Componentwise adaptation for MCMC

Heikki Haario*, Eero Saksman† and Johanna Tamminen‡

January 10, 2003

* *University of Helsinki*
Department of Mathematics
P.O.Box 4, FIN-00014 University of Helsinki, Finland

† *University of Jyväskylä*
Department of Mathematics and Statistics
P.O.Box 35 (MaD), FIN-40014 University of Jyväskylä, Finland

‡ *Finnish Meteorological Institute*
Geophysical Research Division
P.O.Box 503, FIN-00101 Helsinki, Finland [1]

Abstract

Markov chain Monte Carlo methods are widely used for computing posterior distributions of inverse or parameter estimation problems. The simplicity of the original Metropolis–Hastings algorithm has lead to wide exploitation of the algorithm. However, in practice, simple Metropolis–Hastings algorithms require tuning of some internal variables before the algorithms work properly. Recently, many adaptive algorithms have been developed to automatically obtain efficient sampling. Unfortunately, when the dimension becomes fairly large the adaptive methods developed so far often become unpractically inefficient.

Here we introduce a new adaptive MCMC algorithm, based on the basic single component Metropolis–Hastings algorithm. The method is a variant of our earlier adaptive Metropolis algorithm. In the single component version the adaptation is performed component by component. The chain is no more Markovian, but it remains ergodic. The algorithm is demonstrated to work well in varying test cases up to 1000 dimensions. As a realistic problem we have applied the single component adaptive Metropolis algorithm to a 90 dimensional satellite remote sensing problem.

Keywords: Markov chain Monte Carlo, adaptive Markov chain Monte Carlo, convergence, efficiency.
2000 Mathematical Subject Classification: 65C05, 65C40

[1] Send correspondence to J.T.,: E-mail: johanna.tamminen@fmi.fi

1 Introduction

Both measurement techniques and data processing capacity have developed strongly during last years. This entails that more data is available and the number of unknown parameters scientists wish to retrieve from data has exploded. This behavior is especially clear in remote sensing techniques, biotechnology and different tomography measurements. In addition,when large amount of data and high dimensional problems are solved in a fully Bayesian way (which requires computing the high dimensional posterior distribution of the unknown quantities), it is obvious that we still face problems with computing power. Different types of strategies for solving high dimensional problems have lately been under active research. Recent work include, for example, dimension reduction by clustering objects in image processing [14] or the use of an intermediate coarser grid during sampling in tomography type problems [11].

The most important tool for Bayesian data analysis is provided by various Markov chain Monte Carlo (MCMC) algorithms. The power of MCMC methods lies principally in the simplicity and wide applicability of the basic algorithm. However, in practice, the originally simple algorithm, like Metropolis-Hastings algorithm [15, 10] does not always work well without laborious tuning of the algorithm. This problem has recently been studied intensively and many adaptive MCMC methods have been presented. However, the high dimensional problems have been barely touched in these works. In this paper we describe a new adaptive MCMC algorithm that could be used in moderately high dimensional problems.

1.1 Adaptive algorithms

In general, using information from previously sampled points to improve sampling may disturb the Markov property. Indeed, this is why the natural idea of using adaptive approach only during the burn-in period has been used in many practical applications (see e.g. [7] and [9]). In addition to these quasi-adaptive methods there exist numerous fully adaptive algorithms where the adaptation is continued also after the burn-in period. The Adaptive Metropolis algorithm [8] tunes the Gaussian proposal distribution using the whole history of the chain. In Metropolis-Hastings with learning proposals [4] and in Normal Kernel Coupler [20] the authors use multiple chains to learn from the target distribution. A framework for adaptation by using regeneration of the chain is introduced in [6], and the Self regenerative algorithm with adaptation [16] resembles independence sampling with proposal distribution that adapts at so called trouble points. Another interesting idea is to learn from the rejected states instead of the previously sampled points. This idea is developed in [19] under the name delayed rejection. It has been showed to increase mixing of the chain in many cases. Obviously, our list of adaptive algorithms is far from complete.

Most of the adaptive methods have been demonstrated to work well in practice in rather low dimensional problems. It is rather surprising to see that test results for most of the purely adaptive algorithms are presented only in cases with 1-5 unknown parameters, with the exceptions that Normal Kernel Coupler [20] has been demonstrated to work in dimension 20 and Adaptive Metropolis algorithms in dimension 8. However, in many practical problems the dimension of the problem is relatively large. In such cases the behavior of adaptive algorithms may suffer considerably. For example this is the case for AM algorithm: in our tests and in many real applications it works robustly when, roughly speaking, the dimension is less tan 50 — see Sec. 2 below — but the performance becomes less robust in higher dimensions.

1.2 Single component approach

In high dimensional problems, like in tomography, it has traditionally been common to use MCMC methods which proceed componentwise: Gibbs sampling or single component Metropolis–Hastings methods (denoted by SC from now on), see e.g. [12]. One of the reasons being that the Metropolis-Hastings algorithm with multidimensional proposal distribution seems to be more sensitive to the increasing dimension (see Appendix A below).

The disadvantage of single component approach is that we need to evaluate the target function for each parameter separately. It depends on the problem itself, whether this can be done effectively by only updating the contribution of one parameter on the target function or if it is required that the whole target function is recomputed. So far there has been only very little work on adaptive MCMC algorithms which would apply on conditional distributions as noted in [19]. This is one of the reasons why Gibbs sampling has been commonly used in high dimensional problems also in cases where the full conditional distributions are not known and they have to be approximated at each step and for each parameter.

1.3 Purpose of the paper

The purpose of this paper is to introduce a simple and flexible adaptive MCMC algorithm, which, we hope, can be successfully used in many high dimensional problems. The applications that we have in mind are mainly inverse problems rising in physics and chemistry, where number of unknown parameters is from ten to some thousands and the target distribution is unknown but typically, at least roughly, unimodal. The single component adaptive Metropolis (SCAM) algorithm, presented here, can be considered as a single component version of adaptive Metropolis (AM) algorithm, introduced in [8]. The new algorithm is described in Sec. 2. The proof of the ergodicity of SCAM algorithm is found in Appendix B. The performance of the SCAM algorithm is tested in numerous test cases and the results are shown in Sec. 3. As a realistic test case, we have applied in Sec. 4 the SCAM algorithm in a real atmospheric tomography problem, where the number of unknown parameters is 90.

2 Single component adaptive Metropolis algorithm

2.1 Adaptive Metropolis algorithm

The single component adaptive Metropolis (SCAM) algorithm introduced below is a variant of the AM algorithm introduced in [8] which we therefore shortly recall here. The AM algorithm is basically typical Metropolis algorithm with a multi-normal proposal distribution but the proposal distribution is updated along the time using the chain created so far. The acceptance probability of the AM algorithm corresponds formally to the acceptance probability of a symmetric Metropolis algorithm. The algorithm is not Markovian, but the ergodicity can be proven to hold. We refer to [8] for further details. The adaptation ideas and the ergodicity proof presented in [8] have recently been extended in [2] to cover also other types of adaptation. However in [2] the authors concentrate on describing a framework for adaptation and practical solutions for adaptation are not considered.

The AM algorithm has been successfully applied to a variety of real rather low dimensional problems and, indeed, in some sense, we may say that it solved the problem of describing a fully adaptive MCMC algorithm which would be easy to use. However, our tests seem to indicate that in relatively high dimensional problems (as the dimension approaches 100, say) the convergence of the algorithm starts to be slow (at least numerically), and possibly (this is not clear yet) the adaptation procedure may have a negative effect to the speed of the algorithm. This is demonstrated in Fig. 1 where we show the performance of AM algorithm for increasing dimensions. The figure presents the proportion of the samples in the 50 % probability region as a function of dimension. For each dimension the results have been calculated as an average of repeated runs for randomly generated (correlated) linear problems. The chain length was fixed to 40000 in all the cases. The figure suggests that the basic form of AM algorithm (as such) may not work properly in high dimensional problems. This is probably true for also many other adaptive algorithms.

2.2 Adaptive single component Metropolis algorithm

The SCAM algorithm can be considered as a typical SC algorithm where the 1-dimensional proposal distributions are tuned during the sampling in a similar manner as in the AM algorithm: in a sense, we apply the AM on each parameter separately. The proposal distribution of each component is a normal distribution with present point as the center point and with variance that depends on time. During the sampling the variance is updated at pre-defined time intervals and the updating is based on the variance of the so far sampled values for the corresponding parameter.

We now turn to the detailed description of the SCAM algorithm. Let π denote the density of our target distribution, typically posterior density distribution which we can evaluate up to a normalizing constant. We follow closely the notation of [8]. The sequence $X_0, X_1, \ldots$ denotes the full states of the process, that is, we consider a new state updated as soon as all the d coordinates (or components) of the state have been

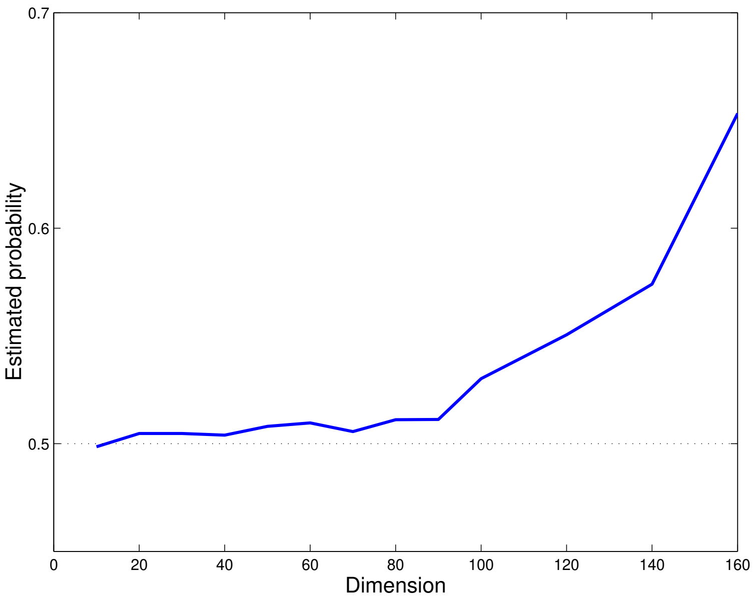

Figure 1: Performance of the AM algorithm as a function of dimension: estimated probability, the correct value 0.5 is denoted with dotted line.

separately updated. When deciding the i:th coordinate X_t^i $(i = 1, \dots, d)$ of the t:th state X_t we apply the 1-dimensional Metropolis step:

1. Sample z^i from 1-dimensional normally distributed proposal distribution $q_t^i \sim N(X_{t-1}^i, C_t^i)$ centered at previous point with covariance (variance) C_t^i. Here

$$C_t^i = \begin{cases} C_0^i, & t \le t_0 \\ s\mathrm{Cov}(X_0^i, \dots, X_{t-1}^i) + s\varepsilon, & t > t_0. \end{cases}$$

2. Accept the candidate point z^i with probability

$$\alpha(X_{t-1}, z) = \min\left(1, \frac{\pi(X_t^1, \dots, X_t^{i-1}, z^i, X_{t-1}^{i+1}, \dots, X_{t-1}^d)}{\pi(X_t^1, \dots, X_t^{i-1}, X_{t-1}^i, \dots, X_{t-1}^d)}\right),$$

   in which case we set $X_t^i = z^i$, and otherwise $X_t^i = X_{t-1}^i$.

Observe that the covariance $C_t^i > 0$ is a real number. Above $C_0^i > 0$ denotes an initial (arbitrary) variance of the proposal distribution for parameter i. Moreover, s denotes the scaling factor for which we have used the value $s = 2.4$ given by [5]. Finally ε is a small constant, which just prevents the covariance of the proposal distribution from shrinking to a singular matrix. It is worth pointing out that the actual values of C_0^i and ε are not important for the performance of the SCAM algorithm, which underlines how easy SCAM is to implement in practice. Parameter t_0 represents some kind of initial burn in period for which we have used the value $t =$, for instance.

Note that the acceptance probability coincides by definition the acceptance probability of basic single component Metropolis algorithm with symmetric proposal distribution. However, since the chain is not Markovian, it is not a priori clear that

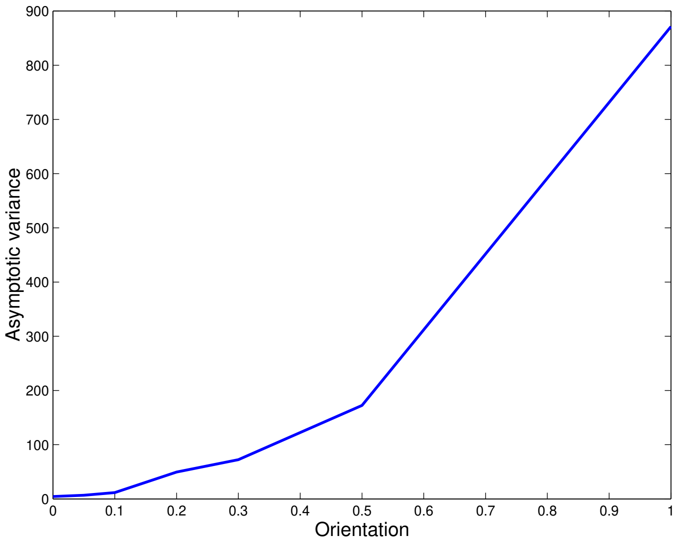

Figure 2: Mean asymptotic variance of SC as a function of chain length with varying target distribution correlation (2-d case).

the desired convergence of the algorithm takes place. The proof of this fact relies on results in [8], and is found in the Appendix B below.

2.3 Rotating the proposal distribution

When the target function is correlated the single component Metropolis algorithms (as well as Gibbs sampling) are known to suffer from poor mixing, see Fig. 2. This is well known and as a typical remedy for this it is common to either rotate the proposal distribution or rotate the target distribution so that the sampling directions of the SC-algorithm corresponds to the directions of the principal vectors. This effect is illustrated in figure 3 by showing two different ways of jumping from X_1 to X_2; either by using the directions of the axis to jump first to z and then to X_2 or first to z' and then to X_2. It is obvious that the latter one is more effective since the probability of jumping to z is very low. Naturally, if not taken into account, also the SCAM algorithm suffers from this effect. We will therefore also suggest to rotate the proposal distribution during the early stage of the sampling (i.e. during the burn-in period).

The rotation of the proposal distribution within SCAM algorithm can be done in a following way: at predefined steps we compute the covariance matrix of the chain so far detected. Then we compute the principal vectors of the covariance matrix and use these directions as sampling directions in the SCAM-algorithm. After the burn-in period the direction of the proposal distribution is fixed and the sampling is done as described above only by updating the size of the one-dimensional Gaussian proposal distribution.

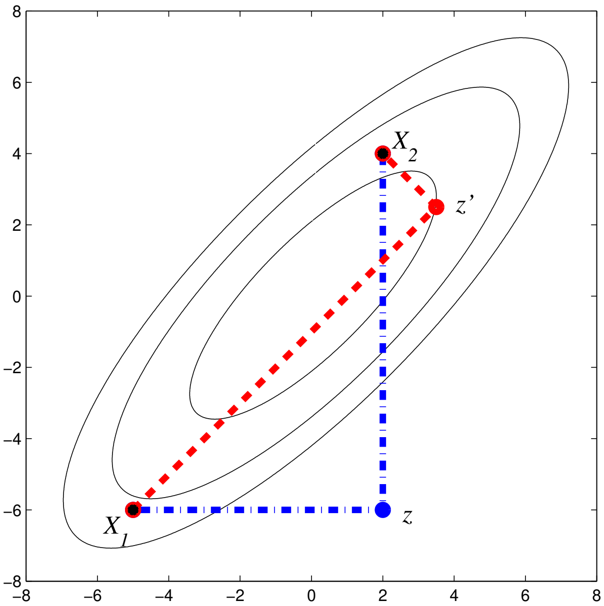

Figure 3: Sampling directions of normal SC algorithm along coordinate axis (dash-dot line) and according to more optimal direction (dashed line). The target function is represented by the contour curves.

3 Tests

We have tested, with toy examples, the performance of SCAM algorithm in three different target distribution cases:

- π_a: slightly correlated 150-dimensional Gaussian target. The target was created by rotating slightly the corresponding non-correlated Gaussian target distribution $N(0, C_a)$ where the covariance $C_a = \mathrm{diag}(\lambda_1^2, \ldots, \lambda_{150}^2)$. We have here used values $\lambda_1 = 10$ and $\lambda_i = 1$, when $1 < i \leq 150$, which define a target distribution where the largest principle axis is 10 times larger than the others.

- π_b: slightly twisted but almost non-correlated Gaussian 150-dimensional target distribution with one axis 10 times longer than the others. We refer to [7] for the definition of the slightly twisted Gaussian target (actually our target here is equal to target π_3 in [7] with $b = 0.01$).

- π_c: 1000-dimensional Gaussian target with one axis longer and one shorter than the others: $N(0, C_c)$ with $C_c = \mathrm{diag}[120, 6 \ldots, 6, 0.3])$

All our test targets are similar to those in [7] apart from a possible rotation in coordinates). The slight rotation in the target distributions has been used to imitate the fact that the orientation of the target distribution is usually not known. The expectation value of all three target distributions has been the origin. In the statistical testing procedure we used the same approach that was used in [7].

In the tables below we list our test results: the mean and standard deviation of the sample mean values, the error and standard deviation from the true 68.3 % probability region, the corresponding 1-dimensional values projected on largest eigenvalue direction and the largest integrated autocorrelation value (τ). In all these benchmark values smaller numbers indicate better performance. In addition we show the mean acceptance ratio of all parameters.

The algorithms compared have been SCAM, 'non-optimal' and 'optimal' SC algorithms and Metropolis algorithm with multi-normal proposal. In the SC algorithms the proposal distributions were Gaussian, centered at current point. In the 'optimal' case the size of the 1-dimensional proposal distributions were equal to the corresponding optimally scaled Gaussian 1-dimensional projections of the target distribution where the small correlations in parameters were not taken into account. In 'non-optimal' cases the sizes of the proposal distributions were selected so that they were slightly, but not dramatically, shifted from the optimal values.

The chain length was 5000 in all single component methods and the burn-in period of 1500 points was not included in the analysis. In the M-H algorithm with multi-normal proposal the chain length was 250000, with burn-in 75000 and only every 50th point was included in the analysis (i.e. we used the same number of points in the analysis as in the SC methods).

We see that the performance of the SCAM algorithm is clearly superior in all the test cases compared to 'non-optimal' SC algorithms and, indeed, it corresponds to the 'optimal' SC algorithm in the case of slightly correlated Gaussian target. The better performance can also be easily seen in the integrated autocorrelation values which indicate also that the SCAM algorithm is more efficient than the 'non-optimal' SC algorithms. We also want to point out that the adaptation in the SCAM algorithm was continued throughout the sampling so that at every 100th step the sizes of the 1-dimensional proposal distributions were scaled.

Table 1: Target π_a: slightly correlated 150 dimensional multi-normal distribution with one axis 10 times longer than the others. SC-a and SC-b refer to two SC runs with different (non-optimal) Gaussian proposal distributions centered at current point with scaling σ (i.e. , proposal for ith coordinate is $z^i \sim N(X_t^i, \sigma^2)$). The values of sigma are given in the table. M–H was also run with 'non-optimal' spherical Gaussian proposal ($z \sim N(X_t, \sigma^2 I_{150})$) For comparison we show in the last column results with 'optimal' SC algorithm (i.e. run with optimal 1-dimensional proposals). The results are averages over 100 repetitions.

Method	**SCAM**	**SC-a**	**SC-b**	**M–H**	**SC-opt**
Chain length	**5 000**	**5 000**	**5 000**	**250 000/5000**	**5 000**
Proposal	**adaptive**	$\sigma = 2$	$\sigma = 0.5$	$\sigma = 0.45$	$\sigma_i = 2.4\lambda_i$
mean($\|\|\mathbb{E}\|\|$)	0.80	1.25	3.7	4.29	0.79
std($\|\|\mathbb{E}\|\|$)	0.23	0.60	2.48	2.98	0.22
err($\leq$ **68.3%**)	0.77	0.34	0.5	0.02	0.76
std($\leq$ **68.3%**)	1.67	1.67	1.95	2.69	1.5
1-d err(68.3-95%)	0.15	0.2	1.08	0.92	0.2
1-d std(68.3-95%)	1.68	2.19	10.34	11.77	1.7
largest τ	11	47	196	216	12
mean acc. rate (%)	39	30	73	34	39

Table 2: Target π_b: moderately twisted distribution with one axis longer than the others. SC was run with 'non-optimal' Gaussian proposal distributions (with deviation 1.2). Results are averages over 50 repetitions.

Method	**SCAM**	**SC**
Chain length	**5 000**	**5 000**
Proposal	**adaptive**	$\sigma = 1.2$
mean($\|\|\mathbb{E}\|\|$)	0.73	5.13
std($\|\|\mathbb{E}\|\|$)	0.38	3.62
err($\leq$ **68.3%**)	0.97	0.44
std($\leq$ **68.3%**)	1.65	2.81
1-d err(68.3-95%)	0.55	0.41
1-d std(68.3-95%)	2.45	12.8
largest τ	31.77	207.08
mean acc. rate (%)	44.14	84.47

Table 3: Target π_c 1000-dimensional multi-normal distribution with one axis longer and one shorter than the others. For comparison the SC algorithm was run with 'non-optimal' Gaussian proposal (with deviation 0.72). Note that results are averages over only 10 repetitions.

Method	**SCAM**	**SC**
Chain length	**5 000**	**5 000**
Proposal	**adaptive**	$\sigma = 0.72$
mean($\|\mathbb{E}\|$)	1.12	3.87
std($\|\mathbb{E}\|$)	0.03	1.06
err($\leq$ **68.3%**)	0.28	0.48
std($\leq$ **68.3%**)	0.58	0.85
1-d err(68.3-95%)	0.11	4.3
1-d std(68.3-95%)	0.71	4.12
largest τ	5	755
mean acc. rate (%)	44	87

4 GOMOS measurements of the atmosphere: 90 dimensional case.

We will here apply the SCAM algorithm to a realistic problem. On-board European Space Agency's Envisat satellite is GOMOS instrument, which detects how stellar light is attenuated in the atmosphere to measure the atmospheric gas densities, and especially ozone density profiles in the stratosphere and mesosphere. The data retrieval problem of the UV-visible spectrometer of GOMOS contains typically approximately 3-5 unknown gas densities at around 30-100 different altitudes, altogether 90-500 unknown parameters. The transmission data itself consists of separately measured transmission spectra at 1417 wavelengths at $M = 30 - 100$ altitudes. It can be modelled with the Beer's law:

$$T_{\lambda,\ell}(\rho(z)) = \exp - \int_{\ell} \sum_{j} \rho_j(z(s)) a_j(\mathrm{t}(z(s)))\ ds,$$

where $\lambda = \lambda = 1, \ldots, 1417$ denotes wavelengths, $\ell = \ell_1, \ldots, \ell_M$ ray paths, and the unknown gas densities are $\rho = \rho_1, \ldots, \rho_J$ See also Fig. 4 for the measurement geometry. The measurement noise can be approximated with Gaussian noise with known variance. Traditionally the inversion problem with non-informative prior has been solved by dividing it to subproblems [13] where instead of inverting directly local gas densities horizontally integrated gas densities, $\int_{\ell} \rho(z(s))\ ds$, have been retrieved at each ray path separately. This first part is called spectral inversion. The local gas density values at each altitude are then solved in vertical inversion step by discretising the atmosphere. The MCMC techniques have earlier been successfully applied to

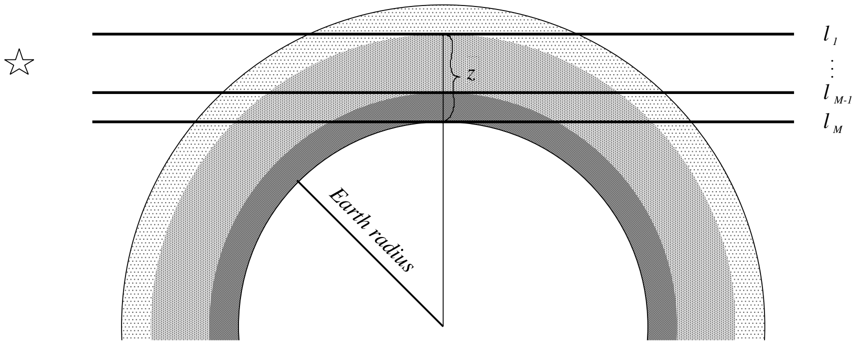

Figure 4: Measurement geometry of GOMOS.

these spectral inversion subproblems. In these subproblems the number of unknown parameters were around 5, see in [7] and [18].

The division of GOMOS inversion into subproblems is possible only if we assume that the cross sections of different gases, $a_j(\ell, \mathrm{t}(s(z)))$, which depend on atmospheric temperature and therefore also on altitude (z) can be approximated with so called effective cross sections. Naturally this approximation is an additional source of error in the inversion. If the problem is solved in the original form such approximations need not to be made. An additional disadvantage of dividing the problem into subproblems is that the a priori information of atmospheric gases is available in the form on constituent profiles. If the prior information is used in the subproblems we need to transform local gas density values to horizontally integrated ones by following the refracted ray paths in the atmosphere. This ray tracing computation creates additional uncertainty to the prior, and due to the varying measurement geometries these time consuming computations need to be repeated for each geographical location separately. Therefore, it would be much simpler to use directly local density priors which can be done if the problem is solved directly.

We will here show that the SCAM algorithm can be easily applied to this original, high dimensional, parameter estimation problem. As the data we used transmission spectra measured at 30 altitudes and we assumed 3 unknown gases: ozone, neutral density and aerosols. The dimension of the problem was 90.

The SCAM algorithm was first run in it's original form. As the starting point we used the point which was obtained by solving the spectral and vertical inversion in a traditional way. This starting point may be therefore considered to be rather good. The SCAM algorithm was run for 5000 samples. The integrated autocorrelation values varied between 8 and 278 with mean at 121 indicating that there were some rather strong dependence between sampled points, see upper panel in Fig. 5. By studying the correlations of the estimated parameters we found good agreement between the worst integrated autocorrelation values and the most correlated parameters. To improve the mixing we rotated the sampling directions to correspond the principal vectors of the parameters. The principal vectors (V) were computed by using the singular value decomposition of the covariance matrix of the so far sampled points (actually we used 2000 points from 1001 to 3000). Then the SCAM algorithm was run

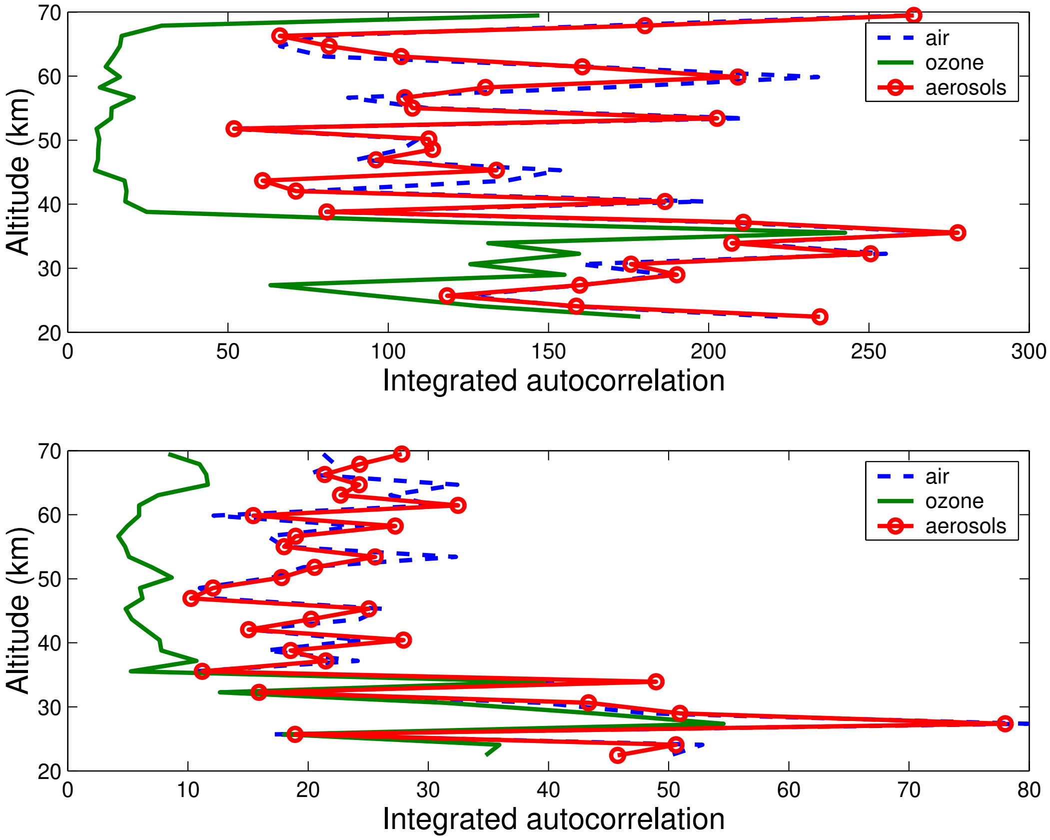

Figure 5: Integrated autocorrelation values. In upper panel: SCAM algorithm by using directions of parameters directly. Lower panel: SCAM algorithm with sampling directions rotated once during the burn-in period

for 5000 more points with the directions rotated by matrix V. After the rotation the integrated autocorrelation values varied between 4 and 80 with mean at 23, see lower panel in Fig. 5. In Fig. 6 we show the 2-dimensional marginal posterior distributions of different gases at some selected altitudes. We want to point out here that the SCAM algorithm has been run in a fully automatic way. The proposal distribution was scaled after each 25 sampled points and we continued the adaptation throughout the sampling.

5 Variants of SCAM

5.1 Adaptation based on acceptance ratio

To obtain better mixing of a MCMC sampler it has been common to monitor the acceptance rate of the sampler and tune the proposal distribution by hand to achieve more optimal acceptance rate. This idea can also be formulated as a variant of the AM or the SCAM algorithms. For simplicity, we illustrate the idea in 1-dimensional case (to treat higher dimension one e.g. just applies the same procedure for each

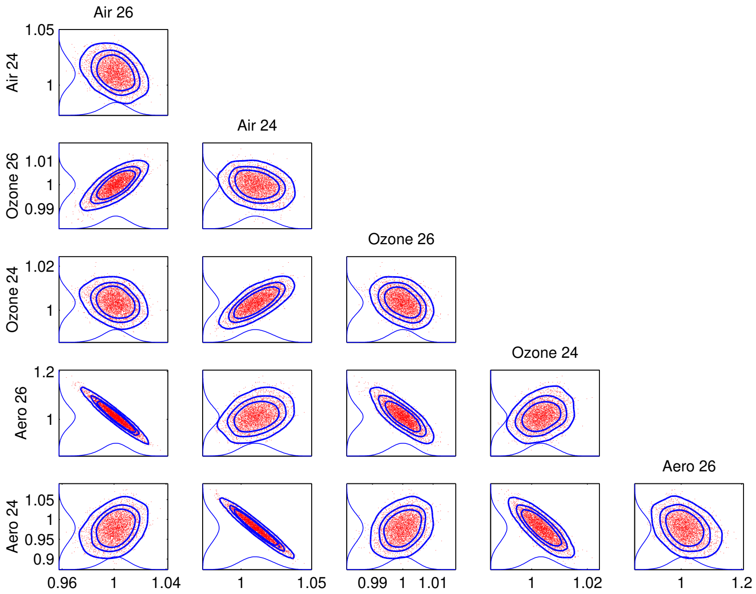

Figure 6: One and two dimensional marginal posterior distributions of lowest gas density values at 26 km and 24 km.

of the parameters separately). Let us denote at time t the 1-dimensional normal proposal distribution by $Q \sim N(\cdot \mid s_t^2)$. Here

$$s_t = k s_{t-1}, \tag{1}$$

where the update scaling parameter $k = k(p_t)$ depends on the acceptance ratio at time t:

$$p_t = \frac{\text{accepted points}}{\text{t}}. \tag{2}$$

One possibility is to define

$$k(p_t) = a^{p_t - p_0} \tag{3}$$

where p_0 is the assumed optimal acceptance ratio (according to [5] in one dimensional Gaussian case $p_0 \approx 0.44$). Observe that if $\lim_{t\to\infty} p_t = p_0$, then $k \longrightarrow 1$. Heuristically, the mechanism should ensure, that in the limit the acceptance rate approaches the value p_0, which is thought to be the ideal one. Theoretically, it is easy to show that, assuming lower and upper bounds for the covariances s_t, the proof presented in the Appendix B for SCAM algorithm applies also here and hence the convergence takes place.

This variant of SCAM algorithm seems to be simpler than basic SCAM by definition but it requires that the internal parameters p_0 and a are defined beforehand, whereas basic SCAM algorithm can be applied as it is. Moreover, it is known that an optimal value p_0 depends strongly on the target distribution.

5.2 Combining AM and SCAM

One may also interpret Fig. 2 in such a way that it might be more efficient to sample a relatively small number of correlated parameters as a group instead of using single component approach as long as the dimension of the space is relatively low. This observation indicates that it is worth considering whether the correlated parameters of the target distribution could be divided into groups which could be treated separately as components in a single component algorithms. In this case we would sample correlated parameters in one step and other (practically non-correlated) parameters separately. A natural further idea is to add adaptation to the picture. This could be done in the following way: in the case where the size of each mutually correlated parameter group is small we use AM approach for adapting the optimal multidimensional proposal distribution for each such group separately, and we use the SCAM algorithm for the remaining parameters.

6 Discussion

Many adaptive algorithms have been recently developed to make the MCMC sampling more efficient and easier to use, even automatic. However, these adaptive methods have not been demonstrated to work in high dimensional problems. In high dimensional cases it is typical to solve the problem using dimension reduction idea by conditioning: MCMC sampling is performed component by component. Gibbs sampling and single component Metropolis–Hastings algorithms are the most commonly used approaches. We have here presented a simple adaptive MCMC algorithm SCAM, which can be easily applied to different types of problems. The SCAM algorithm can be considered as a single component version of the AM algorithm, introduced in [8]. We have here showed in numerous test cases with varying dimensions from 150 to 1000 the performance of SCAM algorithm. The results show that in our test cases the performance of SCAM algorithm is superior to non-optimal basic single component Metropolis–Hastings algorithm.

7 Appendix A: Multidimensional vs. uni-dimensional proposal

The Metropolis–Hastings methods (M–H) with multidimensional proposal distribution can be used effectively as long as the target distribution is rather low dimensional and the proposal distribution is a good one. However, when the dimension of the problem rises longer chains are required to achieve the same accuracy as in lower dimensional problems. This is demonstrated in Fig. 7 where the mean of the integrated autocorrelation times as a function of dimension is plotted. Integrated autocorrelation time:

$$\tau = \sum_{t=-\infty}^{\infty} \frac{C(t)}{C_0},$$

where $C(t) = \mathrm{Cov}[X_0, X_t]$ is the autocorrelation function and C_0 is the variance of the expectation value under target distribution π, can be considered as a measure of efficiency, see [17]. We have here followed the notation of [3] and hence the integrated autocorrelation time differs from Sokal's definition by a factor of 2. For computing the integrated autocorrelation times we have used the adaptive truncated periodogram estimate defined in [17].

As an initial example, the target distribution was multi-normal without correlation and with one axis 10 times longer than the others. In the figure we show two methods with constant chain lengths: Metropolis algorithm with optimally scaled multi-normal proposal distribution (see [5] for optimal scaling) and single component Metropolis algorithm with optimal 1-dimensional normal proposal distributions. The results indicate clearly that in the case of non-correlated target the behavior of M-H methods gets worse as dimension grows, whereas the single component version of the algorithm is equally effective in all dimensions. Note that in this example we used optimal proposal distributions in both algorithms.

8 Appendix B: Proof of the convergence

We will here follow closely the notation of [8]. Let us first assume that S is a measurable and bounded subset of $\mathbb{R}^d$. Recall that the sequence $X_0, X_1, \ldots$ denotes the full states of the process, that is, we consider a new state updated as soon as all the d coordinates (or components) of the state have been separately updated. By the very definition of the algorithm there are the bounds

$$0 < c_2 \le C_t^i \le c_3 \tag{4}$$

for all i and t. The definition of the SCAM algorithm given in section 2.2 yields in a natural way a sequence of generalized transition probability kernels $(K_n)_{n=1}^{\infty}$ where,

$$K_n(x_0, \ldots, x_{n-1}; A)$$

gives us the probability of $X_n \in A \subset S$ under the condition that $X_i = x_i$ for $i = 0, \ldots, n-1$.

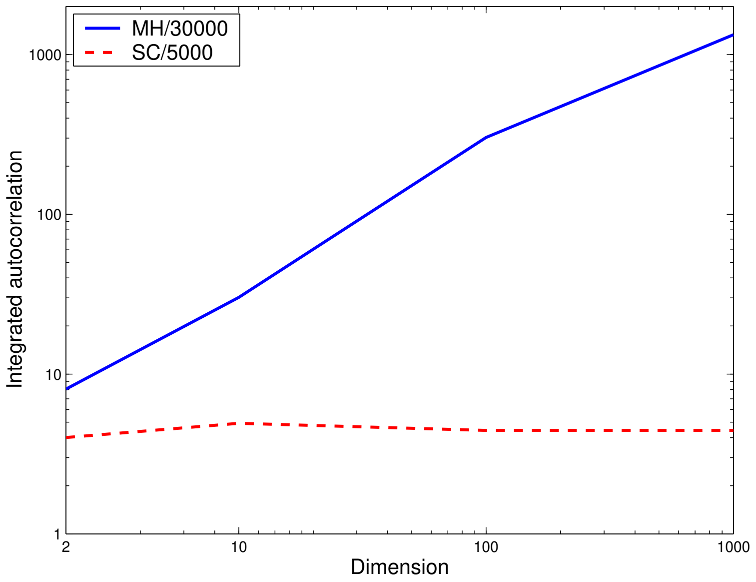

Figure 7: Mean of the integrated autocorrelation times as a function of dimension. Solid line: MH algorithm with optimal multi-normal proposal (30000 points) and dashed line: SC algorithm with optimal 1-dimensional normal proposals (5000 points).

In order to ensure that the single step algorithm is ergodic it is well known that one has to assume something from the geometry of the target distribution. A natural way to do this is the following: for $a > 0$ let $\widehat{M}_a$ be the transition probability kernel on S that corresponds to the single component Metropolis algorithm with target π and such that in each step the density of (one-dimensional) proposal is a uniform distribution centered at the current point, and of length 2a, where $a > 0$. We then assume that

(*) the Markov chain with the transition probability $\widehat{M}_a$ is uniformly ergodic if $a = \mathrm{diam}(\pi)$,

where $\mathrm{diam}(\pi)$ is the diameter of the support of the density π. This is fairly natural assumption, and is usually met in practice.

Theorem 1 *Let π be the density of the target distribution. Assume that it's support lies in a bounded measurable subset $S \subset \mathbb{R}^{\mathrm{d}}$ and that it is bounded from above: $\pi(x) \leq B < \infty$ for all $x \in \mathbb{R}^{\mathrm{n}}$. Assume also, that the geometry of the density π meets the condition (*). Then the chain corresponding to the SCAM algorithm is ergodic in the sense that the law of large numbers holds: for any bounded and measurable function $f : S \longrightarrow \mathbb{R}$ one has*

$$\lim_{n\to\infty} \frac{1}{n+1}(f(X_0) + f(X_1) + \ldots + f(X_n)) = \int_S f(x)\pi(dx).$$

almost surely.

Proof. We rely here on the technical result [8, Theorem 2] which shows that the stated convergence in Theorem 1 follows from conditions (i)–(iii) listed in [8, Theorem

2]. We refer to the stated paper for the exact formulation of these conditions. They can be roughly stated as follows:

(i) Each 'Markovian step' of the algorithm is uniformly ergodic with uniform bounds for the ergodicity.

(ii) The stationary distribution of every 'Markovian step' of the algorithm approximates π quickly enough as time increases.

(iii) The dependence of the chain on fixed size time intervals of the past decreases along time.

The verification of these conditions is fairly straightforward modification of the argument in [8, Theorem 1]. However, for the readers convenience we present below a sketch of the argument.

For simplicity let $d = 2$; the general case is analogous. Just as in the case of the AM algorithm [8] we define the 'freezed' t.p.f. $K_{n,\widetilde{y}_{n-2}}$ (for a given fixed $(n-2)$-tuple $(y_0, y_1, y_2, \ldots, y_{n-2}) \in S^{n-1}$, see [8, p. 229]. From the definition of the SCAM algorithm it follows that the generalized transition probability can be written as the product

$$K_n(y_0, \ldots, y_{n-1}; A) = (K_n^1 K_n^2)(y_0, \ldots, y_{n-1}; A),$$

where K_n^1 corresponds to updating the first variable:

$$K_n^1(y_0, \ldots, y_{n-1}; A) = M_n^1(y_{n-1}; A).$$

Above M_n^1 refers to 1-dimensional Metropolis kernel with respect to the first coordinate and with Normally distributed proposal density $Q_{C_n^1}$. We shall in general denote by $N_U(x-y)$ the 1-dimensional density of Normal distribution centered at y and with covariance $U \in \mathbb{R}$.

Hence, we may write for measurable subsets $B \subset \mathbb{R}$

$$\begin{aligned} M_n^1(y_{n-1}; B \times \{y_{n-1}^2\}) &= \int_B N_{C_n^1}(x - y_{n-1}^1) \min\left(1, \frac{\pi(x, y_{n-1}^2)}{\pi(y_{n-1}^1, y_{n-1}^2)}\right) dx \\ &+ \chi_B(y_{n-1}^1) \int_{\mathbb{R}} N_{C_n^1}(x - y_{n-1}^1) \left[1 - \min\left(1, \frac{\pi(x, y_{n-1}^2)}{\pi(y_{n-1}^1, y_{n-1}^2)}\right)\right] dx. \end{aligned}$$

Of course, one has that $M_n^1(y_{n-1}; \mathbb{R} \times (\mathbb{R} \setminus \{\mathrm{y}_{\mathrm{n}-1}^2\}))=0$. K_n^2 and M_n^2 are defined analogously: $K_n^2(y_0, \ldots, y_{n-1}; A) = M_n^2(y_{n-1}; A)$, together with

$$\begin{aligned} M_n^2(y_{n-1}; \{y_{n-1}^1\} \times B) &= \int_B N_{C_n^2}(x - y_{n-1}^2) \min\left(1, \frac{\pi(y_{n-1}^1, x)}{\pi(y_{n-1}^1, y_{n-1}^2)}\right) dx \\ &+ \chi_B(y_{n-1}^2) \int_{\mathbb{R}} N_{C_n^2}(x - y_{n-1}^2) \left[1 - \min\left(1, \frac{\pi(y_{n-1}^1, x)}{\pi(y_{n-1}^1, y_{n-1}^2)}\right)\right] dx. \end{aligned}$$

Let then $\widehat{M}^1$ be the t.p.f. defined by the same formula as M_n^1, the only difference being that the factor $N_{C_n^1}(x - y_{n-1}^1)$ is replaced by the constant $1/(2a)$ throughout:

$$\widehat{M}^1(y_{n-1}, B \times \{y_{n-1}^2\}) = \left[\frac{1}{2a} \int_B \left(1, \frac{\pi(x, y_{n-1}^2)}{\pi(y_{n-1}^1, y_{n-1}^2)}\right) dx + \ldots\right].$$

Now $\widehat{M}^1$ (and the analogously defined $\widehat{M}^2$) correspond to the algorithm where proposals are uniform: actually $\widehat{M}^1\widehat{M}^2 = \widehat{M}_a$. The assumption (*) now yields a probability measure λ on S and a constant $c > 0$ such that

$$\widehat{M}^1\widehat{M}^2(y; A) = \widehat{M}_a(y; A) \geq c\lambda(A)$$

for all $y \in \mathbb{R}^2$ and for all $A \subset \mathbb{R}$. Whenever $B \times \{y_n^2\} \subset \mathrm{supp}(\pi)$ we have by (4) and the boundedness of $\mathrm{supp}(\pi)$ that $N_{C_n^1}(x - y_{n-1}^1) \geq c_1$ for all values of interesting x and y_{n-1}^1. Similarly for $N_{C_n^2}$. Thus

$$K_{n,\widetilde{y}_{n-2}}(x, A) = (M_n^1 M_n^2)(x; A) \geq c(c_1/2a)^2\lambda(A),$$

which is known to yeald $\delta(K_{n,\widetilde{y}_{n-2}}) \leq 1 - c(c_1/2a)^2$, and proves (i). Here $\delta(\cdot)$ stands for the Dobrushin coefficient of ergodicity of a transition probability function (see [8, p.228]).

The proof of (iii) and (ii) are analogous to the argument in [8, p.230]; exactly in a similar manner one estimates the differences

$$\|M_{n,\widetilde{y}^i_{n-2}} - M_{n+k,\widetilde{y}^i_{n+k-2}}\| \quad (i = 1, 2)$$

which then yields the needed estimate via the formula

$$\begin{aligned} K_{n,\widetilde{y}_{n-2}} - K_{n+k,\widetilde{y}_{n+k-2}} &= M^2_{n,\widetilde{y}_{n-2}}\left(M^2_{n,\widetilde{y}_{n-2}} - M^2_{n+k,\widetilde{y}_{n+k-2}}\right) \\ &+ \left(M^1_{n,\widetilde{y}_{n-2}} - M^1_{n+k,\widetilde{y}_{n+k-2}}\right) M^2_{n+k,\widetilde{y}_{n+k-2}}. \square \end{aligned}$$

Remark In recent work [2] the non-Markovian approach of [8] is generalized in an interesting manner. In addition, the preprint [1] introduces a new approach to the establish convergence of non-Markovian chains, which allows one to get rid of the boundedness conditions for the target density and its support. However, one must add some extra regularity for the density function. Most likely this approach can be applied to our setup here. However, [1] does not yet consider the convergence of ergodic averages, which the authors plan to deal with in a future work, whose manuscript is not available yet. Hence we chose to present the above simple proof here.

Acknowledgments

This work has been supported by the Academy of Finland, MaDaMe project. We would like to thank Dr. Antonietta Mira for useful discussions on the efficiency of MCMC algorithms, Prof. Peter J. Green for providing the code for computing the integrated autocorrelation values and Harri Auvinen for comments on GOMOS one-step problem.

References

[1] C. Andrieu, E. Moulines, and P.Priouret. Stability of stochastic approximation under verifiable conditions. Preprint, 2002.

[2] C. Andrieu and C. P. Robert. Controlled MCMC for optimal sampling. Preprint, 2001.

[3] J. Besag and P. J. Green. Spatial statistics and bayesian computation. *J.R. Statist. Soc. B.*, 55(1):25–37, 1993.

[4] D. Chauveau and P. Vandekerkhove. Improving convergence of the Hastings-Metropolis algorithm with a learning proposal. Preprint, 1999.

[5] A. G. Gelman, G. O. Roberts, and W. R. Gilks. Efficient Metropolis jumping rules. In J. M. Bernardo, J. O. Berger, A. F. David, and A. F. M. Smith, editors, *Bayesian Statistics V*, pages 599–608. Oxford Univ. Press, New York, 1996.

[6] W.R. Gilks, G.O. Roberts, and S.K. Sahu. Adaptive Markov chain Monte Carlo through regeneration. *J. Amer. Statist. Assoc.*, 93:1045–1054, 1998.

[7] H. Haario, E. Saksman, and J. Tamminen. Adaptive proposal distribution for random walk Metropolis algorithm. *Comput. Stat.*, 14:375–395, 1999.

[8] H. Haario, E. Saksman, and J. Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242, 2001.

[9] K. M. Hanson and G. S. Cunningham. Posterior sampling with improved efficiency. In K. M. Hanson, editor, *Proc. SPIE, Medical Imaging: Image Processing*, volume 3338, pages 371–382, 1998.

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

[11] D. Higdon, H. Lee, and C. Holloman. Markov chain Monte Carlo-based approaches for inference in computationally intensive inverse problems. In J. M. Bernardo, editor, *Bayesian Statistics VII, Invited papers*, pages 121–135. Universitat de Valencia, 2002.

[12] J. Kaipio, V. Kolehmainen, E. Somersalo, and M. Vauhkonen. Statistical inversion and Monte Carlo sampling methods in electrical impedance tomography. *Invers Problems*, 16:1487–1522, 2000.

[13] E. Kyrölä, E. Sihvola, Y. Kotivuori, M. Tikka, T. Tuomi, and H. Haario. Inverse theory for occultation measurements, 1, Spectral inversion. *J. Geophys. Res.*, 98:7367–7381, 1993.

[14] J. S. Liu, J. L. Zhang, M. L. Palumbo, and C. E. Lawrence. Bayesian clusetering with variable and transformation selections. In J. M. Bernardo, editor, *Bayesian Statistics VII, Invited papers*, pages 235–257. Universitat de Valencia, 2002.

[15] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machine. *J. Chem. Phys.*, 21:1087–1091, 1953.

[16] S. K. Sahu and A. A. Zhigljavsky. Self regenerative Markov chain Monte Carlo with adaptation. Preprint, 1999.

[17] A. D. Sokal. Monte Carlo methods in statistical mechanics: Foundations and new algorithms. Lecture Notes, Cours de Troisieme Cycle de la Physicque en Suisse Romade, 1989.

[18] J. Tamminen and E. Kyrölä. Bayesian solution for nonlinear and non-gaussian inverse problems by markov chain monte carlo method. *J. Geophys. Res.*, 106(13):14377–14390, 2001.

[19] L. Tierney and A. Mira. Some Adaptive Monte Carlo Methods for Bayesian Inference. *Statistics in Medicine*, 18:2507–2515, 1999.

[20] G. R. Warnes. The normal kernel coupler: An adaptive markov chain monte carlo method for efficiently sampling from multi-modal distributions. Technical Report no 39, Department of Statistics, University of Washington, 2001.