Bayesian Analysis (2012) **7**, Number 3, pp. 715–736

Efficient MCMC for Climate Model Parameter Estimation: Parallel Adaptive Chains and Early Rejection

Antti Solonen[*], Pirkka Ollinaho[†], Marko Laine[‡], Heikki Haario[§], Johanna Tamminen[¶] and Heikki Järvinen[‖]

Abstract. The emergence of Markov chain Monte Carlo (MCMC) methods has opened a way for Bayesian analysis of complex models. Running MCMC samplers typically requires thousands of model evaluations, which can exceed available computer resources when this evaluation is computationally intensive. We will discuss two generally applicable techniques to improve the efficiency of MCMC. First, we consider a parallel version of the adaptive MCMC algorithm of Haario et al. (2001), implementing the idea of inter-chain adaptation introduced by Craiu et al. (2009). Second, we present an early rejection (ER) approach, where model simulation is stopped as soon as one can conclude that the proposed parameter value will be rejected by the MCMC algorithm.

This work is motivated by practical needs in estimating parameters of climate and Earth system models. These computationally intensive models involve nonlinear expressions of the geophysical and biogeochemical processes of the Earth system. Modeling of these processes, especially those operating in scales smaller than the model grid, involves a number of specified parameters, or 'tunables'. MCMC methods are applicable for estimation of these parameters, but they are computationally very demanding. Efficient MCMC variants are thus needed to obtain reliable results in reasonable time. Here we evaluate the computational gains attainable through parallel adaptive MCMC and Early Rejection using both simple examples and a realistic climate model.

Keywords: Adaptive MCMC, Climate Models, Parallel MCMC, Early Rejection

1 Introduction

Markov chain Monte Carlo (MCMC) methods have opened a way for Bayesian statistical analysis of nonlinear mathematical models. However, as the MCMC methods typically require thousands of model evaluations, it is challenging to apply them to problems where the model evaluation is computationally time consuming. One such case of great relevance is parameter estimation in climate models. In a review paper on parameter estimation methods in climate modeling by Annan and Hargreaves (2007), MCMC was

[*]Lappeenranta University of Technology, Lappeenranta, Finland, Antti.Solonen@lut.fi
[†]Finnish Meteorological Institute, Helsinki, Finland, pirkka.ollinaho@fmi.fi
[‡]Finnish Meteorological Institute, Helsinki, Finland, marko.laine@fmi.fi
[§]Lappeenranta University of Technology, Lappeenranta, Finland, heikki.haario@lut.fi
[¶]Finnish Meteorological Institute, Helsinki, Finland, johanna.tamminen@fmi.fi
[‖]Finnish Meteorological Institute, Helsinki, Finland, heiki.jarninen@fmi.fi

considered too computationally expensive for the task. Villagran et al. (2008) found that modern adaptive MCMC methods perform well with surrogate climate models. Järvinen et al. (2010) showed that efficient adaptive variants of MCMC can be used to estimate the distribution of climate model parameters using a full climate model. However, in the latter approach, producing an MCMC run with adequate chain length is computationally demanding. For instance, in our climate model experiments, the computation took about 2 months, although the lowest realistic model resolution was used. For extensive MCMC experimentation and higher model resolution, ways to speed up MCMC are welcome. In this paper, we present two techniques for this purpose: parallel adaptive MCMC and Early Rejection (ER).

Due to the recent developments in high performance computing towards massively parallel systems, it is important to make sure that algorithms can be run efficiently in parallel. In this paper, we present a parallel adaptive MCMC implementation, combining the Adaptive Metropolis (AM) algorithm by Haario et al. (2001) with the concept of inter-chain adaptation introduced by Craiu et al. (2009). We focus on the implementation of adaptive parallel chain MCMC and test its performance against adaptive single-chain algorithms.

In (Craiu et al. 2009), the parallel chain approach was motivated by multimodal target distributions. Here we use parallel chains in order to get samples faster in terms of wall-clock time, and employ inter-chain adaptation to accelerate proposal tuning. In climate modeling, the model code itself enables parallel computing, but the parallelization usually scales efficiently only up to a characteristic number of processors (in our case, a few dozen). Thus, parallel MCMC methods can be used to further utilize the available high-performance computing resources.

In the Early Rejection technique, the model simulation using the proposed parameter value will be terminated as soon as we can conclude that the proposed value will be rejected in the MCMC algorithm. ER is based on a simple re-organization of the calculations in the MCMC algorithm. The trick of inverting the order of simulation and likelihood evaluation has appeared earlier in the literature. It was employed in (Beskos et al. 2006a; Papaspiliopoulos and Roberts 2008) under the title Retrospective Simulation for certain diffusion processes, in order to implement in finite time a simulation for infinite dimensional random variables. See (Beskos et al. 2006b; Dunson and Park 2008) for further applications. Here, we want to emphasize the use of this simple idea as a routine part of basic Metropolis sampling. For a rather wide class of usual estimation problems it provides a way to save CPU time by allowing rejection at an early stage of likelihood evaluation. ER is not approximative: it produces exactly the same results as a full MCMC run, but always faster. We demonstrate by nontrivial examples typical situations where the benefit is most substantial.

The paper is organized as follows. In Section 2 we discuss the climate model parameter estimation problem and the requirements for MCMC in climate model studies. In Section 3 we recall the basics of adaptive MCMC, discuss the parallel adaptive MCMC algorithm and give some numerical examples. The Early Rejection method is described, with examples, in Section 4. In Section 5, we give some specific remarks related to both parallel MCMC and ER. Section 6 concludes the paper.

2 Efficient MCMC for Climate Models

Climate and Earth system models are computational tools in climate science, derived for the most part from first principles. Formally, they consist of coupled non-linear partial differential equations that are solved in discrete form using a computational grid. They are used in a wide variety of research questions, such as assessing the impacts of anthropogenic greenhouse gas emissions on the Earth's climate system.

Many important climatic processes, such as cloud formation, operate in much smaller scales than the grid interval, and they cannot be explicitly represented in the grid. The net effects of these unresolved processes are included in the models using parameterizations that utilize variables which are resolved in the grid. These parameterizations are not limited to climate modeling: for instance, in fluid dynamical models, turbulence due to unresolved scales of motion is expressed by parameterization schemes. These schemes contain predefined closure parameters, that 'close' the further modeling of unresolved processes. Predefining the closure parameter values leads to parametric uncertainty in the models, which may be challenging to estimate. In climate modeling, the closure parameters are typically related to sub-grid scale processes, such as cloud microphysics. For example, the changes in the amount of liquid water and ice in a grid cell can be estimated using a suitable parameterization, based on the grid-mean temperature, humidity, and pressure. Latent heating/cooling due to condensation/evaporation then feeds back to these larger scale state variables.

The closure parameters act as 'tuning handles' of the simulated climate. Currently, optimal closure parameter values are based mostly on expert knowledge, using a relatively small number of simulations. Thus, the climate model tuning procedure contains a subjective element which is open for criticism: above all, no proper uncertainty estimate is obtained, nor is the procedure entirely transparent.

Recently, attempts have been made to develop algorithmic approaches for closure parameter estimation (Jackson 2009; Järvinen et al. 2010). Järvinen et al. (2010) applied an adaptive MCMC method for estimating the joint posterior probability density of a small number ($n = 4$) of closure parameters appearing in the ECHAM5 climate model (see Roeckner et al. 2003, for model description). While it was shown that MCMC is a viable option for closure parameter estimation, the applied model resolution was relatively low and running sufficiently long MCMC chains took a very long time (1–3 months). Thus, ways to speed-up MCMC are needed to make parameter estimation practically more attainable.

3 Adaptive MCMC and Parallel Chains

One of the most popular MCMC algorithms for drawing samples from the posterior distribution $\pi(\theta|\mathbf{y})$ is the random walk Metropolis-Hastings (MH) algorithm (Metropolis et al. 1953; Hastings 1970). At each MH step i, one randomly selects a candidate value θ_{i+1} from a proposal distribution $q(\cdot|\theta_i)$ that may depend on the current point θ_i. Assuming a symmetric proposal distribution, the proposed point is accepted with probability $\alpha = \min(1, \pi(\theta_{i+1}|\mathbf{y})/\pi(\theta_i|\mathbf{y}))$. If a point is rejected, the previous point is replicated. A random walk constructed this way results in a Markov chain that can be used to approximate the target distribution $\pi(\theta|\mathbf{y})$, see for example (Robert and Casella 2005) for theoretical justification.

The practical difficulty in implementing the MH algorithm is the tuning of the proposal distribution q so that the sampling is efficient. One of the recent improvements of MCMC efficiency has been the introduction of adaptive samplers. In adaptive MCMC, one uses the sample history to automatically tune the proposal distribution 'on-line' as the sampling proceeds, see, e.g., (Gilks et al. 1998; Haario et al. 2001; Roberts and Rosenthal 2007; Andrieu and Moulines 2006) for algorithms and theoretical analysis. In the Adaptive Metropolis (AM) algorithm of Haario et al. (2001) the empirical covariance from the samples obtained so far is used as the covariance of a Gaussian proposal. That is, new candidates are proposed as $\theta_{i+1} \sim N(\theta_i, \mathbf{\Sigma}_i)$, where $\mathbf{\Sigma}_i = \mathrm{Cov}(\theta_1, ..., \theta_i) + \epsilon\mathbf{I}$ and $\mathbf{I}$ is the identity matrix. In the formulation of AM of Haario et al. (2001), the diagonal 'regularization' term was added to the sample covariance to make sure that the covariance stays positive definite and that the method has correct ergodicity properties.[1]

The simplicity of AM adaptation allows for its use in a variety of sampling schemes. The delayed rejection (DR) method by Mira (2001) can be combined with AM, as done by Haario et al. (2006). This method (DRAM) has been shown to be efficient in many applications, see e.g. (Villagran et al. 2008; Smith and Marshall 2008). In DR, when a proposed candidate point in a Metropolis-Hastings chain is rejected, a second stage move is proposed around the current point. For example, one can use downscaled versions of the proposals given by AM adaptation as second stage proposals in DR. This is especially helpful to get the sampler moving (to get accepted points) in the beginning of the MCMC run. In addition, AM adaptation can be performed component-wise, as is done in the single component adaptive Metropolis (SCAM) algorithm of Haario et al. (2005), which can be more efficient in high-dimensional problems.

Parallelizing the adaptive MCMC algorithms has been studied relatively little. Combining parallel computing and MCMC is inherently difficult, since MCMC is serial by nature. In (Brockwell and Kadane 2005), a parallel MCMC implementation in the context of regeneration, introduced by Mykland et al. (1995), was studied. Running many parallel chains independent of each other (see e.g. Rosenthal 2000) may not be satisfactory, since it takes time for each single chain to find the mode(s) of the target and for the proposal to adapt. The question whether it is better to run multiple (non-adaptive) short chains or a single long chain has been considered in many studies, see (see e.g. Gelfand and Smith 1990; Gelman and Rubin 1992; Geyer 1992). In this paper, we do not study this question, since running a single long chain is simply not feasible with computationally intensive models. Instead, we study how the use of parallel adaptive chains

[1]Vihola (2011) has later shown that the term is not needed under certain conditions for the target distribution.

can speed up the mixing of the MCMC chains. For this purpose, we present a parallel chain version of the AM algorithm. To parallelize AM, we use a simple mechanism called inter-chain adaptation, recently introduced by Craiu et al. (2009). In inter-chain adaptation, instead of just running independent adaptive samplers in parallel, one uses the samples generated by all parallel chains to perform proposal adaptation, and the resulting proposal is used for all the chains. This naturally means that one has more points for adaptation and the convergence of every individual MCMC chain is expected to speed up.

The parallel chain approach is rather straightforward to implement. The only difference to running independent parallel AM samplers is that each sampler uses and updates the same proposal covariance. Covariance updating can be performed at any given update interval, for instance using the rank-1 covariance update formulas, see (Haario et al. 2001). Our parallel chain implementation with AM adaptation is given as a flowchart in Figure 1. Note that also more advanced adaptation schemes, such as the DRAM and SCAM methods discussed above, can be easily combined with the inter-chain adaptation.

3.1 Example: Gaussian Target Distribution

Let us first consider a 4-dimensional Gaussian target with zero mean and identity covariance $\mathbf{I}$. Naturally, MCMC is not needed to sample Gaussian variates; the example is used to demonstrate how the proposal distribution develops with and without inter-chain adaptation in a case where the true distribution is known. We start the samplers from the true mean, but with small initial diagonal covariance $\mathbf{\Sigma}_0 = \alpha\mathbf{I}$. We note that this is a common way to tune the proposal if no proper initial guess is available: make the proposal small enough to get the sampler moving and let the adaptation expand the covariance. Here, we examine the convergence of this adaptation: how fast we reach a proposal that correctly samples, for instance, a 50% quantile of the χ^2-distribution[2].

We experiment with different parallel configurations using chains of length 2000. To obtain simulation results with comparable variability, the runs are replicated so that we have 20 chains with each configuration. Thus, the single chain run is replicated 20 times, parallel run with 2 chains is replicated 10 times and so on. The adaptation is performed at every step. In Figure 2, the 50% χ^2 quantiles calculated from the samples are plotted. The quantiles are calculated from batches of 500 consecutive iterations: for an MCMC step, we take all points from the next 500 steps and calculate the quantiles. Figure 2 shows that the inter-chain adaptation speeds up the convergence for individual chains: for example, with 10 parallel chains, individual chains have converged after about 200 MCMC steps, whereas the single chain needs more than 1500 iterations. Note that, interestingly, the acceleration effect seems to saturate as more parallel chains are added: the results with 10 and 20 chains are essentially the same. In addition, Figure

[2]We perform the χ^2 calculations as follows. If $x \sim N(0, C)$, then $x^T C^{-1} x \sim \chi^2_d$, where $d = \dim(x)$. After we have sampled points, we can calculate the proportion of the samples whose χ^2-statistics fall below, say, the 50% probability limit of the χ^2-distribution (q for which $P(x^T C^{-1} x < q) = 0.5$). If the samples are from the correct distribution, this proportion should be close to 50%.

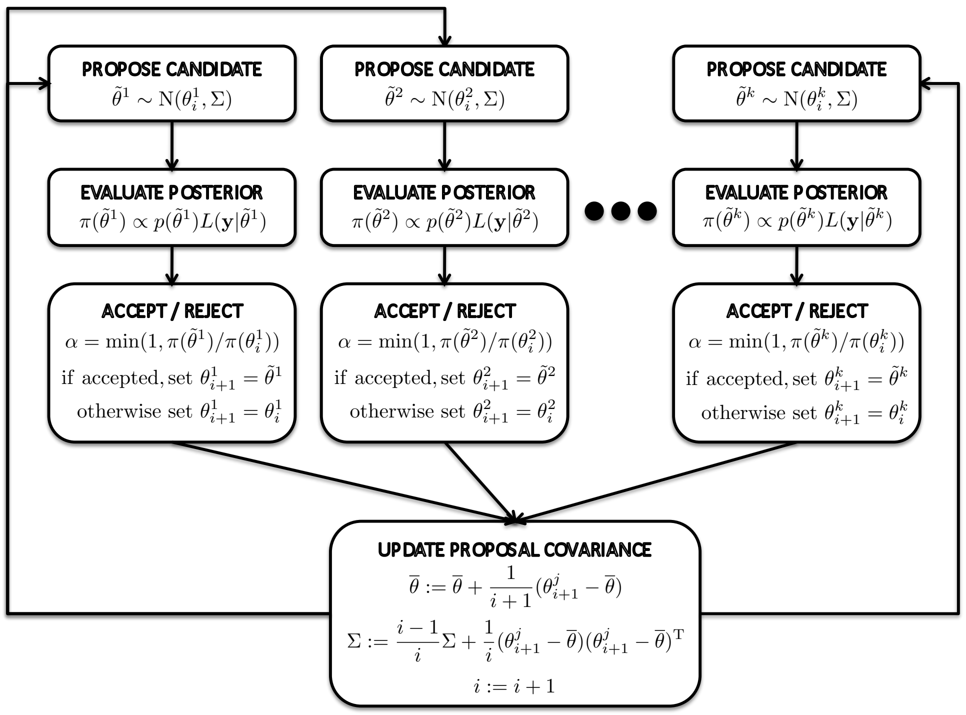

PROPOSE CANDIDATE
$\tilde\theta^1 \sim \mathrm{N}(\theta_i^1, \Sigma)$
PROPOSE CANDIDATE
$\tilde\theta^2 \sim \mathrm{N}(\theta_i^2, \Sigma)$
PROPOSE CANDIDATE
$\tilde\theta^k \sim \mathrm{N}(\theta_i^k, \Sigma)$
EVALUATE POSTERIOR
$\pi(\tilde\theta^1) \propto p(\tilde\theta^1)L(\mathbf{y}|\tilde\theta^1)$
EVALUATE POSTERIOR
$\pi(\tilde\theta^2) \propto p(\tilde\theta^2)L(\mathbf{y}|\tilde\theta^2)$
EVALUATE POSTERIOR
$\pi(\tilde\theta^k) \propto p(\tilde\theta^k)L(\mathbf{y}|\tilde\theta^k)$
ACCEPT / REJECT
$\alpha = \min(1, \pi(\tilde\theta^1)/\pi(\theta_i^1))$
if accepted, set $\theta_{i+1}^1 = \tilde\theta^1$
otherwise set $\theta_{i+1}^1 = \theta_i^1$
ACCEPT / REJECT
$\alpha = \min(1, \pi(\tilde\theta^2)/\pi(\theta_i^2))$
if accepted, set $\theta_{i+1}^2 = \tilde\theta^2$
otherwise set $\theta_{i+1}^2 = \theta_i^2$
ACCEPT / REJECT
$\alpha = \min(1, \pi(\tilde\theta^k)/\pi(\theta_i^k))$
if accepted, set $\theta_{i+1}^k = \tilde\theta^k$
otherwise set $\theta_{i+1}^k = \theta_i^k$
UPDATE PROPOSAL COVARIANCE
$\bar\theta := \bar\theta + \frac{1}{i+1}(\theta_{i+1}^j - \bar\theta)$
$\Sigma := \frac{i-1}{i}\Sigma + \frac{1}{i}(\theta_{i+1}^j - \bar\theta)(\theta_{i+1}^j - \bar\theta)^{\mathrm{T}}$
$i := i + 1$

Figure 1: Inter-chain adaptation. Several chains are run in parallel independent of each other, but the proposal covariance is adapted using the output from all chains. The ith sample of the jth parallel chain is denoted by θ_i^j and the proposed candidate for chain j by $\tilde\theta^j$. In the proposal covariance update, the chain index is $j \in 1, ..., k$. The notation $:=$ means substitution.

2 illustrates the chain evolution from the 10 single adaptive chains and 10 parallel chains with inter-chain adaptation. One can clearly see the improvement in 'adaptation convergence speed': the inter-chain adaptation is able to inflate the too small initial covariance faster than the single-chain adaptation.

3.2 Example: Climate Model Closure Parameter Estimation

We have implemented the parallel chain algorithm with DRAM adaptation (Haario et al. 2006) for the ECHAM5 climate model closure parameter estimation problem. We sample from the distribution $\pi \propto \exp(-J(\theta)/2)$, where $J(\theta)$ is a specifically chosen cost function that compares the model simulation output to observed data. Here, we used annual global and monthly zonal averages of net radiation at the top of the atmosphere

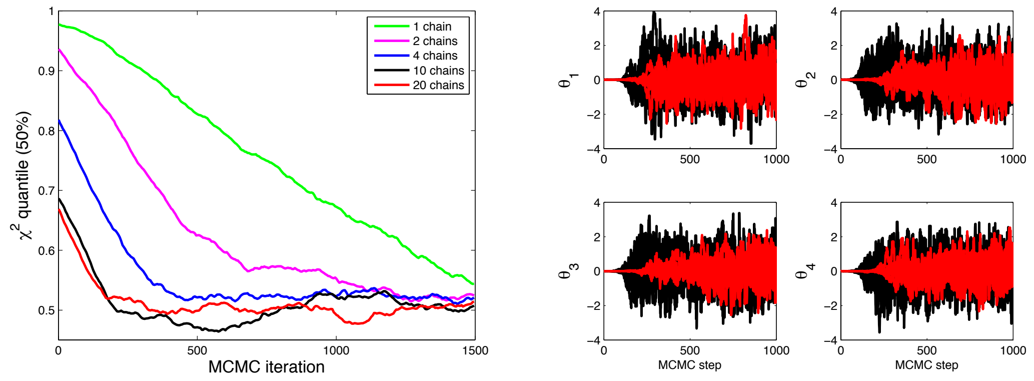


Figure 2: Left: convergence comparison with a varying number of parallel chains with initial covariance $\Sigma_0 = \alpha\mathbf{I}$ where α =1e-9. The theoretically correct value for the χ^2 quantile is 0.5. Right: 10 single adaptive chains (red) and 10 parallel chains with inter-chain adaptation (black).

as a cost function, defined as follows:

$$J(\theta) = \frac{(F(\theta) - F^o)^2}{\sigma^2} + \sum_{t=1}^{12}\sum_{y} \frac{(F_{t,y}(\theta) - F^o_{t,y})^2}{\sigma^2_{t,y}}, \tag{1}$$

where $F(\theta)$ is the annual mean flux and $F_{t,y}(\theta)$ is the mean flux for month t and zonal band y, simulated with the climate model with closure parameter values θ. The same quantities calculated from observation data sets are denoted in the formulas by superscript o and variances are denoted by σ^2. The cost function $J(\theta)$ is calculated by running the climate model with parameters θ for one year, computing the annual and monthly zonal means from the output, and comparing the results to the same quantities calculated from observations. In this example, we use model simulations with default parameter values as the observations, instead of real data, since we want to study the identifiability of the parameter estimation problem in the ideal case, without having to consider, e.g., bias in the model. A uniform distribution with rather wide upper and lower bounds for the parameters is used as the prior. The variances σ^2, that essentially define how large deviations from the observations are allowed, are assumed to be known and are estimated from climatological data.

The two-dimensional marginals of the parameter distributions are given in Figure 3. One can see that the parallel algorithm captures the posterior more comprehensively, while the single chain approach has not yet explored the entire distribution: the single-chain sampler has not yet reached the tail of the distribution that is found by the parallel sampler. Note that if one just had the single chain results, the conclusions about the parameters would be rather different from those obtained using the results of the parallel algorithm. From the practical point of view, the parallel algorithm was beneficial here in terms of obtaining the results in a more reasonable elapsed time. Finally, the 'exploration speed' is illustrated in Figure 4, which indicates that for two out of four parameters the speed-up effect is very pronounced.

Although the main focus of this paper is methodological, we point out that the conclusion from this run is that one cannot identify the parameters by looking only at the net radiation averages. Ongoing research attempts to define a cost function that can uniquely determine the parameters. In any case, we note that this kind of insensitivity of parameters can be revealed by MCMC even with highly demanding models, supposing that sampling with effective mixing can be achieved. For details about the closure parameter estimation problem and comprehensive parameter estimation results with this climate model, we refer to (Järvinen et al. 2010).

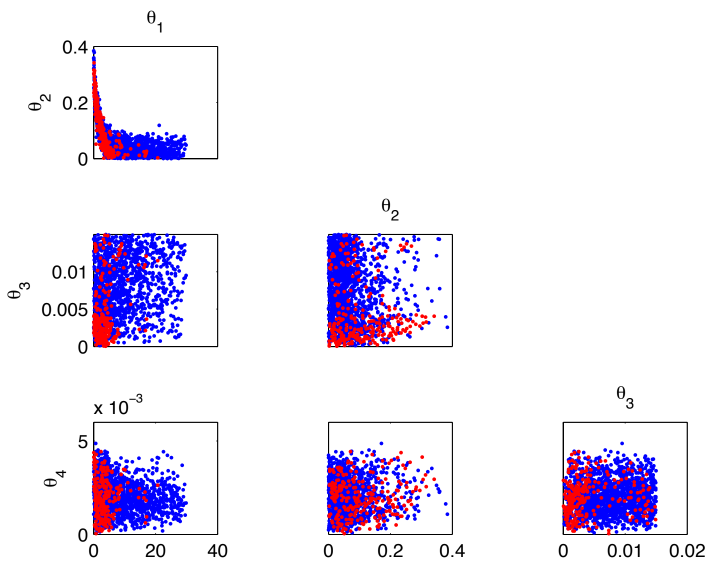


Figure 3: Paired marginal posterior distributions from a single chain (red) and 7 parallel chains (blue) using 1000 MCMC iterations per chain.

4 Early Rejection by Retrospective Simulation

Application of MCMC to parameter estimation problems typically requires a large number of model evaluations. When the simulation model is time consuming, the elapsed time for obtaining reliable results can be prohibitively long.

In addition to parallelization, other approaches have been presented to make MCMC samplers more suitable for cases where model simulation is time consuming. For exam-

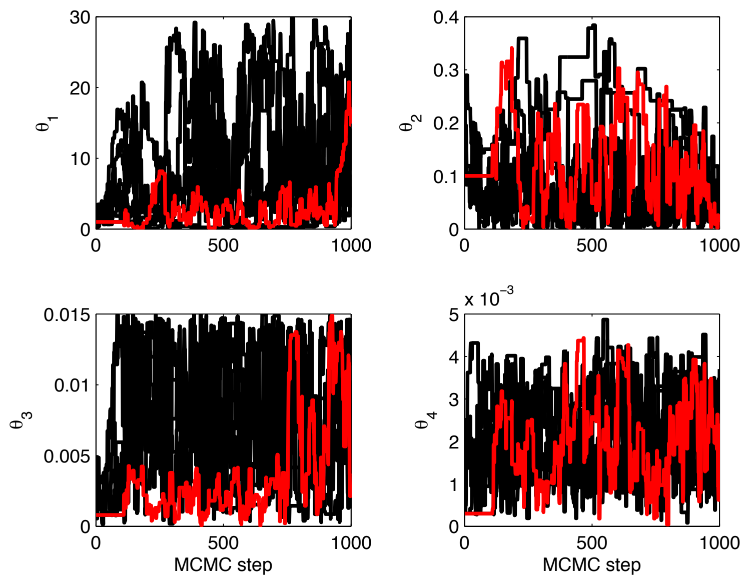


Figure 4: MCMC chains from a single chain (red) and 7 parallel chain (black) experiments.

ple, Drignei et al. (2008) and Villagran et al. (2008) suggest to replace the computationally intensive climate models with approximative, empirical surrogate models that are fast to evaluate. Christen and Fox (2005) show how MCMC can be carried out in two stages, if one has an approximative model at hand. In their delayed acceptance approach, the parameter likelihood is evaluated with the full model only if it was accepted by the approximative model.

Here we present a simple, non-approximative trick for an early rejection (ER) in Metropolis-Hastings based MCMC algorithms. The idea is to stop the simulation as soon as we know that the proposed parameter will be rejected.

Suppose the current parameter value in the MH algorithm is θ_i. Recall that MH proceeds by proposing a candidate value θ_{i+1} and accepting the proposed value with probability $\alpha = \min(1, \pi(\theta_{i+1})/\pi(\theta_i))$, assuming that the proposal density is symmetric. In practice, one first evaluates $\pi(\theta_{i+1})$, then simulates a uniform random number $u \sim U(0, 1)$ and accepts θ_{i+1} if $u < \alpha$. Thus, a point will be rejected if $u > \pi(\theta_{i+1})/\pi(\theta_i)$.

Let us assume that the likelihood can be divided into n independent parts: $\pi(\theta) \propto p(\theta) \prod_{i=1}^{n} L(y_i|\theta)$. Moreover, let us denote by $\pi_k(\theta)$ the unnormalized posterior density when k parts (where $k \leq n$) are considered: $\pi_k(\theta) = p(\theta) \prod_{i=1}^{k} L(y_i|\theta)$. Assume that the density $\pi_k(\theta)$ is monotonically decreasing with respect to the index k. This is the case, for example, if the likelihood has an exponential form $L(y|\theta) \propto \exp(-l(y|\theta))$, with $l(y|\theta) \geq 0$. In this case, we can reject as soon as $\pi_k(\theta_{i+1})/\pi(\theta_i) < u$ for some value of k. Thus, we can speed up the sampling simply by switching the order of the calculations: generate the random number u first, evaluate the likelihood part by part and check, after each evaluation, if the proposed value will end up being rejected. Note that before evaluating any likelihood terms, we can check if the proposed point will be rejected based on the prior only, even if the likelihood evaluation cannot be divided into blocks. In principle, the prior term could also be computed after likelihood, but the computational cost of evaluating the prior is usually negligible compared to the cost of the likelihood. In ER, it makes sense to organize the calculations so that the computationally simple terms are evaluated first.

This simple, non-approximative ER mechanism can be applied to a large variety of cases: the only requirements are that the likelihood can be factorized and that the factorized posterior $\pi_k(\theta) = p(\theta)\prod_{i=1}^{k} L(y_i|\theta)$ is monotonically decreasing with respect to k. There are no other specific requirements for the form of the prior and the likelihood. The amount of calculation saved by ER depends on the problem (amount of data, properties of the model, shape of the parameter distribution) and on the tuning of the proposal. In cases where the topology of the parameter posterior distribution is complicated (strongly nonlinear, multimodal), the MH sampler, even if properly tuned, results in low acceptance rates and potentially large performance gains can be achieved through ER. If the model equations exhibit strongly chaotic behavior with respect to the model parameters, many proposed parameters will be rejected early, making ER beneficial. In addition, if too large a proposal covariance is used, many points are rejected and ER is beneficial again. Below, we demonstrate ER with these kind of examples.

4.1 Example: Nonlinear Correlations Between Parameters

The most common proposal distribution used with MH sampling is Gaussian with a fixed covariance. Finding an 'optimal' proposal covariance may require exhausting hand–tuning. The adaptive methods indeed aim at an automatic convergence of the initial proposal towards a more optimal one. Well–known criteria exist for optimality of Gaussian proposals for Gaussian targets, see e.g. (Roberts and Rosenthal 2001). Often, such proposals perform well also for mildly non-Gaussian targets, see e.g. the synthetic examples of Haario et al. (1999) and the application studies of Malve et al. (2005); Vahteristo et al. (2009).

A problem remains if the parameter posterior exhibits strong nonlinear correlations. Such 'banana-shaped' distributions are in fact quite common for nonlinear models loaded with many poorly identified parameters. In such cases, a Gaussian proposal does not work well: the acceptance rate remains high only if the proposal distribution is narrow. As the size of the proposal grows to better match the size of the target – as with the AM adaptation, for example – the nonlinearity necessarily causes frequent proposals that miss the essential support of the posterior, and the sampling becomes ineffective due to low acceptance rates. In such situations we can expect that, when applicable, ER will be helpful: if a proposal is clearly outside the posterior with a low likelihood, the evaluation of the expressions $\pi_k(\theta_{i+1})/\pi(\theta_i)$ should, in average, drop below the rejection threshold at low values of k.

We demonstrate the situation with the simple model $y = b_1(1-\exp(-b_2 x))+\epsilon$, where the true parameter value is given as $\theta = (b_1, b_2) = (1, 0.2)$ and the measurements are linearly spaced on the interval $0 \le x \le x_{\max}$. In our examples we use 20 measurements ($n = 20$), but the results are roughly the same if the number of data points is varied between, say, 10 and 500. The noise term ϵ is $N(0, \sigma^2)$ with $\sigma = 0.03$. We use improper uniform priors $p(\theta) \propto 1$ for the parameters.

We show results for two different values for $x_{\max}$ that lead to different types of posteriors. First, we concentrate the observations to small values of x and set $x_{\max} = 4$. This produces a difficult case, where the parameters are rather poorly identified and a strong nonlinear correlation between the parameters is present. Setting $x_{\max} = 10$ gives an easier, almost Gaussian posterior. In both cases, we run 50000 steps of MH with a fixed proposal that is tuned beforehand using AM. In Figure 5, the parameter posteriors and tuned proposal covariances are illustrated for both cases. The histograms of the measurement indices where early rejection happens are also given. Recall that $n = 20$ and therefore when the rejection index k is equal to 20 it means that an acceptance is obtained. In the strongly nonlinear case, many proposed parameter values miss the region of high probability and are rejected early. In this case, ER saved roughly 50% of the model computations. In the easier case, the proposal is a better approximation of the target and rejections typically occur only in the end of the run, resulting in about 15% computational savings.

Increasing $x_{\max}$ further leads to even better parameter identification with an approximately Gaussian posterior, and the computational savings drop below 10%. On the other hand, decreasing $x_{\max}$ leads to a more and more difficult posterior, where ER can save up to 80% of the computations. Thus, ER works especially well for this type of strongly correlated, non-Gaussian posteriors, which are intractable without an MCMC type of approach. We note that for 'easy', approximatively Gaussian targets, for which ER is less beneficial, the whole MCMC approach may not be really needed, as the same results can be obtained using classical linear theory, too.

4.2 Example: Chaotic Dynamics in the Lorenz ODE

Here we test the ER procedure using a three-parameter model (Lorenz 1963), written as a coupled system of ordinary differential equations (ODE):

$$\begin{aligned}
\frac{dx}{dt} &= \alpha(y - x), \\
\frac{dy}{dt} &= x(\rho - z) - y, \\
\frac{dz}{dt} &= xy - \beta z.
\end{aligned}$$

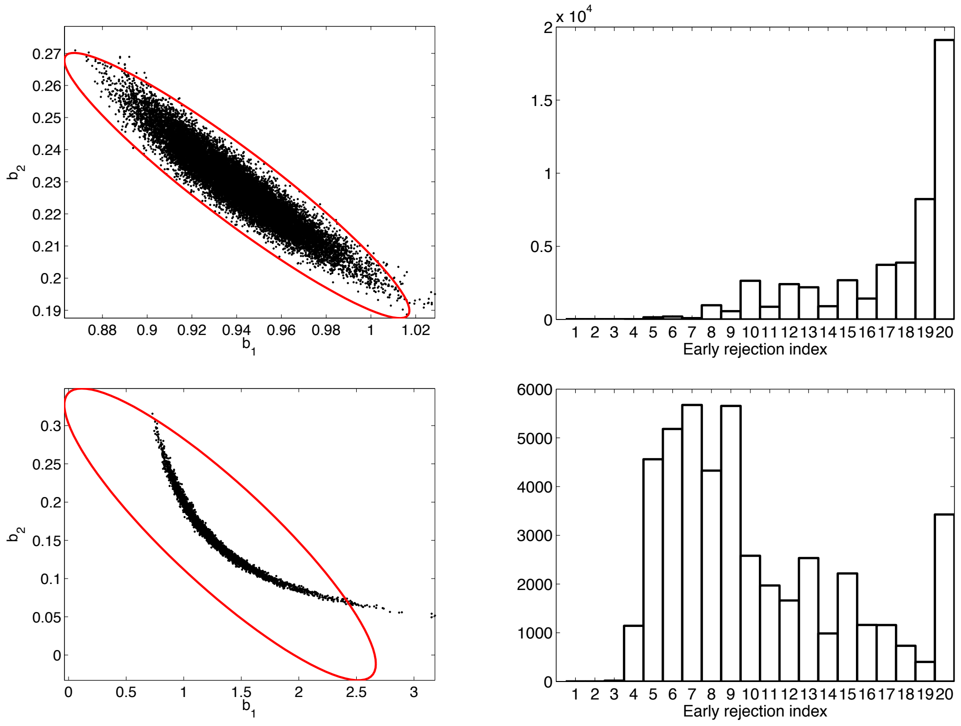


Figure 5: Left: parameter distributions and tuned proposal covariances in the mildly nonlinear (top) and strongly nonlinear (bottom) case. Right: histograms of the observation index when rejection happens (last index corresponds to accepted points).

We use $\alpha = 10$, $\rho = 28$ and $\beta = 8/3$, $x(0) = 26.61$, $y(0) = -2.74$ and $z(0) = 0.95$ as the 'true' parameter and initial values in the simulations. With these parameter values, the model is in its chaotic regime. Gaussian noise with variance $\sigma^2 = 1$ is added to the model responses. Data was generated for 15 points with time interval 0.2.

We consider two problems: estimating the parameters $\theta_{\mathrm{PAR}} = (\alpha, \rho, \beta)$ and estimating the initial values $\theta_{\mathrm{INIT}} = (x(0), y(0), z(0))$, using the measurements (t_i, x_i, y_i, z_i) where $i = (1, ..., 15)$. In both cases, the AM sampler was first run to tune the covariance of the Gaussian proposal, and the experiment was then run with the resulting fixed proposal. This was done to remove the effect of adaptation and proposal tuning from the ER experiment. A uniform prior distribution was used in both cases.

The sampling of the parameters θ_{PAR} works in a routine way, roughly 45% of the proposals are accepted. In contrast, the sampling of the initial values θ_{INIT} — with a similarly tuned proposal — results in a very high rejection rate, higher than 99%. This happens because of the chaotic nature of the initial values: a point that produces a bad fit can be close to an accepted point, while a more far away initial value again fits the

Table 1: Results from computational experiments: in this case, ER saves about 80% in computation time in the initial value estimation case and 30% in the parameter estimation case. ODE counts (how many times the ODE function is evaluated) are given in thousands.

	Parameter estimation		**Initial value estimation**	
	time (sec)	ODE count	time (sec)	ODE count
ER	262.8	1134	69.9	299
MH	376.4	1615	442.5	2092
ratio	0.698	0.702	0.165	0.143

original response, illustrating the fractal nature of the model attractor.

As can be expected, ER speeds up the calculations in both cases. In the parameter estimation case, the improvement is less dramatic. This is natural, since the likelihood decreases 'smoothly' as a function of the observations, and typically most of the likelihood components have to be calculated before the rejection happens. The benefit is clearly higher in the initial value estimation problem: most of the proposals are rejected and many of the rejections are captured early in the model simulation. This happens because the chosen true initial values are in the 'chaotic' region of the state space (close to a bifurcation point of the Lorenz system). This is illustrated in Figure 6, where an example of a rejected simulation is presented for both cases.

For a CPU time comparison, we produced 1000 samples for both cases $\pi(\theta_{\mathrm{PAR}}|\mathbf{y})$ and $\pi(\theta_{\mathrm{INIT}}|\mathbf{y})$ with and without ER. The experiments were performed in MATLAB, using the numerical ODE solver `ode15s`. The absolute and relative tolerances of the solver were both set to 1e-8 to diminish numerical errors (indeed, the simulation results are rather sensitive to these tolerances). The execution time and counts of evaluations of the right hand side of the ODE system are summarized in Table 1. As can be observed from the results, the performance gain is remarkable in the chaotic initial value estimation problem, about 80% reduction of CPU times. In the parameter estimation case, using ER saved roughly 30% of CPU time.

4.3 Example: Climate Model Closure Parameter Estimation

Here, the performance of ER is demonstrated in the context of climate model parameter estimation. The experimental setup is such that the ECHAM5 climate model is run with a similar cost function as in Section 3.2:

$$J(\theta) = \sum_{t=1}^{12} \left(\frac{(F_t(\theta) - F_t^o)^2}{\sigma_t^2} + \sum_y \frac{(F_{t,y}(\theta) - F_{t,y}^o)^2}{\sigma_{t,y}^2}, \right), \tag{2}$$

where $F_t(\theta)$ and $F_{t,y}(\theta)$ are the simulated and F_t^o and $F_{t,y}^o$ the observed monthly global and monthly zonal net radiation averages. Upper and lower bounds for the parameters were given as a prior, see (Järvinen et al. 2010) for details. The average fluxes $F_t(\theta)$ are now calculated separately for each month, compared to the cost function of the example in Section 3.2. The cost function divides into 12 independent parts and the rejection threshold in ER can be checked after every part (after the model has been run for that month). Thus, for a proposed parameter value, we evaluate the model for a month, calculate the part of the cost function for that month and check if the rejection threshold is exceeded. If not, we simulate the next month, update the cost function, check rejection and so on. Without ER, one would run the model for the whole year, evaluate $J(\theta)$ and decide whether to accept or reject based on the results.

In Figure 7, we demonstrate how ER behaves in the climate model case by providing examples on the aggregation of the cost function as more months are simulated, and indicating when early rejections occur. We note that many proposed parameter values are rejected early, some as early as after the second simulated month. However, the relative CPU savings brought by ER were rather small in this case, approximately 19%. This is probably due to the cost function: as seen in Figure 7, the cost function grows smoothly and no sudden jumps to higher cost function values appear. More specifically, out of the 3204 simulation years, ER saved 595 simulation years, equaling approximately 695 hours (29 days) of saved CPU time. Although the relative savings are not too significant, the absolute savings are very beneficial. With ER, we are able to obtain more samples with the available computer resources.

5 Discussion

Besides parallel chain approaches, there are ways to parallelize also single chain MCMC algorithms. Brockwell (2006) and Strid (2009) developed a parallelization scheme called pre-fetching for the single-chain Metropolis-Hastings algorithm, where the possible future steps of the algorithm (rejections and acceptances) are evaluated in advance in parallel. However, the number of possible future paths increases exponentially as a function of the number of future steps simulated in advance, and the pre-fetching approach is efficient if the acceptance rate is either high or low.

Some Monte Carlo estimation methods are trivially parallelizable, or 'embarrassingly parallel', such as independent Metropolis-Hastings samplers and importance sampling based methods, like population Monte Carlo methods (e.g. Cappe et al. 2004) where importance sampling is performed iteratively. However, random walk methods are often preferred over independent samplers, since a good proposal distribution in independent sampling should cover the whole target, which can be difficult to achieve. In addition, the problem in using importance sampling based methods for climate models is that one can often run only a rather small number of model evaluations simultaneously (in our case, maximum 10–20), which might be too small for performing importance resampling efficiently. Novel ways to implement importance sampling, such as the adaptive multiple importance sampling (AMIS) algorithm of Cornuet et al. (2012), could help, but this question is out of the scope of this paper.

There are two ways to implement the communication between the chains in inter-chain adaptation. In the *synchronous* version, the likelihood for all chains is evaluated simultaneously, the covariance is updated and new likelihood evaluations are started. Thus, all samplers have to finish their step before a new step is started. In the *asynchronous* version, parallel samplers just keep running and update the covariance independently. Obviously, the asynchronous version is more CPU-effective, and also more robust in cases where parameter variations proposed by MCMC can lead to abnormal termination of execution of some of the chains. We used the synchronous version for studying the behavior of parallel samplers, since it is easy to keep track of what happens in the algorithm. For real sampling tasks, such as the climate model parameter estimation case, we implemented the algorithm asynchronously.

We have implemented the adaptive parallel chain algorithm in several computing environments. For the climate model experiments, we developed an asynchronous 'script-level' implementation that utilizes the batch job system in the supercomputer. This is a natural approach for large models that are already built to utilize supercomputing environments. Basically, one only needs to implement a script that sends single MCMC steps (model evaluations) to the batch job system and analyzes the results (accepts or rejects, adapts the proposal and proposes the parameter values for the next MCMC step). For smaller models and experimentation, we developed a synchronous code that implements inter-chain adaptation into our existing Fortran90 adaptive MCMC toolbox. The code was built using the Message Passing Interface (MPI), which is a standard technique for parallel programming in distributed memory platforms. We also have a synchronous Matlab implementation that runs on top of the Techila computer grid (see Rantala and Piche 2009, for an introduction) and uses the Matlab MCMC toolbox of Laine (2008). In the Fortran MPI and Matlab implementations, we have a designated 'adaptation node' that receives parameter values from individual samplers and sends the updated proposal covariance back to the samplers. If the likelihood evaluation is relatively fast compared to the covariance adaptation, the CPU overhead caused by such traffic might become relevant. In case the adaptation becomes a bottleneck, perhaps due to a high-dimensional parameter space, one can naturally perform adaptation only at predefined intervals instead of every step, as is commonly done in adaptive MCMC. If the model is expensive to evaluate, the overhead of adaptation is likely insignificant: in our climate model experiments, for instance, the adaptation was performed at every step.

Although the relative benefits achieved by ER can sometimes be small, it is worth noting that, besides coding the method, the computational savings always come at no additional cost and without approximations. Note also that ER seems to be most helpful with difficult posteriors that lead to frequent rejections. In addition to the estimation problems presented, a potential application that can benefit remarkably from ER is approximative Bayesian computation (ABC). In ABC, after a parameter candidate is proposed, a data set is simulated with the model, and the generated data set is compared to the actual data using a certain summary statistic. The proposed point is accepted if the difference between the summary statistics is small enough. The acceptance probability in ABC is commonly very low, and many proposals are rejected, and ER can potentially help to detect the rejections sooner.

The applicability of ER is case dependent. For example, some advanced MCMC methods, such as the Delayed Rejection Adaptive Metropolis (DRAM) of Haario et al. (2006), cannot be used with ER, since their formulation requires the full evaluation of the posterior at a rejected point. The component-wise and the Metropolis–within–Gibbs methods, on the other hand, can benefit from ER. Some specific cost function formulations, such as those based on 'empirical orthogonal functions' (principal components) that are often used in climate science, may not be compatible with ER since their calculation cannot be naturally divided into independent parts if the principal components are calculated over time.

6 Conclusions

In this paper, we present two ways to improve the efficiency of MCMC for computationally intensive models. The work is motivated by practical computing resource limitations faced in parameter estimation of climate and Earth system models. We present a parallel adaptive MCMC approach that combines the Adaptive Metropolis algorithm with inter-chain adaptation. With the parallel adaptive method, it is possible to speed up the convergence of each individual chain. In the climate model case, the parallel approach is crucial when running extensive MCMC experiments with different likelihood formulations. In addition, we introduce an early rejection mechanism for terminating the likelihood evaluation as soon as it is evident that the parameter value under evaluation will be rejected. The effectiveness and applicability of ER depends on the case: in the examples of this paper we achieve a reduction factor that ranges between 10% and 80% in computation times. The benefit of ER is especially pronounced with difficult posteriors, a non-Gaussian target distribution of strongly correlated parameters as a generic example. Both toy examples and a realistic climate model case show that parallel adaptation and early rejection can significantly improve the efficiency of MCMC.

References

Andrieu, C. and Moulines, E. (2006). "On the ergodicity properties of some adaptive MCMC algorithms." *Annals of Applied Probability*, 16(3): 1462–1505. 718

Annan, J. D. and Hargreaves, J. C. (2007). "Efficient estimation and ensemble generation in climate modeling." *Philosophical Transactions of the Royal Society A*, 365: 2077–2088. 715

Beskos, A., Papaspiliopoulos, O., and Roberts, G. (2006a). "Retrospective exact simulation of diffusion sample paths with applications." *Bernoulli*, 12(6): 1077. 716

Beskos, A., Papaspiliopoulos, O., Roberts, G., and Fearnhead, P. (2006b). "Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes (with discussion)." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3): 333–382. 716

Brockwell, A. (2006). "Parallel Markov Chain Monte Carlo Simulation by Pre-fetching." *Journal of Computational and Graphical Statistics*, 15(1): 246–26. 728

Brockwell, A. and Kadane, J. (2005). "Identification of regeneration times in MCMC simulation, with application to adaptive schemes." *Journal of Computational and Graphical Statistics*, 14(2): 436–458. 718

Cappe, O., Guillin, A., Marin, J., and Robert, C. P. (2004). "Population Monte Carlo." *Journal of Computational and Graphical Statistics*, 13: 907–929. 728

Christen, J. A. and Fox, C. (2005). "MCMC using an Approximation." *Journal of Computational and Graphical Statistics*, 14(4): 795–810. 723

Cornuet, J., Marin, J., Mira, A., and Robert, C. (2012). "Adaptive Multiple Importance Sampling." *Scandinavian Journal of Statistics, available on-line in Early View*. URL http://arxiv.org/abs/0907.1254 728

Craiu, R. V., Rosenthal, J., and Yang, C. (2009). "Learn From Thy Neighbor: Parallel-Chain and Regional Adaptive MCMC." *Journal of the American Statistical Association*, 104(488): 1454–146. 715, 716, 719

Drignei, D., Forest, C. E., and Nychka, D. (2008). "Parameter estimation for computationally intensive nonlinear regression with an application to climate modeling." *The Annals of Applied Statistics*, 2(4): 1217–1230. 723

Dunson, D. and Park, J. (2008). "Kernel stick-breaking processes." *Biometrika*, 95(2): 307–323. 716

Gelfand, A. and Smith, A. (1990). "Sampling-Based Approaches to Calculating Marginal Densities." *Journal of the American Statistical Association*, 85: 398–409. 718

Gelman, A. and Rubin, D. B. (1992). "Inference from Iterative Simulation Using Multiple Sequences." *Statistical Science*, 7(4): 457–472. 718

Geyer, C. J. (1992). "Practical Markov Chain Monte Carlo." *Statistical Science*, 7(4): 473–483. 718

Gilks, W., Roberts, G., and Sahu, S. (1998). "Adaptive Markov chain Monte Carlo through regeneration." *Journal of the American Statistical Association*, 93(443): 1045–1054. 718

Haario, H., Laine, M., Mira, A., and Saksman, E. (2006). "DRAM: Efficient adaptive MCMC." *Statistics and Computing*, 16(3): 339–354. 718, 720, 730

Haario, H., Saksman, E., and Tamminen, J. (1999). "Adaptive proposal distribution for random walk Metropolis algorithm." *Computational Statistics*, 14: 375–395. 724

— (2001). "An adaptive Metropolis algorithm." *Bernoulli*, 7(2): 223–242. 715, 716, 718, 719

— (2005). "Componentwise adaptation for high dimensional MCMC." *Computational Statistics*, 20(2): 265–273. 718

Hastings, W. K. (1970). "Monte Carlo sampling using Markov chains and their applications." *Biometrika*, 57(1): 97–109. 718

Jackson, C. S. (2009). "Use of Bayesian inference and data to improve simulations of multi-physics climate phenomena." *Journal of Physics: Conference Series*, 180. SciDAC 2009 14–18 June 2009, San Diego, California, USA. 717

Järvinen, H., Räisänen, P., Laine, M., Tamminen, J., Ilin, A., Oja, E., Solonen, A., and Haario, H. (2010). "Estimation of ECHAM5 climate model closure parameters with adaptive MCMC." *Atmospheric Chemistry and Physics*, 10(20): 9993–10002.
URL `http://www.atmos-chem-phys.net/10/9993/2010/` 716, 717, 722, 728

Laine, M. (2008). *Adaptive MCMC Methods with Applications in Environmental and Geophysical Models*. Finnish Meteorological Institute Contributions, No. 6.
URL `http://www.helsinki.fi/mjlaine/mcmc/` 729

Lorenz, E. N. (1963). "Deterministic nonperiodic flow." *Journal of the Atmospheric Sciences*, 20: 130–141. 725

Malve, O., Laine, M., and Haario, H. (2005). "Estimation of winter respiration rates and prediction of oxygen regime in a lake using Bayesian inference." *Ecological Modelling*, (182): 183–197. 724

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). "Equation of State Calculations by Fast Computing Machines." *The Journal of Chemical Physics*, 21(6): 1087–1092. 717

Mira, A. (2001). "On Metropolis-Hastings algorithms with delayed rejection." *Metron*, LIX(3–4): 231–241. 718

Mykland, P., Tierney, L., and Yu, B. (1995). "Regeneration in Markov Chain Samplers." *Journal of the American Statistical Association*, 90(429): 233–241. 718

Papaspiliopoulos, O. and Roberts, G. (2008). "Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models." *Biometrika*, 95(1): 169–186. 716

Rantala, J. and Piche, R. (2009). "Software Systems for Distributed Scientific Computing." Research Report 95, Tampere University of Technology. Department of Mathematics.
URL `http://math.tut.fi/en/wp-content/uploads/2009/10/report95.pdf` 729

Robert, C. P. and Casella, G. (2005). *Monte Carlo Statistical Methods*. New York: Springer, second edition. 718

Roberts, G. and Rosenthal, J. (2001). "Optimal scaling for various Metropolis-Hastings algorithms." *Statistical Science*, 16(4): 351–367. 724

— (2007). "Coupling and Ergodicity of Adaptive MCMC." *Journal of Applied Probability*, 44(2): 458–475. 718

Roeckner, E., Bäuml, G., Bonaventura, L., Brokopf, R., Esch, M., Giorgetta, M., Hagemann, S., Kirchner, I., Kornblueh, L., Manzini, E., Rhodin, A., Schlese, U., Schulzweida, U., and Tompkins, A. (2003). "The atmospheric general circulation model ECHAM5, Part I Model Description." Technical Report No. 349, Max-Planck-Institut fr Meteorologie. 717

Rosenthal, J. S. (2000). "Parallel Computing and Monte Carlo algorithms." *Far Eastern Journal of Theoretical Statistics*, 4: 207–236. 718

Smith, T. J. and Marshall, L. A. (2008). "Bayesian methods in hydrologic modeling: A study of recent advancements in Markov chain Monte Carlo techniques." *Water Resources Research*, 44(W00B05): 1–9. 718

Strid, I. (2009). "Efficient parallelisation of Metropolis-Hastings algorithms using a prefetching approach." *Computational Statistics and Data Analysis*, 54(11): 2814–2835. 728

Vahteristo, K., Maury, S., Laari, A., Solonen, A., Haario, H., and Koskimies, S. (2009). "Kinetics of neopentyl glycol esterification with different carboxylic acids." *Industial and Engineering Chemistry Research*, 48(13): 6237–6247. 724

Vihola, M. (2011). "Can the Adaptive Metropolis Algorithm Collapse Without the Covariance Lower Bound?" *Electronic Journal of Probability*, 16: 45–75. 718

Villagran, A., Huerta, G., Jackson, C. S., and Sen, M. K. (2008). "Computational Methods for Parameter Estimation in Climate Models." *Bayesian Analysis*, 3(3): 1–27. 716, 718, 723

Acknowledgments

The authors wish to thank Dr. Harri Auvinen for contributing in the development of the Fortran90 MPI implementation of the parallel MCMC algorithm. Moreover, we thank the editor, associate editor and an anonymous reviewer for reviewing the paper and suggesting useful improvements. This research has been supported by the Academy of Finland projects 132808, 134937 and 134999.

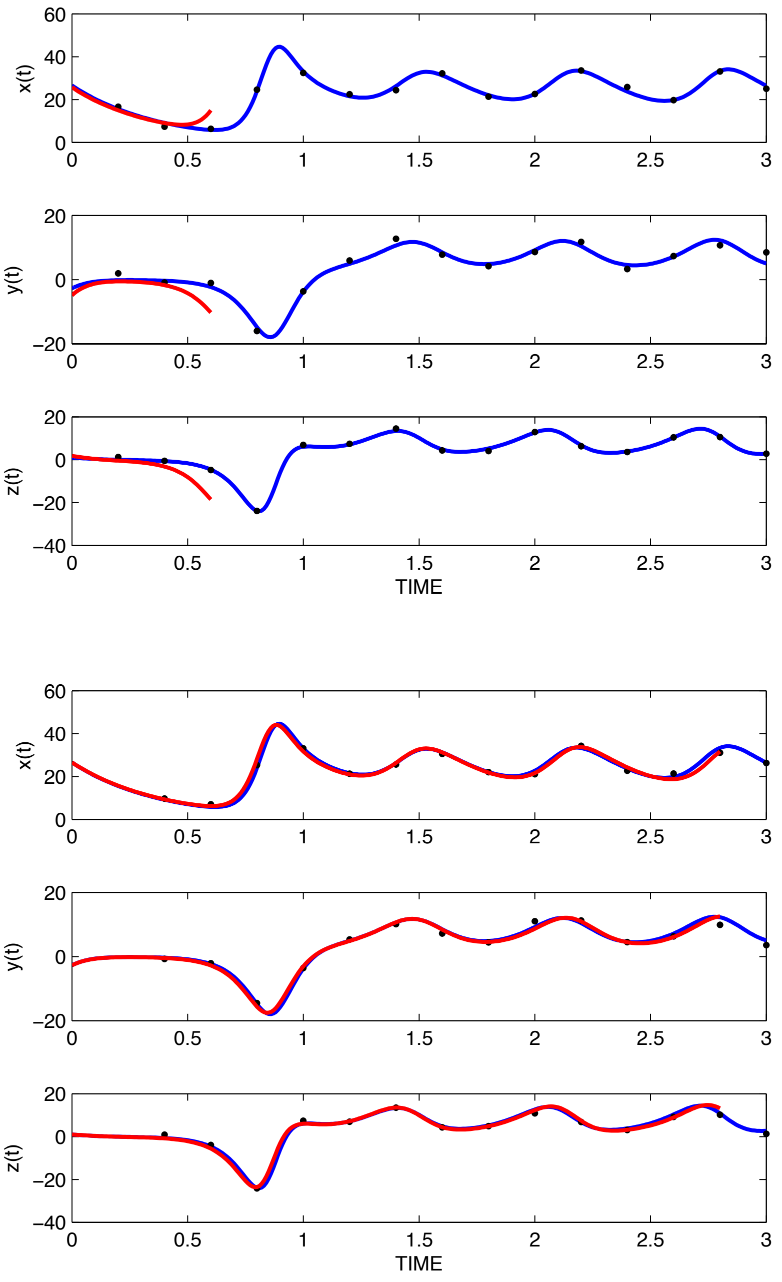
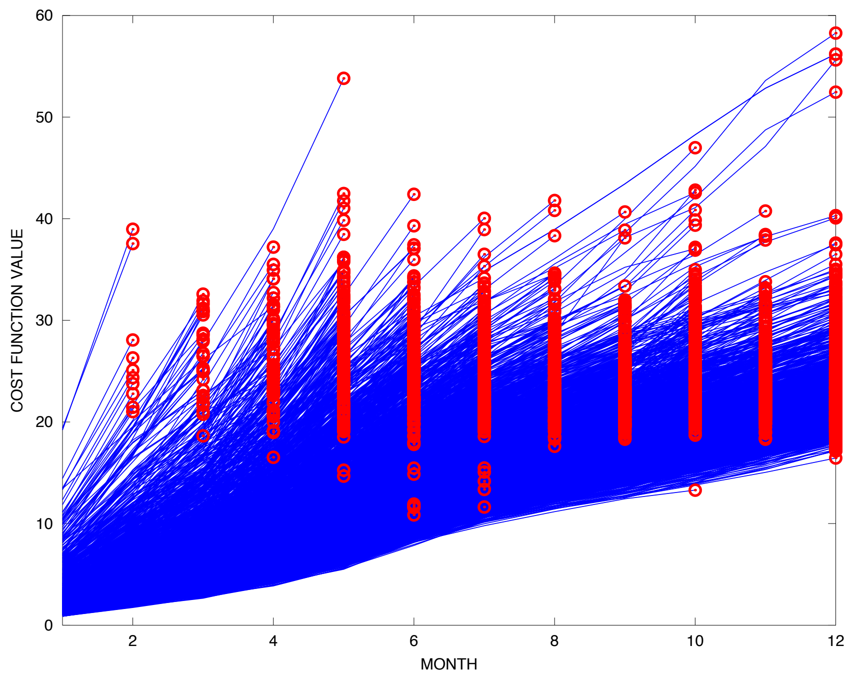


Figure 6: Examples of an early rejection in the initial value estimation case (top) and in the parameter estimation case (bottom). Blue line represents the truth from which observations (black dots) are generated. Red line is the model run until an early rejection. Changing the initial values causes large, sudden deviations from the reference data, and changing the parameters results in smaller and smoother changes.



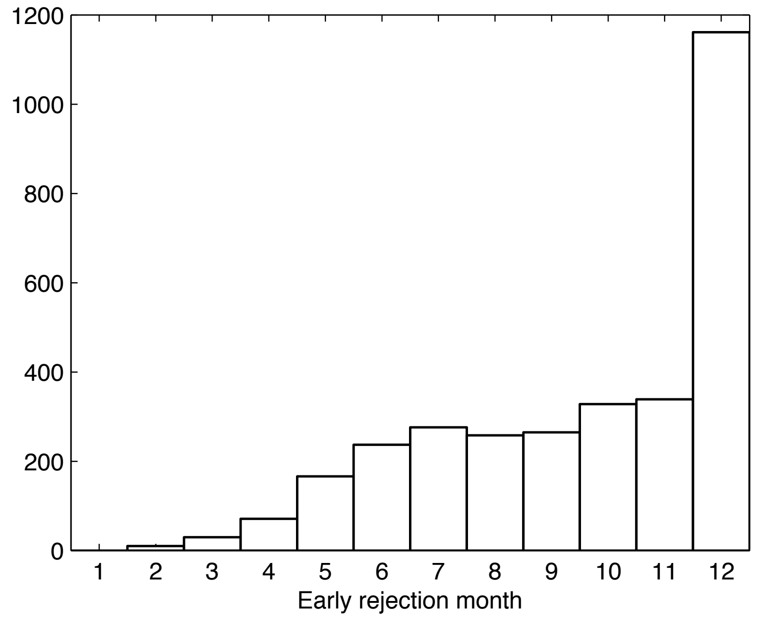


Figure 7: Top: examples of the evolution of the cost function (blue line) in the ECHAM5 climate model run until early rejection (red circle). Bottom: histogram of the months when early rejection happens, 12 months means that the proposed value was accepted.