set. We then have

$$f(y|x) \geq r_i(y) \geq 0 \quad \forall x \in \mathcal{A}_i, \quad \forall y \in \mathcal{S} \tag{59}$$

and $q_i(y|x) = f(y|y) - r_i(y)$. We have $\rho_i = \int r_i(y')dy'$, and as with the multi-gamma coupler we define

$$R_i(y) = \rho_i^{-1} \int_{-\infty}^{y} r_i(y')dy', \tag{60}$$

$$Q_i(y|x) = (1-\rho_i)^{-1} \int_{-\infty}^{y} q_i(y'|x)dy', \tag{61}$$

and

$$\text{Prob}(\psi_i(U) \leq y) = R_i(y), \tag{62}$$

$$\text{Prob}(\kappa_i(x,U) \leq y) = Q_i(y|x) \tag{63}$$

The updating of the Markov chain at step t is done through

$$\phi(x_{t-1}, U_t) = \begin{cases} \psi_i(U_t^{(2)}) & : \quad U_t^{(1)} \leq \rho_i \\ \kappa_i(x_{t-1}, U_t^{(2)}) & : \quad U_t^{(1)} > \rho_i \end{cases} \tag{64}$$

when $x_{t-1} \in \mathcal{A}_i$. Initially we have $\mathcal{R}_{-M} = \mathcal{S} = \cap_{i=1}^{m} \mathcal{A}_i$. In each step t the updating of chains with current state $x_{t-1} \in \mathcal{A}_i$ is with probability ρ_i done via the functions $\psi_i(U_t)$, and in all other steps the updating is done via $\kappa_i(\cdot, U_t)$. If the the chains are updated via $\psi_i(U_t)$ all these chains will couple to the state $\psi_i(U_t)$. Since the Markov chain is irreducible states in different partitions will communicate, and therefore $\mathcal{R}_t$ will shrink to a single element in finite time.

The following pseudo-code is given for the special case with $\rho_i = \rho$ for all subsets $\mathcal{A}_i$. The first time the chains are updated via $\psi_i(U_t)$ then $\mathcal{R}_t$ will be shrink to a finite set with generally m elements, one for each partition $\mathcal{A}_i$. The next time this occurs all chains with current state $x_{t-1} \in \mathcal{A}_i$ will couple to the state $\psi_i(U_t)$

Partitioned-Multigamma(M):

$t \leftarrow -M$
$\mathcal{R}_t \leftarrow \mathcal{S}$
while $\mathcal{R}_t = \mathcal{S}$ and $t < 0$
    $t \leftarrow t + 1$
    if $U_t^{(1)} < \rho$
        $\mathcal{R}_t \leftarrow \left\{\psi_i(U_t^{(2)}), i = 1, \ldots, m\right\}$
    else
        $\mathcal{R}_t \leftarrow \mathcal{S}$
while $t < 0$
    $t \leftarrow t + 1$
    if $U_t^{(1)} < \rho$
        $\mathcal{R}_t \leftarrow \left\{\psi_i(U_t^{(2)}) : \mathcal{A}_i \cap \mathcal{R}_{t-1} \neq \emptyset\right\}$
    else
        $\mathcal{R}_t \leftarrow \cup_{i=1}^m \left\{\kappa_i(x, U_t^{(2)}) : x \in \mathcal{A}_i \cap \mathcal{R}_{t-1}\right\}$
if $\#\mathcal{R}_0 = 1$
    return $\mathcal{R}_0$
else
    return Partitioned-Multigamma($2M$)

7.1.3 THE REJECTION COUPLER

In many MCMC problems we do not know the normalised transition kernel $f(y|x)$, only some unnormalised distribution $g(y|x)$ where

$$f(y|x) = k(x)g(y|x), \quad \forall x, y \in \mathcal{S} \tag{65}$$

As an example, such situations arise frequently when the transition kernel is some conditional distribution only known to be proportional to another distribution, e.g. the joint distribution. This is quite common in Bayesian models.

Suppose we in addition to a lower bound, $r(y)$, may find an upper bound, $h(y)$, on the unnormalised transition kernel, such that

$$0 \leq r(y) \leq g(y|x) \leq h(y), \quad \forall x, y \in \mathcal{S} \tag{66}$$

where $\nu = \int h(y)dy < \infty$ and $\rho = \int r(y)dy / \int h(y)dy > 0$. From this we can construct a rejection sampler for $f(y|x)$ using the proposal $h(y)/\nu$ with update function $\psi(U)$. We get the upper bound

$$\frac{f(y|x)}{h(y)/\nu} = \frac{k(x)g(y|x)}{h(y)/\nu} \leq \frac{k(x)}{\nu} = L \tag{67}$$

and a hence sample y is accepted to come from $f(y|x)$ with the probability

$$\frac{f(y|x)}{Lh(y)/\nu} = \frac{g(y|x)}{h(y)} \tag{68}$$

Since $r(y) \leq g(y|x)$ we also get that a new state y is accepted with *at least* probability $r(y)/h(y)$ for all possible current states $x \in \mathcal{S}$.

The idea behind the rejection sampler is, at each step, to propose J new states $y_j = \psi(U_j),\ j = 1,\ldots,J$ where J is determined randomly such that y_J under all circumstances had been accepted for all possible states $x \in \mathcal{S}$. We check this by comparing a uniform $(0,1)$ random variable v_j with $r(y_j)/h(y_j)$, and the proposed state y_j is, regardless of the current state, accepted if $v_j \leq r(y_j)/h(y_j)$. This corresponds to assign to each y_j a partition $\mathcal{B}_j$ of $\mathcal{S}$, where $\mathcal{B}_j$ consists of the intersection of those states for which we would accept y_j as the new state and those states which not would give accept for $y_1,\ldots,y_{j-1}$ [1]. As usual we have $\mathcal{R}_{-M} = \mathcal{S}$, and therefore we set $\mathcal{R}_{-M+1} = \{y_1,\ldots,y_J\}$. This means that set of current states will be finite in the first step of the Markov chain. For all other steps t, we set $\mathcal{R}_t = \{y_j : \mathcal{B}_j \cap \mathcal{R}_{t-1} \neq \emptyset\}$. The sets $\mathcal{B}_j,\ j = 1\ldots,J$ is not explicitly computed, but given implicit as we update a state $x \in \mathcal{R}_{t-1}$ to the first y_j where $v_j \leq g(y_j|x)/h(y_j)$. If $\mathcal{R}_0$ consists of a single state then this will be our sample, else we restart at time $-2M$.

Rejection(M):

```
t ← −M
R_t ← S
j ← 0
repeat
    j ← j + 1
    y_j ← ψ(U_j)
    if v_j ≤ r(y_j)/h(y_j)
        J ← j
        break
t ← t + 1
R_t ← {y_1, ..., y_J}
while t < 0
    t ← t + 1
    j ← 0
    repeat
        y_j ← ψ(U_j)
        if v_j ≤ r(y_j)/h(y_j)
            J ← j
            break
    R_t ← {y_j : B_j ∩ R_{t−1} ≠ ∅}
if #R_0 = 1
    return R_0
else
    return Rejection(2M)
```

The expected number of states in $\mathcal{R}_{-M+1}$ is $1/\rho$, and if the bounds are too loose this number will be too large to be of practical interest. Murdoch and Green [9] show how to apply partitioning in advance to the state space, similar to the partitioned multi-gamma coupler.

[1] Note a minor inaccuracy in the paper of Murdoch and Green [9], since they say that $\mathcal{B}_j$ consists of the intersection of those states which would give accept for y_j and all those states which not would give accept for y_{j-1}.

7.1.4 THE PARTITIONED REJECTION COUPLER

Suppose we can partition the state space $\mathcal{S}$ into m disjoint sets $\mathcal{A}_i$, $i = 1, \ldots, m$ and that we can find upper and lower bounds on $g(y|x)$ within each cell such that

$$0 \leq r_i(y) \leq g(y|x) \leq h_i(y) \quad \forall x \in \mathcal{A}_i, \quad \forall y \in \mathcal{S} \tag{69}$$

where $\nu_i = \int h_i(y)dy < \infty$ and $\rho_i = \int r_i(y)dy / \int h_i(y)dy > 0$. Now, within each cell $\mathcal{A}_i$ we proceed exactly as we do with the rejection sampler for the whole state space. We propose J_i new states $y_{i1}, \ldots, y_{iJ_i}$ from $h_i(y)/\nu_i$ such that y_{iJ_i} under all circumstances had been accepted for all current states $x \in \mathcal{A}_i$. Again, this corresponds to implicitly partition $\mathcal{A}_i$ in J_i sets $\mathcal{B}_{ij}$, $j = 1, \ldots, J_i$ such that a current state $x \in \mathcal{B}_{ij}$ is updated to the value y_{ij}. In practice this is done by updating $x \in \mathcal{A}_i$ to the first y_{ij} in $(y_{i1}, \ldots, y_{iJ_i})$ where $v_{ij} \leq g(y_{ij}|x)/h_i(y_{ij})$. Given a current set of states $\mathcal{R}_{t-1}$ these states are updated to $\mathcal{R}_t = \{y_{ij} : \mathcal{B}_{ij} \cap \mathcal{R}_{t-1} \neq \emptyset\}$

Partitioned-Rejection(M):

```
t ← −M
R_t ← S
for i = 1, ..., m
    j ← 0
    repeat
        j ← j + 1
        y_ij ← ψ_i(U_j)
        if v_ij ≤ r_i(y_ij)/h_i(y_ij)
            J_i ← j
            break
t ← t + 1
R_t ← {y_ij, i = 1, ..., m   j = 1, ..., J_i}
while t < 0
    t ← t + 1
    for i = 1, ..., m
        j ← 0
        repeat
            j ← j + 1
            y_ij ← ψ_i(U_j)
            if v_ij ≤ r_i(y_ij)/h_i(y_ij)
                J_i ← j
                break
    R_t ← {y_ij : B_ij ∩ R_{t−1} ≠ ∅}
if #R_0 = 1
    return R_0
else
    Partitioned-Rejection(2M)
```

This gives an expected number of states equal to $\sum_i^m = 1/\rho_i$, which can be much smaller than $1/\rho$ with a clever partition.

7.1.5 The Metropolis Hastings Coupler

In this section we consider the case where the stationary distribution π is known up to a constant of proportionality, but where the kernel density f is not explicitly known. As explained in section 2.2 we may then use the Metropolis-Hastings coupler with proposal distribution $q(y|x)$. A new proposed state y is accepted with probability $\alpha_{x,y}$, where $\alpha_{x,y}$ is given in equation 9. If the proposal is rejected we keep the current state x. As usual we run Markov chains from time $-M$ to time 0, starting from every possible state $x \in \mathcal{S}$, and outputs x_0 if all chains have coalesced. To sample from $q(y|x)$ we use the PROPOSAL FUNCTION $\psi(x, U)$ such that

$$\text{Prob}(\psi(x,U) \leq y) = \int_{-\infty}^{y} q(y'|x)dy' \tag{70}$$

Let $\mathcal{R}_{t-1} \subseteq \mathcal{S}$ denote the set of states at time $t-1$, and let $\mathcal{C}_t \subseteq \mathcal{R}_{t-1}$ denote the set of states which are not updated at step t. The set of states at time t is then given by $\mathcal{R}_t = \mathcal{C}_t \cup \mathcal{F}_t$, where $\mathcal{F}_t = \{\psi(x, U_t) : x \in \mathcal{R}_t - \mathcal{C}_t\}$. Suppose the function $\psi(\cdot, u)$ takes on only a finite number of values when u is fixed. It is then clear that the whole set $\mathcal{R}_t$ of current states consist of a union of a finite set $\mathcal{F}_t$ and a possibly uncountable set $\mathcal{C}_t$. Since $q(y|x) > 0 \quad \forall x, y$, with probability 1 the set $\mathcal{R}_t$ will be finite, and since irreducibility is required $\mathcal{R}_t$ will shrink to a single state in finite time. For a general pseudo-code, see Murdoch and Green [9].

To apply this algorithm we must calculate the set $\mathcal{C}_t$ and, as Murdoch and Green [9] point out, this is not always simple. We therefore consider the special case with the INDEPENDENCE SAMPLER with $q(y|x) = q(y)$, i.e. a proposal distribution that do not depend on the current state. All states that are updated are in this case updated to a single state y, with $\mathcal{R}_t$ consisting of the single state y and the set $\mathcal{C}_t$. If we can find an upper bound L such that

$$\sup_x \frac{\pi(x)}{q(x)} \leq L < \infty \tag{71}$$

then this implies that

$$\alpha_{xy} \geq \frac{\pi(y)}{q(y)L} \tag{72}$$

and hence if $U < \frac{\pi(y)}{q(y)L}$ all chains have simultaneously coalesced, and we set $\mathcal{R}_t = y$. Otherwise we set $\mathcal{R}_t$ equal to the whole state space. When the coupling occurs the sampled value y itself will be an exact sample from π, with no need for running the Markov chain to time 0. To see this, just consider the algorithm as an rejection sampler with proposal q and acceptance probability $\frac{\pi}{qL}$.

7.2 The Gibbs sampler for the continuous case

Murdoch and Green [9] also show how the multi-gamma coupler and rejection coupler may be applied as building blocks for sampling a vector $X = (X_1, \ldots, X_n)$, with state space $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$, using a cyclic Gibbs sampler. The updating density for each component of state vector is then the conditional density for the component given the other components. We denote the set of current states (as described in section 7.1) for each component X_i by $\mathcal{R}_{it}$. The set of current states for the whole state vector is then $\mathcal{R}_t = \mathcal{R}_{1t} \times \cdots \times \mathcal{R}_{nt}$, an n-dimensional set. To get a perfect sample, we require that that $\#\mathcal{R}_{it} = 1$ for each component X_i. To achieve this we could apply a multigamma or rejection coupler to each component. Suppose however that a component X_i only depends on component X_j. If $\mathcal{R}_j$ finite, then $\mathcal{R}_i$ will finite with the same or less number of states than $\mathcal{R}_j$. It we apply one of the coupling algorithms to X_j it then suffices to update X_i using the usual conditional density. As long as $\mathcal{R}_{jt}$ is uncountable we just set $\mathcal{R}_{it} = \mathcal{S}_i$, else we set $\mathcal{R}_{it}$ to the set of samples we receive by sampling X_i conditional on the states in $\mathcal{R}_{jt}$.

8 A CASE STUDY: SURGICAL FAILURE RATES

8.1 THE PROBLEM

The data and model for this example is collected from the BUGS example manual [14], and considers mortality rates for cardiac surgery in hospitals. Data are collected from $k = 12$ hospitals, and these data are shown in Table 4. The number of operations is n_i, and we model the number r_i of deaths as

Table 4: SURGICAL DATA. *The number of respectively operations and deaths and the corresponding failure rates in 12 hospitals*

Hospital	No. of deaths	No. of operations	failure rate
A	0	47	0
B	18	148	0.122
C	8	119	0.067
D	46	810	0.057
E	8	211	0.038
F	13	196	0.066
G	9	148	0.061
H	31	215	0.144
I	14	207	0.068
J	8	97	0.082
K	29	256	0.113
L	24	360	0.067

a binomial random variable with "success" probability p_i

$$r_i|n_i, p_i \sim \text{Binomial}(p_i, n_i) = \binom{n_i}{r_i} p_i^{r_i}(1-p_i)^{n_i-r_i}, \quad i = 1, \ldots, 12 \tag{73}$$

From data we have calculated the failure rate for each hospital. However, we have no idea of the uncertainty in these estimates. Also, we may suspect that the failure rates are similar in some way. If we suppose p_i have some prior distribution we can make inference of p_i and the parameters in the priori distribution by sampling from the posteriori distribution.

8.2 THE BETA MODEL

A natural choice for the prior is the beta distribution

$$p_i \sim \text{Beta}(\alpha, \beta) \tag{74}$$

If $\alpha = \beta = 1$ is known we have the non informative prior Uniform(0,1), and this corresponds to the case where the p_i's is assumed to be independent (fixed effect). The posteriori distribution of p_i is then Beta$(r_i + 1, n_i - r_i + 1)$. By sampling from this distribution we get estimates for the failure rates and credibility intervals.

A more realistic case is to assume a random effects model, where α and β have some non-informative prior distributions. Typically we set

$$\alpha \sim \text{Gamma}(a, b) \tag{75}$$

$$\beta \sim \text{Gamma}(c, d) \tag{76}$$

This model can be seen graphically as the directed acyclic graph in Figure 12. The posteriori distribu-

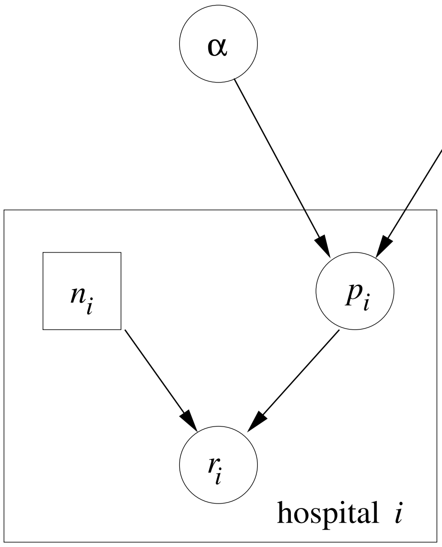


Figure 12: Directed acyclic graph for the *Beta*-model

tion is then proportional to

$$\begin{aligned}\pi(p, \alpha, \beta | n, r) &\propto \frac{b^a d^c}{\Gamma(a)\Gamma(b)} \alpha^{a-1} e^{-b\alpha} \beta^{c-1} e^{-d\beta} \\ &\quad \times \prod_{i=1}^{k} \binom{n_i}{r_i} p_i^{r_i} (1 - p_i)^{n_i - r_i} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p_i^{\alpha-1} (1 - p_i)^{\beta-1} \\ &\propto \alpha^{a-1} e^{-b\alpha} \beta^{c-1} e^{-d\beta} \prod_{i=1}^{k} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p_i^{\alpha + r_i - 1} (1 - p_i)^{\beta + n_i - r_i - 1}\end{aligned} \tag{77}$$

We first investigated if we could use the multigamma or rejection coupler as building blocks in a Gibbs sampler. The distribution of p_i conditional on α and β is a Beta$(\alpha + r_i, \beta + n_i - r_i)$. Since α and β can be arbitrary large we can not find a lower bound on this distribution. The posteriori distributions of α and β conditional on the p_i's is rather intractable, and since these distributions is proportional to p_i and $1 - p_i$ we can not find a lower bound on these distributions. The next step is to consider the Metropolis-Hasting coupler using an independence sampler $q(p, \alpha, \beta | n, r)$. A natural choice is to sample α and β from their prior distributions and sample the p_i's from the posteriori distribution. The independence sampler is then proportional to

$$q(p, \alpha, \beta | n, r) \propto \alpha^{a-1} e^{-b\alpha} \beta^{c-1} e^{-d\beta} \prod_{i=1}^{k} \frac{\Gamma(\alpha + \beta + n_i)}{\Gamma(\alpha + r_i)\Gamma(\beta + n_i - r_i)} p_i^{\alpha + r_i - 1} (1 - p_i)^{\beta + n_i - r_i - 1} \tag{78}$$

which gives the ratio

$$\begin{aligned}\frac{\pi(p,\alpha,\beta|n,r)}{q(p,\alpha,\beta|n,r)} &= \prod_{i=1}^{k}\frac{\Gamma(\alpha+r_i)\Gamma(\beta+n_i-r_i)\Gamma(\alpha+\beta)}{\Gamma(\alpha+\beta+n_i)\Gamma(\alpha)\Gamma(\beta)} \\ &= \prod_{i=1}^{k}\frac{(\alpha+r_i-1)\cdots\alpha\Gamma(\alpha)(\beta+n_i-r_i-1)\cdots\beta\Gamma(\beta)\Gamma(\alpha+\beta)}{(\alpha+\beta+n_i-1)\cdots(\alpha+\beta)\Gamma(\alpha+\beta)\Gamma(\alpha)\Gamma(\beta)} \\ &= \prod_{i=1}^{k}\frac{(\alpha+r_i-1)\cdots\alpha(\beta+n_i-r_i-1)\cdots\beta}{(\alpha+\beta+n_i-1)\cdots(\alpha+\beta)}\end{aligned} \tag{79}$$

Note that this is a function of α and β only, which is explained by the fact that the p_i's are sampled from their correct posteriori conditional distributions. The question is now whether we can find an upper bound on this function. By independently considering each part of this product we find that

$$\begin{aligned}\frac{(\alpha+r_i-1)\cdots\alpha(\beta+n_i-r_i-1)\cdots\beta}{(\alpha+\beta+n_i-1)\cdots(\alpha+\beta)} &\leq \frac{(\alpha+r_i-1)^{r_i}(\beta+n_i-r_i-1)^{n_i-r_i}}{(\alpha+\beta)^{n_i}} \\ &= S_i(\alpha,\beta)\end{aligned} \tag{80}$$

Maximising $S_i(\alpha,\beta)$ when one of the parameters is held fixed gives the relation between α and β at maximum, and we get (for fixed α) $\beta = \beta(\alpha) = ((n_i - r_i)/r_i)\alpha + n_i - n_i/r_i$. Some tedious calculations show that the function $S_i(\alpha,\beta(\alpha))$ is an increasing function of α and bounded by

$$\lim_{\alpha\to\infty} S_i(\alpha,\beta(\alpha)) = (\frac{r_i}{n_i})^{r_i}(1-\frac{r_i}{n_i})^{n_i-r_i} \tag{81}$$

Note that in the case $r_i \in \{0, n_i\}$ we get the upper bound 1, which is found directly from (79). A sample from the independence sampler (78) is therefore accepted with probability

$$\prod_{i=1}^{k}\frac{(\alpha+r_i-1)\cdots\alpha(\beta+n_i-r_i-1)\cdots\beta}{(\alpha+\beta+n_i-1)\cdots(\alpha+\beta)L_i} \tag{82}$$

where L_i is defined as

$$L_i = \begin{cases} (r_i/n_i)^{r_i}(1-r_i/n_i)^{n_i-r_i} & : \quad r_i \neq 0, n_i \\ 1 & : \quad r_i = 0, n_i \end{cases} \tag{83}$$

Unfortunately this bound was too large to be of practical interest. "Good" values of α and β gave an acceptance rate in the order 10^{-7}, while the expected value of α and β (with respect to their prior distributions) gave an acceptance rate in the order 10^{-20}. We therefore have to find a tighter bound on (79), and find proposal distributions for α and β that suit the observed data better. The posteriori distributions of α and β involve gamma functions with α and β as arguments, and are hence rather intractable. Instead of searching for tighter bounds and better proposal distributions we decided to investigate a model with a reparametrization of the failure probabilities, the *logit* model.

8.3 THE *logit* MODEL

In this model the failure probabilities p_i are the *logit* transformations of a random variable b_i, with

$$p_i = \frac{\exp(b_i)}{1+\exp(b_i)} \tag{84}$$

and where b_i is normally distributed with expectation μ and precision τ. The distribution function is

$$\pi(b_i|\mu,\tau) = \frac{\sqrt{\tau}}{\sqrt{2\pi}} \exp\left(-\frac{\tau}{2}(b_i-\mu)^2\right) \tag{85}$$

The next step is to assume prior distributions for either μ, τ or both. This model can be seen graphically as the directed acyclic graph in Figure 13.

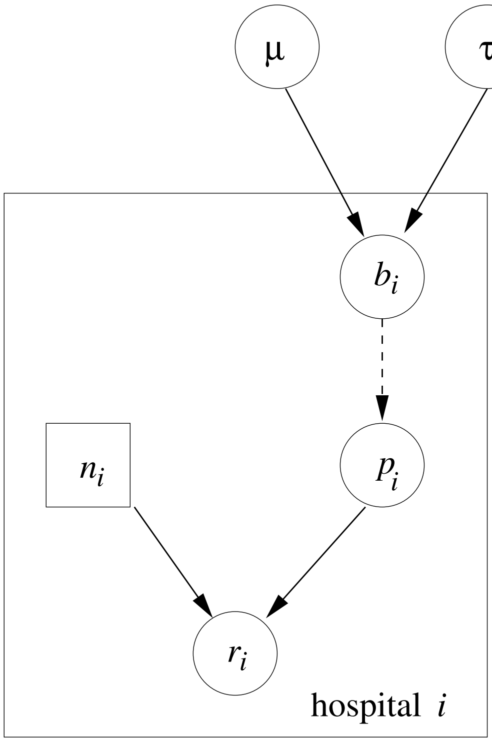


Figure 13: Directed acyclic graph for the *logit*-model

8.3.1 Unknown expectation, known precision

We first assume that the precision τ is known, and that the priori distribution of μ is non-informative normal with expectation 0 and precision γ. The posteriori multivariate distribution of b, μ given the observed failures is proportional to the following

$$\pi(b,\mu|n,r) \propto \frac{\sqrt{\gamma}}{\sqrt{2\pi}} \exp\left(-\frac{\gamma}{2}\mu^2\right) \prod_{i=1}^{k} \binom{n_i}{r_i} p_i^{r_i}(1-p_i)^{n_i-r_i} \frac{\sqrt{\tau}}{\sqrt{2\pi}} \exp\left(-\frac{\tau}{2}(b_i-\mu)^2\right) \tag{86}$$

where $b = (b_1,\ldots,b_k)$, $n = (n_1,\ldots,n_k)$, $r = (r_1,\ldots,r_k)$ and p_i is given in Equation (85).

Consider first sampling from this distribution using a cyclic Gibbs sampler. Since b only depend on μ and visa versa the space of current states will be finite as soon as the space for either μ or b is finite. We will now investigate if we can use the multigamma-coupler or the rejection sampler for the conditional distributions. The distribution of b conditional on μ and the failures is proportional to (86) but non-trivial, and hence we can not find the normalising constant. Therefore the multi-gamma coupler is excluded. Also, since $|\mu|$ can be arbitrarily large we can not find a lower bound on this density, and hence we can not use the rejection sampler. The normal family is its own conjugate, and this gives a normal posteriori distribution for μ given the b_i's. This distribution is given by

$$q(\mu|b) = \frac{\sqrt{\gamma + k\tau}}{\sqrt{2\pi}} \exp\left(-\frac{(\gamma + k\tau)}{2}\left(\mu - \frac{1}{1+\frac{\gamma}{k\tau}}\bar{b}\right)^2\right) \tag{87}$$

where $\bar{b}$ is the mean of $b = (b_1, \ldots, b_k)$. Unfortunately, since $\left|\bar{b}\right|$ can be arbitrarily large we can not find a lower bound on this density.

Consider next the Metropolis-Hastings coupler. The task is to construct an independence sampler q with a reasonable bound. The most reasonable approach is to sample either μ or b conditional on the observed failures, and then sample the other variable conditional on the first. Since the posteriori distribution of μ is normal it is natural to first sample b. We sample b by independently sampling the transformed values $p_i, i = 1, \ldots, k$ from the Beta$(r_i + 1, n_i - r_i + 1)$ distribution. This corresponds to sampling from the posteriori distribution of p when the p_i's a priori is independently and uniformly distributed. Since the b_i's are functions of the p_i's the distribution of b is the distribution of p times the *Jacobian*, i.e

$$\begin{aligned} q(b|n,r) &= \prod_{i=1}^{k} q(p_i|n_i, r_i)\left|\frac{dp_i}{db_i}\right| \\ &= \prod_{i=1}^{k} \frac{\Gamma(n_i+2)}{\Gamma(r_i+1)\Gamma(n_i-r_i+1)} p_i^{r_i}(1-p_i)^{n_i-r_i}\left|\frac{\exp(b_i)}{(1+\exp(b_i))^2}\right| \end{aligned} \tag{88}$$

such that the full distribution of the independence sampler is

$$\begin{aligned} q(b,\mu|n,r) &= q(\mu|b)q(b|n,r) \\ &= \frac{\sqrt{\gamma + k\tau}}{\sqrt{2\pi}} \exp\left(-\frac{(\gamma + k\tau)}{2}\left(\mu - \frac{1}{1+\frac{\gamma}{k\tau}}\bar{b}\right)^2\right) \\ &\quad \times \prod_{i=1}^{k} \frac{\Gamma(n_i+2)}{\Gamma(r_i+1)\Gamma(n_i-r_i+1)} p_i^{r_i}(1-p_i)^{n_i-r_i}\frac{\exp(b_i)}{(1+\exp(b_i))^2} \end{aligned} \tag{89}$$

Taking the ratio gives

$$\frac{\pi(b,\mu|n,r)}{q(b,\mu|n,r)} \propto \exp\left(-\frac{\tau}{2}\left(\sum_{i=1}^{k} b_i^2 - \frac{k}{1+\frac{\gamma}{k\tau}}\bar{b}^2\right)\right)\prod_{i=1}^{k}\frac{(1+\exp(b_i))^2}{\exp(b_i)} \tag{90}$$

which is unbounded. Hence we cannot use this methods without imposing a bound on the state space of μ or b. This is the same conclusion as Murdoch and Green reached in their paper ([9]) for their application, and it is a general problem when the state spaces are unbounded. Murdoch and Green then showed how to apply an approximative algorithm by imposing a bound on some of their parameters, and they made the algorithm more effective by partitioning the state space.

To find a bound on (90) we performed a 40000 iteration run with the simulation program BUGS [13], monitoring the value of this ratio at each step. A trace plot and histogram over the observed values (denoted " *ratio*") is shown in Figure 14. The maximum value observed where 2.8×10^{14}, and thus we

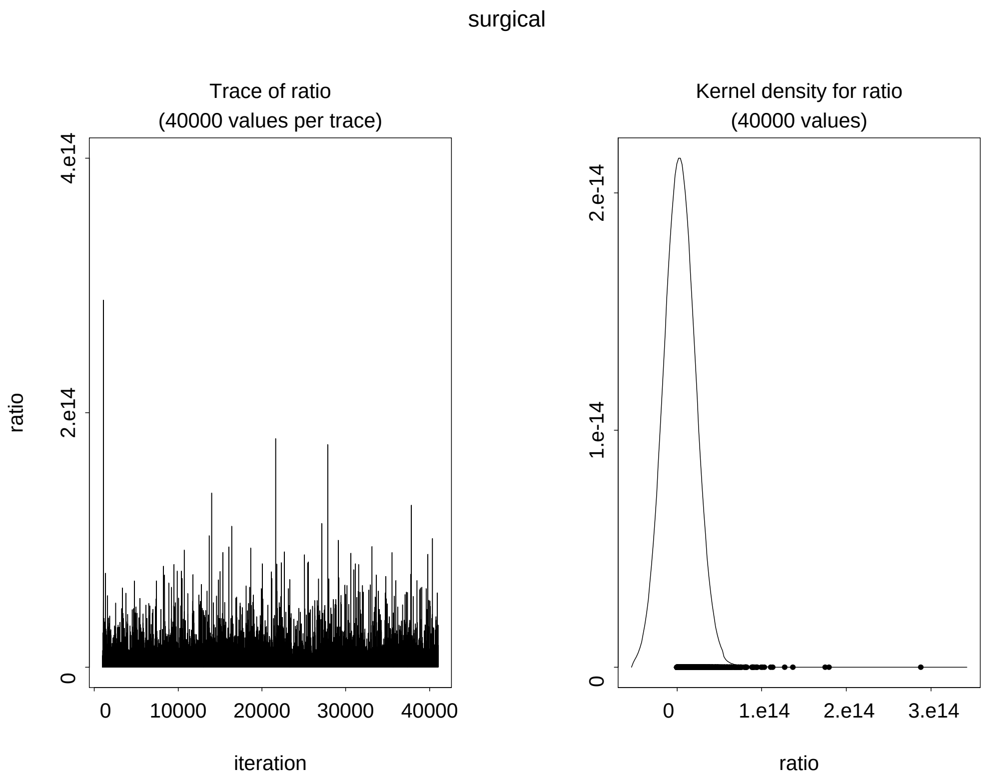


Figure 14: Trace plot and histogram over the ratio (90) observed in a 40000 iteration run in `BUGS`.

imposed the bound 3×10^{14} on the ratio. Statistics from 500 independent samples using the Metropolis-Hastings coupler are shown in Table 5. As a comparison we show the statistics from the a 1000000 iteration run using `BUGS` where we have saved the sample at each 1000'nd iteration. As seen from Table 5 the estimates are of the same size using both methods. The large number of simulations using `BUGS` therefore strongly supports that our implementation of the Metropolis-Hastings coupler is correct, and that the samples are distributed according to the desired posteriori distribution. Of course, the procedure of simulating the Markov chain to find a bound on the ratio (90) can also be applied to the ratio (79). It is an approximative procedure, and further work is necessary to develop truly perfect sampling algorithms useful in practice.

Table 5: STATISTICS. *This table shows the estimated values for the mean and standard deviation for the parameters in the logit-model. The values for the Metropolis-Hastings algorithm are computed from 500 independent samples, while we performed 1000000 iterations in* `BUGS` *and saved the sample at each 1000'nd iteration. The estimates are of the same size, and the large number of simulations using* `BUGS` *therefore strongly supports that the samples is distributed according to the desired posteriori distribution.*

	METROP.-HASTINGS		BUGS	
$variable$	$mean$	$st.deviation$	$mean$	$st.deviation$
μ	7.278E-2	1.032E-2	7.221E-2	1.067E-2
p_1	5.087E-2	1.891E-2	4.957E-2	1.810E-2
p_2	1.079E-1	2.367E-2	1.072E-1	2.158E-2
p_3	7.018E-2	1.822E-2	7.070E-2	1.810E-2
p_4	5.800E-2	7.345E-3	5.800E-2	7.611E-3
p_5	4.802E-2	1.220E-2	4.825E-2	1.208E-2
p_6	6.826E-2	1.529E-2	6.762E-2	1.566E-2
p_7	6.532E-2	1.668E-2	6.524E-2	1.634E-2
p_8	1.300E-1	2.187E-2	1.291E-1	2.074E-2
p_9	6.954E-2	1.483E-2	6.909E-2	1.526E-2
p_{10}	7.888E-2	2.221E-2	7.828E-2	2.068E-2
p_{11}	1.064E-1	1.817E-2	1.048E-1	1.689E-2
p_{12}	6.785E-2	1.192E-2	6.738E-2	1.143E-2

9 Other work within perfect sampling

In this section we give some references to other work done with exact sampling algorithms. A useful guide to the topic is the homepage of D.B. Wilson at

`http://dimacs.rutgers.edu/~dbwilson`

Together with the original paper [11] on coupling from the past Propp and Wilson have written a "user's guide" on perfect sampling and some papers on generating random spanning tree of a directed graph.

Foss and Tweedie [3] place the coupling from the past algorithm in the context of stochastically recursive sequences. The also show that successful CFTP can be constructed if and only if the chain is uniformly geometric ergodic.

Kendall [7] shows how to apply perfect sampling algorithms for simulation of point processes. A modification of the monotone CFTP is made, since the state space is infinite with no top state. Considers also repulsive point processes, which are not monotone.

Häggström, Lieshout and Møller [10] presents a faster algorithm than Kendall, applied to spatial point processes. Møller [8] also give a review of spatial point processes including perfect simulation.

Häggström and Nelander [4] shows how to use coupling from the past on anti-monotone systems.

Johnson [6] uses coupling in the study of convergence of Markov chains.

10 CONCLUSION

This report gives a review of the basic ideas in the new methods of perfect sampling using Markov chains.

The first part covers the discrete case, and we consider the algorithms of Propp-Wilson and Fill. Both these algorithms are tested on two applications, the Ising model and an image noise reduction system. We get satisfactory results, though the algorithms are rather time-consuming. For the "coupling from the past" technique of Propp and Wilson we performed a regression analysis of the coupling time T as function of the grid size n. The empirical results show that we have $T = \mathcal{O}(n^{\alpha})$, where we have the estimate $\alpha = 2.44$. According to Fill we have done the first implementation of Fills rejection coupler, and we have verified the correctness through a χ^2-test.

In the second part we proceed to the continuous case. We present several algorithms developed by Murdoch and Green. These algorithms are also based on the "coupling from the past" technique. We discuss how these methods may be applied to a Bayesian analysis problem of surgical failure rates. Samples from one of the methods, the Metropolis Hastings coupler, are compared with samples received from the simulation program `BUGS`. Even for this simple problem we had to impose some restrictions on the state space using the Metropolis Hastings coupler, and this suggest that the perfect sampling methods for continuous state spaces need further investigation.

REFERENCES

[1] Stephen P. Brooks. Markov Chain Monte Carlo and its Application. Available at http://www.stats.bris.ac.uk/.

[2] J. A. Fill. An Interruptible Algorithm for Perfect Sampling via Markov Chains, 1997.

[3] S. G. Foss and R. L. Tweedie. Perfect simulation and backward coupling. *to appear in Stochastic Models*, 1997.

[4] O. Häggström and K. Nelander. Exact sampling from anti-monotone systems. Available at http://dimacs.rutgers.edu/~dbwilson, 1997.

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

[6] V. E. Johnson. Studying convergence of markov chain monte carlo algorithms using coupled sample paths. *Journal of the American Statistical Association*, 91(433):154–166, 1996.

[7] W. S. Kendall. Perfect simulation for the area-interaction point process. In *Proceedings of the Symposium on Probability Towards the Year 2000*. World Scientific Press, 1996. To appear in C.C. Heyde and L. Accardi, editors, Probability Perspective,.

[8] J. Møller. Markov chain monte carlo and spatial point processes. In W. S. Kendall O. Barndorff-Nielsen and M. N. M. van Lieshout, editors, *Proceedings Seminaire Europen de Statistiqe, "Stochastic geometry, likelihood, and computation*. Chapman and Hall.

[9] D. J. Murdoch and P. J. Green. Exact Sampling from a Continuous State Space. Available at http://www.stats.bris.ac.uk/MCMC.

[10] M. N. M. van Lieshout O. Häggström and J. Møller. Characterisation results and Markov chain Monte Carlo algorithms including exact simulation for some spatial point processes. Technical Report R-96-2040, Aalborg University, 1996.

[11] J. G. Propp and D. B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9:223–252, 1996.

[12] B. D. Ripley. *Stochastic Simulation*. John Wiley & Sons.

[13] D. Spiegelhalter, A. Thomas, N. Best, and W. Gilks. *BUGS 0.5*Bayesian inference Using Gibbs Sampling Manual*, 1996.

[14] D. Spiegelhalter, A. Thomas, N. Best, and W. Gilks. *BUGS 0.5*Examples, Volume 1*, 1996.

[15] H. M. Taylor and K. Samuel. *An Introduction To Stochastic Modeling*. Academic Press, Inc., 1984.

[16] G. Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*. Springer, Berlin, 1995.

A THE χ^2 GOODNESS OF FIT TEST

We give a short introduction to the χ^2 test for goodness of fit. For more on this test, see any introductory text in statistics. The main purpose of this testing is to investigate whether or not observed data fit a hypothesised distribution. Consider a sample of n observations coming from a finite and discrete distribution, each configuration having probability p_i, $i = 1, \ldots, r$. This corresponds to the multinomial situation. We have the null hypothesis

$$H_0 : p_1 = p_{10}, \ldots, p_r = p_{r0} \tag{91}$$

where the p_{i0} are the expected probabilities. N_i counts the number of observed configurations for all i, and the expected number e_i of observations equals $e_i = E[N_i] = np_i$. The random variable

$$W = \sum_{i=1}^{r} \frac{(N_i - e_i)^2}{e_i} \tag{92}$$

approaches a chi-square distribution with $r-1$ degrees of freedom as n tends to infinity. It is clear that a great value of the observed test statistic w suggests rejection of the null hypothesis, and we reject H_0 if $w \geq \chi^2_{\alpha,r-1}$, α being our confidence level. In our case we have $r = 512$, and this suggests the approximation

$$\frac{W - E[W]}{\sqrt{Var[W]}} \sim N(0,1) \tag{93}$$

where the expectation of a χ^2_ν random variable equals ν and the variance is 2ν. The p-values are then easily found. Since this is a approximative test, a rule of thumb is to perform the test only when $e_i > 5$ for all i.