Antti Solonen

BAYESIAN METHODS FOR ESTIMATION, OPTIMIZATION AND EXPERIMENTAL DESIGN

Thesis for the degree of Doctor of Science (Technology) to be presented with due permission for public examination and criticism in the Auditorium 1383 at Lappeenranta University of Technology, Lappeenranta, Finland on the 11th of November, 2011, at noon.

Acta Universitatis
Lappeenrantaensis 447
Preface

This work started in 2007 at the Department of Mathematics and Physics of the Lappeenranta University of Technology (LUT). After that, the postgraduate studies have taken me all the way to the U.S., and back to Finland, to Helsinki. For these opportunities, and for the numerous interesting research topics, acknowledgements go to my supervisor, Heikki Haario. Thank you, Heikki, for your guidance and support during the past 5 years.

I got my first touch to science in Lappeenranta, where we enjoy a fruitful collaboration with the Department of Chemical Technology. I want to thank especially Kari Vahteristo, Arto Laari and Markku Kuosa for the interesting applications, that eventually formed the first part of this thesis.

After a few years in Lappeenranta, I had the chance to go and study in the U.S., at the University of Montana. Thank you John Bardsley for inviting me, I will never forget my year in the Rocky Mountains. The work done during the visit plays an important role in this thesis: thank you Rebecca McCaffery and Elizabeth Crone, it was a pleasure working with you.

After coming back from the U.S., Johanna Tamminen arranged an office for me at the Finnish Meteorological Institute (FMI). Johanna, thank you for this great opportunity. I wish to thank also my other collaborators during my FMI years: Marko Laine, Heikki Järvinen, Janne Hakkarainen, Pirkka Ollinaho and Petri Räisänen from FMI, and Alexander Ilin and Erkki Oja from Aalto University. Special thanks to Marko, I have learned much from you.

Warm thanks also all my colleagues at LUT, especially Harri Auvinen and Tuomo Kau-ranne for scientific collaboration. And thank you ‘yläkerran pojat’ (Jere Heikkinen, Tapio Leppälampi and Jouni Sampo) for all the fun, see you on the next ‘pelipaivä’.

For making this work financially possible, I thank my department at LUT, the Center of Excellence in Inverse Problems, the Graduate School of Inverse Problems, the ASLA-Fulbright Program, the Finnish Graduate School of Computational Science (FICS) and the Computational Science Research Program of the Academy of Finland.

I thank my opponent Colin Fox, who agreed to fly across the world to argue with me in public. My gratitude goes also to the reviewers, Jukka Corander and John Bardsley. This thesis is better because of your efforts.

Finally, I wish to thank Maiju Kansanen, with whom I have been able to share both the successes and the difficulties during this long project.

Helsinki, October 2011

Antti Solonen
Mathematical models often contain parameters that need to be calibrated from measured data. The emergence of efficient Markov Chain Monte Carlo (MCMC) methods has made the Bayesian approach a standard tool in quantifying the uncertainty in the parameters. With MCMC, the parameter estimation problem can be solved in a fully statistical manner, and the whole distribution of the parameters can be explored, instead of obtaining point estimates and using, e.g., Gaussian approximations. In this thesis, MCMC methods are applied to parameter estimation problems in chemical reaction engineering, population ecology, and climate modeling. Motivated by the climate model experiments, the methods are developed further to make them more suitable for problems where the model is computationally intensive.

After the parameters are estimated, one can start to use the model for various tasks. Two such tasks are studied in this thesis: optimal design of experiments, where the task is to design the next measurements so that the parameter uncertainty is minimized, and model-based optimization, where a model-based quantity, such as the product yield in a chemical reaction model, is optimized. In this thesis, novel ways to perform these tasks are developed, based on the output of MCMC parameter estimation.

A separate topic is dynamical state estimation, where the task is to estimate the dynamically changing model state, instead of static parameters. For example, in numerical weather prediction, an estimate of the state of the atmosphere must constantly be updated based on the recently obtained measurements. In this thesis, a novel hybrid state estimation method is developed, which combines elements from deterministic and random sampling methods.

Keywords: Bayesian estimation, MCMC, stochastic optimization, design of experiments, state estimation, ensemble filtering

UDC 519.2 : 519.6 : 542.9 : 551.5 : 574
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>List of the original articles and the author’s contribution</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>Introduction</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>Parameter Estimation with MCMC</td>
<td>13</td>
</tr>
<tr>
<td>3.1</td>
<td>Monte Carlo Sampling Algorithms</td>
<td>14</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Metropolis Algorithm and Adaptive Variants</td>
<td>14</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Gibbs Sampling</td>
<td>15</td>
</tr>
<tr>
<td>3.2</td>
<td>Example: MCMC vs. Classical Methods</td>
<td>15</td>
</tr>
<tr>
<td>3.3</td>
<td>Case I: Chemical Reaction Kinetics</td>
<td>16</td>
</tr>
<tr>
<td>3.4</td>
<td>Case II: Mark-Recapture Analysis of a Frog Population</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>MCMC for Climate Models</td>
<td>25</td>
</tr>
<tr>
<td>4.1</td>
<td>Estimating ECHAM5 Closure Parameters with Adaptive MCMC</td>
<td>26</td>
</tr>
<tr>
<td>4.2</td>
<td>Improving MCMC Efficiency for Computationally Intensive Models</td>
<td>27</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Parallel Adaptive Chains</td>
<td>28</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Early Rejection</td>
<td>29</td>
</tr>
<tr>
<td>5</td>
<td>Optimal Experimental Design</td>
<td>33</td>
</tr>
<tr>
<td>5.1</td>
<td>Classical Optimal Design</td>
<td>33</td>
</tr>
<tr>
<td>5.2</td>
<td>Bayesian Optimal Design</td>
<td>34</td>
</tr>
<tr>
<td>5.3</td>
<td>Simulation-Based Design with Response Variances</td>
<td>35</td>
</tr>
<tr>
<td>5.4</td>
<td>Example: Chemical Kinetics</td>
<td>36</td>
</tr>
<tr>
<td>6</td>
<td>Model-Based Optimization</td>
<td>39</td>
</tr>
<tr>
<td>6.1</td>
<td>Direct Monte Carlo Sampling</td>
<td>40</td>
</tr>
<tr>
<td>6.2</td>
<td>MCMC Sampling and Simulated Annealing</td>
<td>40</td>
</tr>
<tr>
<td>6.3</td>
<td>Example: Simple Chemical Reaction</td>
<td>41</td>
</tr>
<tr>
<td>7</td>
<td>State Estimation: Variational Ensemble Kalman Filter</td>
<td>47</td>
</tr>
<tr>
<td>7.1</td>
<td>Kalman Filter and Extended Kalman Filter</td>
<td>49</td>
</tr>
<tr>
<td>7.2</td>
<td>Variational Kalman Filter</td>
<td>50</td>
</tr>
<tr>
<td>7.3</td>
<td>Ensemble Kalman Filter</td>
<td>51</td>
</tr>
<tr>
<td>7.4</td>
<td>Variational Ensemble Kalman Filter</td>
<td>51</td>
</tr>
<tr>
<td>7.5</td>
<td>Example: Lorenz 95</td>
<td>53</td>
</tr>
<tr>
<td>8</td>
<td>Conclusions</td>
<td>57</td>
</tr>
</tbody>
</table>
Bibliography

I  Kinetics of Neopentyl Glycol Esterification with Different Carboxylic Acids 65

II Frog population viability under past, present and future climate conditions: a Bayesian state-space approach 79

III Estimation of ECHAM5 climate model closure parameters with adaptive MCMC 101

IV Efficient MCMC for Climate Model Parameter Estimation: Parallel Adaptive Chains and Early Rejection 113

V Simulation-Based Optimal Design Using a Response Variance Criterion 131

VI Model-Based Process Optimization in the Presence of Parameter Uncertainty 153

VII Variational Ensemble Kalman Filtering Using Limited Memory BFGS 175
List of the original articles and the author’s contribution

This thesis consist of an introductory part and 7 scientific articles. The articles and the author’s contributions in them are summarized below.


VI Solonen A. and Haario H., 2011. Model-Based Process Optimization In the Presence of Parameter Uncertainty. Accepted for Publication in *Engineering Optimization*.


The author has computed most of the statistical analyses in paper I. The author has actively participated in the writing of the article, especially in the methods and results sections.
1. List of the original articles and the author’s contribution

The author is responsible for the mathematical part in the parameter estimation in paper II. The computer code for performing the analyses is written mostly by the author. The author has also participated in writing the paper.

In paper III, the author has participated in the discussions and in the writing of the article. In paper IV, the author is responsible for most of the writing and experimentation.

The ideas in paper V and VI are mainly developed by the author. All of the analyses are computed by the author and most of the writing process has been done by the author.

For paper VII, the author has implemented most of the algorithms and computed most of the results in the paper. The author has had a central role in developing the ideas used in the method. The author has been responsible for writing the article.
Mathematical modeling is a central tool in most fields of science and engineering. Mathematical models can be either ‘mechanistic’, which are based on principles of natural sciences, or ‘empirical’ where the phenomenon cannot be modeled exactly and the model is built by inferring relationships between variables directly from the available data.

Models are simplifications of reality, and therefore model predictions are always uncertain. Moreover, the obtained measurements, with which the models are calibrated, are often noisy. The task of statistical analysis of mathematical models is to quantify the uncertainty in the models. In this thesis, the focus is on uncertainty quantification of nonlinear mechanistic models.

In a typical modeling project, a mathematical model is formulated first and data is then collected to verify that the model is able to describe the phenomenon of interest. The models usually contain some unknown quantities (parameters) that need to be estimated from the measurements; the goal is to set the parameters so that the model explains the collected data well. The statistical analysis of the model happens at this ‘model fitting’ stage. Uncertainty in the data implies uncertainty in the parameter values.

Statistical methods are also needed when experiments are designed. Some measurements are more informative than others with respect to the parameters, and experimental design methods can be used to set the next measurements up so that they give as much information about the parameters as possible.

After enough data has been collected and the model has been successfully calibrated, one can start to use the model. The model can be used to simulate the phenomenon in different circumstances, e.g., as a part of larger systems, or to optimize processes that involve the phenomenon. To sum up, the following tasks and analyses are often present in a modeling project:

- Formulating the model \( s = f(x, \theta) \), where \( s \) is the model state, \( x \) are the design variables and \( \theta \) are the unknown parameters.

- Obtaining measurements \( y \) and mapping them to the model states with \( y = g(s) + \varepsilon \), where \( g \) is the observation function and \( \varepsilon \) represents the measurement error.
2. Introduction

- Model fitting: estimating parameters \( \theta \) when measurements \( y \) are given.
- Design of experiments: if more data is needed, setting the design variables \( x \) for the next measurements.
- Using the model for simulation studies or for optimizing a quantity that depends on the model fitting results, for example.

As mentioned, the statistical analysis happens at the model fitting stage. Traditionally in nonlinear model fitting, point estimates for the parameters are obtained, for example, by solving a least squares optimization problem. The uncertainty can be approximately obtained by linearizing the model at the point estimate and using results from linear normal theory, see e.g. (Bard 1974; Seber and Wild 1989). The result of the statistical analysis is typically given in a Gaussian form (as a covariance matrix).

Recently, the Bayesian framework for model fitting has become a popular approach for dealing with uncertainty in parameter estimation. In Bayesian model fitting, the parameters are considered as random variables, and the target for estimation is the distribution of the parameters instead of a point estimate. In the Bayesian approach, both the data and the prior knowledge of the parameters are modeled statistically, which gives a solid basis for the uncertainty analysis. For an introduction to Bayesian estimation, see (Gelman et al. 1996).

In practice, the reasons behind the popularity of the Bayesian approach are the fast developments in computing and the introduction of efficient numerical algorithms for carrying out the computations. Especially, the Markov Chain Monte Carlo (MCMC) sampling methods have made it possible to solve various nonlinear parameter estimation problems in a fully statistical manner, without performing, e.g., Gaussian approximations. In MCMC, the parameter distribution is approximated by producing a set of random samples from it. Thus, the answer to a given parameter estimation problem is given as a ‘chain’ of parameters instead of a single estimate.

In this thesis, Bayesian numerical methods for the last three items of the above list are studied. First, MCMC methods are applied to different parameter estimation problems in chemical engineering, population ecology and climate modeling. A special methodological focus is MCMC for computationally intensive models, motivated by issues encountered when applying MCMC to climate model parameter estimation. In addition to MCMC applications and methodological development, this thesis presents ways to incorporate the output of MCMC parameter estimation in two tasks that often follow model fitting: optimal design of experiments and optimization of model-dependent quantities.

In some cases, one is interested in inferring, instead of the static model parameters, the dynamically changing model states \( s_t \) at times \( t \) from measurements \( y_t \). These state estimation problems often have to be solved ‘on-line’ in real time. A typical example of a small dimensional problem is object tracking, where the aim is to estimate the position of an object in real-time using indirect measurements. A much larger scale example is numerical weather prediction, where the state of the weather is constantly updated using numerous measurements to allow model predictions. The theory and methods for small-scale problems are rather well established, but applying the methods to high-dimensional problems is difficult. In this thesis, a new hybrid method for high-dimensional cases is proposed, that combines components from random sampling and deterministic methods.
The introduction part of the thesis is organized as follows. In Chapter 3, the basics of Bayesian parameter estimation are discussed. Two case examples, that summarize the applications in papers I and II, are given. Chapter 4 concentrates on the specific topic of papers III and IV: applying MCMC to climate model parameter estimation. Using the output of Bayesian parameter estimation in optimal design of experiments and in model-based optimization is discussed in Chapters 5 and 6, summarizing papers V and VI. In Chapter 7, the topic of paper VII - high-dimensional dynamical state estimation - is summarized. Chapter 8 concludes the thesis.
2. Introduction
Let us consider the parameter estimation setting described in the previous chapter. In parameter estimation, parameters $\theta$ are estimated based on measurements $y$, traditionally using, e.g., a least squares approach. In Bayesian parameter estimation, $\theta$ is modeled as a random vector and the goal is to find the posterior distribution $\pi(\theta|y)$ of the parameters. The posterior distribution gives the probability density for values of $\theta$, given measurements $y$. Using the Bayes' formula, the posterior density can be written as

$$
\pi(\theta|y) = \frac{l(y|\theta)p(\theta)}{\int l(y|\theta)p(\theta) d\theta}.
$$

(3.1)

The likelihood $l(y|\theta)$ contains the measurement error model and gives the probability density of observing measurements $y$ given that the parameter value is $\theta$. For example, using the notation defined in the previous chapter and employing an additive Gaussian i.i.d. error model, $\epsilon \sim \mathcal{N}(0,\sigma^2\mathbf{I})$, gives likelihood

$$
l(y|\theta) \propto \prod_{i=1}^{n} l(y_i|\theta) \propto \exp \left( -\frac{1}{2\sigma^2} \sum_{i=1}^{n} [y_i - g(f(x_i,\theta))]^2 \right).
$$

(3.2)

The prior distribution $p(\theta)$ contains all existing information about the parameters, such as simple bounds and other constraints.

Different point estimates can be derived from the posterior distribution. The Maximum a Posteriori (MAP) estimate maximizes $\pi(\theta|y)$ and the Maximum Likelihood (ML) estimate maximizes $l(y|\theta)$. If the prior distribution is uniform within some bounds, ML and MAP coincide. With the Gaussian i.i.d. error assumption, ML coincides also with the classical Least Squares (LSQ) estimate, since minimizing the sum of squares term $SS(\theta) = \sum_{i=1}^{n} [y_i - g(f(x_i,\theta))]^2$ is equivalent to maximizing $l(y|\theta)$ in equation (3.2).

The real power of the Bayesian approach, however, stems from the numerical methods that allow the exploration of the whole posterior distribution, instead of only producing point estimates. With Markov Chain Monte Carlo (MCMC) methods, one can approximate the posterior distribution by producing samples from it. MCMC will produce samples of parameter values in proportion to their posterior density, i.e., many possible values for the parameters that match the prior and the likelihood.
In the next section, the numerical methods used in this thesis for producing random samples from the posterior distribution $\pi(\theta|y)$ are shortly discussed. After that, a simple example of Bayesian parameter estimation is given, and the difference between Bayesian estimation and classical nonlinear regression is discussed. Finally, the two case studies of papers I and II are discussed.

3. Monte Carlo Sampling Algorithms

Throughout this work, two different types of MCMC algorithms are used in parameter estimation. The work contains mainly parameter estimation with nonlinear mechanistic models, described as differential equation systems. For this purpose, the workhorses have been the adaptive variants of the Metropolis-algorithm, especially the Adaptive Metropolis (AM) algorithm, see (Haario et al. 2001), and the Delayed Rejection Adaptive Metropolis (DRAM) algorithm, see (Haario et al. 2006). For the mark-recapture data analysis (see Section 3.4), the availability of analytical expressions for conditional posteriors justifies the use of Gibbs sampling. In this Section, the basics of these sampling algorithms are briefly discussed.

3.1 Metropolis Algorithm and Adaptive Variants

One of the most widely used MCMC algorithms is the random walk Metropolis-Hastings (MH) algorithm introduced in (Metropolis et al. 1953). At MH step $i$, one randomly selects a candidate value $\theta_{i+1}$ from a proposal distribution $q(\cdot|\theta_i)$ that depends on the current point $\theta_i$ (usually a normal distribution centered at $\theta_i$). The proposed point is accepted with probability $\alpha = \min(1, \pi(\theta_{i+1}|y)/\pi(\theta_i|y))$. If the point is rejected, the current point $\theta_i$ is repeated. It can be shown that the resulting Markovian random walk can be regarded as samples from $\pi(\theta|y)$.

One of the recent trends to improve MCMC efficiency has been the introduction of adaptive samplers, initiated by the Adaptive Metropolis (AM) algorithm (Haario et al. 2001). In adaptive MCMC, one uses the sample history to tune the proposal distribution $q$ ‘on-line’ as the sampling proceeds. In AM, one calculates the empirical covariance from the samples obtained so far and uses that as the covariance of a Gaussian proposal. That is, new candidates are proposed as $\theta_{i+1} \sim N(\theta_i, \Sigma_i)$, where $\Sigma_i = \text{Cov}(\theta_1,\ldots,\theta_i) + \epsilon I$. In the original formulation of AM, the ‘regularization’ term $\epsilon I$ was added to make sure that the covariance stays positive definite and that the method has correct ergodicity properties, see (Vihola 2011) for recent results about the issue.

AM adaptation can be used in connection with other proposal schemes. The delayed rejection (DR) method by (Mira 2001) can be combined with AM, as done in (Haario et al. 2006). This DRAM method has been shown to be efficient in many applications. In DR, when a proposed candidate point in a Metropolis-Hastings chain is rejected, a second stage move is proposed around the current point. The process of delaying rejection can be iterated for a fixed or random number of stages and the higher stage proposals are allowed to depend on the candidates so far proposed and rejected. In the DRAM implementation used here, downscaled versions of the proposals given by AM adaptation are employed. This is especially helpful to get the sampler moving (get accepted points) in the beginning of the MCMC run. See (Haario et al. 2006) for DRAM details.

In this thesis, AM and DRAM methods are used in all sampling tasks in papers I, III, IV, V and VI. For a more detailed introduction of adaptive MCMC algorithms, see (Laine 2008).
3.2 Example: MCMC vs. Classical Methods

3.1.2 Gibbs Sampling

The idea of Gibbs sampling is very simple: instead of sampling directly from the full posterior \( \pi(\theta | y) \), one sweeps through conditional posteriors. If the parameter vector is divided into \( K \) groups \( \theta = (\theta_1, \theta_2, \ldots, \theta_K) \), the Gibbs sampler proceeds by sampling in turn from conditional posteriors \( \pi(\theta_i | \theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_K) \), where \( i = 1, \ldots, K \). In the extreme (yet typical) case, \( K = \text{dim}(\theta) \) and one samples from one-dimensional conditional distributions.

Gibbs sampling can be useful in high-dimensional problems, since the dimension of the random sampling is reduced. However, one has to perform \( K \) likelihood evaluations per sweep, which is problematic if likelihood evaluation is computationally expensive. In some cases there is a clear structure in the parameters, and some conditional distributions may be available analytically. One can also perform, for example, Metropolis steps for the conditional distributions, which is called Metropolis-within-Gibbs (MwG) sampling. The proposal adaptation ideas can be combined with component-wise sampling, see e.g. the single component adaptive Metropolis algorithm (SCAM) of (Haario et al. 2005).

Gibbs samplers naturally emerge for example in hierarchical Bayesian models and when so called conjugate priors are employed, see (Gelman et al. 1996). Conjugate priors are often used for computational convenience; they have the property that multiplying them with the likelihood results in a posterior that is of the same form as the prior. A useful example of this is the estimation of the measurement error variance \( \sigma^2 \) in basic model fitting cases, as in equation (3.2). Traditionally, a fixed value for \( \sigma^2 \) is used, estimated separately from repeated measurements or residuals, for example. The Bayesian way is to include \( \sigma^2 \) in the estimation problem and specify a prior \( p(\sigma^2) \) for it. Specifying a suitable conjugate prior (inverse gamma) for the variance, the posterior \( p(\sigma^2 | y, \theta) \) given parameter values \( \theta \) is available in closed form. Thus, in nonlinear model fitting cases, after applying a MH step to sample \( \theta \), one can sample new \( \sigma^2 \) for the next MCMC step directly from an inverse Gamma distribution. See (Laine 2008) for details about sampling \( \sigma^2 \).

In this thesis, Gibbs sampling is used, in addition to estimating the error variance \( \sigma^2 \), in the mark-recapture data analysis of paper II. In the mark-recapture case, conditional posteriors are all available analytically, which makes Gibbs sampling a natural approach.

3.2 Example: MCMC vs. Classical Methods

In this Section, the difference between classical nonlinear regression analysis and the Bayesian approach is illustrated using a simple toy example. The model \( y = \theta_1 (1 - \exp(-\theta_2 x)) \) is fitted to data \( y = (0.076, 0.258, 0.369, 0.492, 0.559) \) obtained at time points \( x = (1, 3, 5, 7, 9) \). First, the parameters are estimated using the classical least squares approach, and the covariance of the parameters is approximated using a linearization around the least squares estimate \( \hat{\theta} \). Linear normal theory gives the covariance for the estimate as \( \text{Cov}(\hat{\theta}) \approx \sigma^2 (J^T J)^{-1} \), where \( J \) is the Jacobian matrix of the parameters, \( J = \partial f(x, \theta) / \partial \theta \), evaluated at \( \hat{\theta} \), and \( \sigma^2 \) is the measurement error variance (assumed here to be i.i.d. Gaussian). The measurement error variance was estimated from the residuals, giving \( \sigma = 0.014 \).

Then, the DRAM sampler is run for 10000 steps to get samples from the posterior distribution. In Figure 3.1, the parameter distributions obtained with the classical linearization-based approach and with the MCMC approach are presented. One can see how the classical approach
can be misleading, if the likelihood is not well approximated by a Gaussian distribution. In the right-hand figure, the distribution of the model predictions beyond the measurement region is calculated from the MCMC output, illustrating that extending the uncertainty analysis to predictions (and other functions of the parameters) can be easily achieved, simply by simulating the model with different parameter values given by MCMC.

Figure 3.1: Left: parameter posterior with MCMC (blue) and linearization around the LSQ estimate (red). Right: predictive distributions with MCMC and the single prediction produced by the LSQ estimate. Gray colors correspond to 50%, 80%, 95% and 99% confidence envelopes.

3.3 Case I: Chemical Reaction Kinetics

In this Section, the model fitting case of paper I is summarized. The paper is chosen as an example from a series of MCMC studies in chemical reaction engineering, that have proceeded as follows. In (Kuosa et al. 2007), the parameters of an ozonation reaction, written as a partial differential equation system, were studied using MCMC. In (Kuosa et al. 2009), the study was extended to a more general reaction scheme. In (Vahteristo et al. 2008), MCMC was used, in addition to quantifying uncertainties in model parameters and predictions, to analyze the reliability of model-based optimization results, when the task was to set the temperature profile for the reaction so that the concentration of the desired product is maximized. In (Vahteristo et al. 2010), MCMC was used to compare two alternative reaction schemes. In paper I that is discussed here, MCMC results were extended to comparisons of different reaction rates.

The paper studies the reaction kinetics of neopentyl glycol (NPG) esterification with three different carboxylic acids. Such reactions are important, for example, in producing latex paints (see the original paper). The kinetic model is formulated as an ordinary differential equation (ODE) system, and the parameters that define the reaction rates are analyzed using MCMC. The results of the MCMC analysis are further used to estimate the distribution of model predictions and different functions of the reaction rate parameters.
3.3 Case I: Chemical Reaction Kinetics

**Kinetic Model.** The general esterification reaction of glycols with carboxylic acids can be written as

\[
\text{Glycol + Acid} \leftrightarrow \text{Monoester + Diester} \\
\text{Monoester + Diester} \leftrightarrow \text{Diester + Water} \\
2\text{Monoester} \leftrightarrow \text{Glycol + Diester}.
\]

In the specific esterification reactions studied here, all reaction components were observed. The glycol studied was neopentyl glycol (NPG) and the three acids compared were isobutyric acid (IBA), propionic acid (PRO) and 2-ethylhexanoic acid (EHA). The corresponding monoesters are denoted by MEIBA, MEPRO and MEEHA and diesters by DEIBA, DEPRO and DEEHA. Every reaction produced also water (W), which was continuously removed from the reactor.

After some assumptions and manipulations (see the original paper), the ODE model for the esterification reaction is written as

\[
\begin{align*}
\dot{C}_\text{NPG} &= -k_1C_{\text{NPG}}C_{\text{ACID}} - k_3C_{\text{DE}}C_{\text{NPG}} + k_{-3}C_{\text{ME}}^2 - \frac{C_{\text{NPG}}}{m_{\text{mix}}} \frac{dm_{\text{mix}}}{dt} \\
\dot{C}_\text{ACID} &= k_1C_{\text{NPG}}C_{\text{ACID}} - k_2C_{\text{ME}}C_{\text{ACID}} - \frac{C_{\text{ACID}}}{m_{\text{mix}}} \frac{dm_{\text{mix}}}{dt} \\
\dot{C}_\text{ME} &= k_1C_{\text{NPG}}C_{\text{ACID}} - k_3C_{\text{ME}}C_{\text{ACID}} + 2k_1C_{\text{DE}}C_{\text{NPG}} - 2k_{-3}C_{\text{ME}}^2 - \frac{C_{\text{ME}}}{m_{\text{mix}}} \frac{dm_{\text{mix}}}{dt} \\
\dot{C}_\text{DE} &= k_2C_{\text{ME}}C_{\text{ACID}} - k_3C_{\text{DE}}C_{\text{NPG}} + k_{-3}C_{\text{ME}}^2 \\
\dot{m}_{\text{mix}} &= -(k_1C_{\text{NPG}}C_{\text{ACID}} + k_2C_{\text{ME}}C_{\text{ACID}})m_{\text{mix}}M_W,
\end{align*}
\]

where \(C_{\text{ACID}}\) is the concentration of the acid (PRO, IBA or EHA) and \(C_{\text{ME}}\) and \(C_{\text{DE}}\) refer to the corresponding monoester (MEPRO, MEIBA, MEEHA) and diester (DEPRO, DEIBA, DEEHA). The total weight of the reaction mixture is denoted by \(m_{\text{mix}}\).

The temperature dependency of the reaction rates \(k_i\) are expressed by the Arrhenius law

\[
k_i(T) = A_i \exp \left( -\frac{E_i}{R} \left( \frac{1}{T} - \frac{1}{T_{\text{mean}}} \right) \right).
\]

The equilibrium constant for the last reaction is defined as \(K_3 = k_3/k_{-3}\). The vector of parameters that are estimated from data is \(\theta = [A_1, A_2, A_3, E_1, E_2, E_3, K_3]\).

**MCMC Analysis.** Altogether 32 experiments were carried out at different temperatures with different carboxylic acids and different catalysts, see the original paper for a detailed description of the experiments. For parameter estimation, the DRAM algorithm was run for 50000 steps. As an example, the model fit with IBA as the acid at temperature 383K is plotted in Figure 3.2. The posterior distribution of the parameters is given for the same case in Figure 3.3. In this case, the parameters are rather well identified and the distribution is approximately Gaussian, centered at the least squares estimate. The identifiability is worse when EHA is used as the acid, as can be seen from Figure 3.4: the disproportionation reaction parameters cannot be properly estimated due to the lack of experiments.
Figure 3.2: Predictive distribution (95% confidence envelopes) for the model (dark gray) and for the observations (light gray). The figure is taken from paper I.

Figure 3.3: 2D and 1D marginal posterior distribution plot for the parameters when esterification is done with IBA. The black lines represent the 95% and 50% confidence regions and one-dimensional density estimates. The figure is taken from paper I.
3.3 Case I: Chemical Reaction Kinetics

The analysis was extended to the reaction rate constants of equation (3.3). The purpose was to compare the reaction rates with different acids. Of particular interest was the ratio $k_1/k_2$ of the monoester formation and the consecutive esterification of monoester to diester. In Figure 3.5, the ratios and their confidence limits given by MCMC are illustrated. With the results, one can perform the reaction rate comparisons in a statistical manner.

All in all, MCMC was successfully used to study the uncertainty and identifiability of the model parameters. In addition, the confidence limits for the model predictions and the ratios of the apparent reaction rate constants could be easily produced using the MCMC output.
3.4 Case II: Mark-Recapture Analysis of a Frog Population

In this Section, the parameter estimation case of paper II is summarized. In the paper, the vital rate parameters (survival and growth) of a frog population in Montana, USA, are estimated from mark-recapture data. In mark-recapture studies, individuals are captured, marked, released and recaptured. The data consists of individual capture histories. Instead of a mechanistic model, as in the chemical reaction kinetics example in Section 3.3, a probabilistic model is used here. The life of a frog is divided into four stages: juvenile, subadult, adult female and adult male. The estimated parameters are the recapture, survival and transition probabilities. The following notation is used:

- \( x_{kt} \in \{0, 1, 2, 3, 4\} \) observed life stage for individual \( k \) at year \( t \) (0: unobserved)
- \( z_{kt} \in \{1, 2, 3, 4\} \) true life stage for individual \( k \) at year \( t \)
- \( S_{it} \in [0, 1] \) survival probability from year \( t \) to year \( t+1 \) for life stage \( i \)
- \( R_{it} \in [0, 1] \) recapture probability in year \( t \) for life stage \( i \)
- \( P_{ijt} \in [0, 1] \) transition probability from stage \( j \) at year \( t \) to stage \( i \) at year \( t+1 \).

Each row in the data matrix \( x \) includes an observed capture history for an individual. In the parameter estimation step, the Bayesian framework developed in (Clark et al. 2005) is used. Data was gathered during 10 summers, following the robust sampling design of (Pollock 1982). Thus, frogs were captured each summer in multiple consecutive days. Parameters were estimated across years, and across days it was assumed that the population was closed to immigration, emigration, births, and deaths. The data consists of ten-year capture histories of 4362 individual frogs. For details about the data, see (McCaffery and Maxell 2010).

Parameter Estimation

Survival, transition and recapture probabilities are estimated separately for each year and each life stage. The recapture probabilities can be directly estimated from the observations. The secondary sampling sessions within one summer are used to estimate the recapture probability for that year. The likelihood for observing \( x \) for a given year \( t \) with recapture probabilities \( R \) is

\[
p(x_t | R_t) \propto \prod_{i=1}^{4} R_{oi}^{o_i} (1 - R_{ui})^{u_i}, \tag{3.4}
\]

where the indicator variables \( o_i \) and \( u_i \) indicate the number of observed and unobserved individuals at year \( t \) and life stage \( i \), respectively. For estimating the survival and transition probabilities, an estimate of the true life stage \( z \) is needed. The likelihoods for life stages \( z \) for a given year \( t \) with given survival and transition probabilities \( S \) and \( P \) are

\[
p(z_t | S_t) \propto \prod_{i=1}^{4} S_{oi}^{o_i} (1 - S_{ui})^{u_i}, \tag{3.5}
\]

\[
p(z_t | P_{ij}) \propto \prod_{i=1}^{4} \prod_{j=1}^{4} P_{ij}^{#(i \rightarrow j)}, \tag{3.6}
\]
where indicator variables \( s_{it}, d_{it} \) and \( #(i \rightarrow j) \) indicate the number of survivals, deaths and transitions at year \( t \), respectively.

For estimating the true life stages of unobserved individuals, the stage at year \( t \) is conditioned on the previous and next year. The probability of being in a certain stage \( j \) is calculated as the product of probabilities for surviving previous year, transitioning to stage \( j \), not being captured, surviving to next year and transitioning to the next stage:

\[
p(z_{it} = j | z_{i,t-1} = m, z_{i,t+1} = l, R, S, P) \propto S_{m,t-1}P_{jm,t-1}(1 - R_{jt})S_{jt}P_{ljt}.
\] (3.7)

The total likelihood for observing data \( x \) and having true life stages \( z \), given the vital rate parameters, is obtained by taking a product of equations (3.4-3.6) over years. When a flat prior is used for the parameters, the recapture and survival probabilities are Beta distributed and the transition probabilities have Dirichlet distributions. Thus, given the observed data \( x \) and an estimate for life stages \( z \), one can sample new values for \( R, S \) and \( P \) from these known distributions. In contrast, for some given values of the vital rates, one can calculate the probabilities for different \( z_{it} \) using equation (3.7) and sample new values from the resulting multinomial distribution. Alternating these conditional samplings, the following Gibbs sampler is obtained:

i. Initialize: set initial guesses for \( R, S, P \) and \( z \) and set iteration counter \( i \rightarrow 1 \)

ii. Given \( x \) and \( z \), sample \( R, S \) and \( P \) from their conditional posteriors defined by equations (3.4) - (3.6)

iii. Loop over individuals \( i \) and time points \( t \) where \( x_{it} = 0 \), sample a new life stage \( z_{it} \) from equation (3.7)

iv. Set \( i \rightarrow i + 1 \) and go to step ii, until a desired number of iterations is obtained.

To clarify the parameter estimation setting, the data, the parameters and the notation are illustrated in Figure 3.6. Note that one might want to set an informative prior for \( R, S \) and \( P \), instead of the uniform priors used here. Convenient choices are conjugate priors (Beta and Dirichlet in this case), with which the posteriors are still available in closed form, as mentioned in Section 3.1.2.

**Forecasting Population Viability**

The estimation results were further used to study correlations between the vital rates (survival and growth) and certain climate variables that describe winter severity. The mean snowpack (snow-water equivalent, SWE) was found to correlate with survival and transition, see Figure 3.7. The Bayesian inference could be easily extended to the regression study by fitting the linear models separately for the possible vital rate values found by MCMC.

The relationship between winter severity and the vital rates were further used to make long term predictions of what happens to the population under different climate scenarios. The
Figure 3.6: An example of a row in the data matrix (an individual frog) $x_i$ is given, together with one possible true life stage vector $z_i$. The recapture probabilities are estimated within years directly from $x$, but survival and transition probabilities are estimated between years from $z$. In addition to sampling the vital rate parameters, the unknown (unobserved) components of $z$ (red) are varied as well.

Figure 3.7: Annual posterior estimates (with 95% credible intervals) for juvenile, subadult, and adult female survival in relation to SWE. The line shows predicted values for survival for different values of SWE, with 50% (dark gray) and 95% (light gray) confidence envelopes.
3.4 Case II: Mark-Recapture Analysis of a Frog Population

stochastic matrix \( M_t \) for simulating the development of the population from year \( t \) to year \( t + 1 \) was written as

\[
M_t = \begin{pmatrix}
0 & 0 & b_t C_t S_{2t} & b_t C_t S_{3t} \\
S_{0t} & P_{11t} S_{1t} & 0 & 0 \\
0 & P_{21t} S_{1t} & P_{22t} S_{2t} & 0 \\
0 & P_{31t} S_{1t} & P_{32t} S_{2t} & S_{3t}
\end{pmatrix}, \quad (3.8)
\]

where \( P_{ijt} \) and \( S_{it} \) are the vital rates described above. In the matrix, the columns represents pre-juvenile frogs, juvenile frogs, sub-adults and adult females, respectively. The survival probability from the pre-juvenile stage to the juvenile stage at year \( t \) is denoted by \( S_{0t} \) and \( b_t \) is the breeding probability for year \( t \). Clutch size at year \( t \) is denoted by \( C_t \). These parameters were estimated from separate data sets, see the original paper for details.

The matrix model was used to predict the long term development of the frog population, given different possible future scenarios for SWE. The effect of SWE was studied by varying the mean and the variance for the future SWE. Since the Bayesian sampling approach produces vital rate estimates as a set of samples, it is easy to use the samples to estimate the uncertainty of any function of the vital rates and extend the Bayesian analysis to all calculations made with the estimates. In this study, this property was utilized in all of the computations (regressions with vital rates and snowpack and matrix model predictions). In Figure 3.8, different population viability metrics are plotted with different values for mean and variance of future SWE. One can see that increasing both mean and variance of future SWE decreases population viability, but the effect of the mean is stronger. Note that the used viability metrics are based on the obtained samples for the vital rate parameters, and can only be calculated using the Bayesian parameter estimation approach.

In conclusion, the studied population is expected to benefit from the decrease in snow pack, likely brought along by climate change. Again, the Bayesian MCMC sampling approach was useful to propagate the uncertainty through all computations.
Figure 3.8: Contour plots showing the effect of changes in mean SWE (x-axis) and standard deviation of SWE (y-axis) on three different population viability metrics: (a) mean of the leading eigenvalue of $M_t$, (b) proportion of simulations where the population is growing and (c) percent of simulations in which the population went extinct. The figure is taken from paper II.
In the first chapters of this thesis, MCMC was applied to quantify uncertainty in both mechanistic and probabilistic models. However, the likelihoods in the discussed models have been simple enough that making thousands of evaluations is not an issue. If the model evaluation is computationally demanding, however, the time required to run sufficiently many MCMC steps can be prohibitively long. One such case, and the motivation behind this chapter, is the estimation of certain parameters in climate models.

Climate models are used to make long term predictions and to assess the effects of climate change, for example. However, climate models, like all models, are approximative, and the future predictions are uncertain. Quantifying this uncertainty is crucial to be able to assess the reliability of climate model predictions. Some statistics can be obtained by comparing predictions made with different climate models, see (Palmer et al. 2004). However, detailed analysis of the sources of uncertainty must be carried out to rigorously analyze the reliability of individual climate models.

Climate models are basically Navier-Stokes systems. To solve the partial differential equations, the governing equations are expressed in a discrete form using a computational grid. However, many important processes, such as cloud formation, operate in much smaller scales than the grid interval. These processes are included in the models using parameterizations that utilize variables which are resolved in the grid. For example, the amount of clouds and precipitation in a grid cell can be estimated using a parameterization based on the temperature and humidity in the cell. The parametric representations lead to situations where some free parameters necessarily appear. Predefining these ‘closure parameters’ leads to parametric uncertainty in the models.

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

Recently, attempts have been made to develop objective, algorithmic approaches for closure parameter estimation and uncertainty analysis, see e.g. (Jackson et al. 2008). In paper III, MCMC sampling was used for the first time for this purpose with a full climate model (the
ECHAM5 climate model). While it was shown that MCMC is a viable option for closure parameter estimation, the applied model resolution was low and running the MCMC experiments took a very long time (1–3 months). Thus, various ways to speed up MCMC are needed to make parameter estimation studies practically attainable. In paper IV, two ways to improve the efficiency of MCMC are presented. Note that while the motivation for paper IV was the climate model closure parameter estimation problem, the presented algorithms apply rather generally to problems where the model evaluation is computationally demanding. Moreover, the closure parameter problem itself is not limited to climate models: the problem is generic for multi-scale modeling.

In this chapter, the closure parameter estimation approach of paper III is summarized first. Then, the methodological developments of paper IV are discussed.

4.1 Estimating ECHAM5 Closure Parameters with Adaptive MCMC

In the MCMC study, version 5.4 of the ECHAM5 atmospheric general circulation model (Roeckner et al. 2003; Roeckner et al. 2006) was used. Four ECHAM5 closure parameters were considered, all related to physical parameterizations of clouds and precipitation.

As mentioned, climate model parameters are currently defined rather subjectively. A human expert is efficient in distinguishing ‘good’ and ‘bad’ parameter values. The central goal of this research is to construct a ‘cost function’ (likelihood) which would measure the validity of a climate simulation and replace this human element. The parameter distribution is conditional on the choice of this function.

The cost functions tested in paper III were related to the net radiative flux at the top of the atmosphere, see the original papers for details. Net radiation was chosen as the first cost function candidate, since it is often used as a criterion in non-algorithmic model tuning. The radiation needs to be in balance to keep the simulated climate sensible, and net radiation therefore is a natural constraint that can be used to distinguish realistic parameter values from unrealistic ones. Four ECHAM5 closure parameters were chosen for the study, all related to physical parameterizations of clouds and precipitation. The parameters affect the model cloud fields and therefore potentially also the radiative fluxes used in the objective function.

Here, the cost function that involves yearly global and monthly zonal net radiation averages is considered. The target distribution is \( \pi(\theta) \propto \exp(-J(\theta)/2) \), where the cost function \( J(\theta) \) is written as

\[
J(\theta) = \frac{(F - F^o)^2}{\sigma^2} + \sum_{t=1}^{12} \sum_{y=1}^{32} \frac{w_y (F_{t,y} - F_{t,y}^o)^2}{\sigma_{t,y}^2},
\]

where \( F \) is the annual mean flux and \( F_{t,y} \) is the mean flux for month \( t \) and zonal band \( y \). The observations are denoted by superscript \( o \). The weights \( w_y \) represent area fractions for the zonal bands, and the normalizing factors are the standard deviations of the inter-annual variability in monthly and zonal mean net fluxes. The observational estimates are taken from the Clouds and the Earth’s Radiant Energy System (CERES) Energy Balanced and Filled (EBAF) dataset (Loeb et al. 2009). The inter-annual standard deviations are derived from the European Centre for Medium-Range Weather Forecasts (ECMWF) reanalysis data (ERA-40; Uppala et al. 2005).
4.2 Improving MCMC Efficiency for Computationally Intensive Models

An MCMC chain consisting of 4500 model runs was performed, using the DRAM algorithm. Each model evaluation represents a one-year climate simulation with the low-resolution ECHAM5 model. Figure 4.1 displays posterior distributions of the four parameters computed from runs 1001 through 4500. The MCMC algorithm is able to provide some constraints on all four parameters considered, but the cost function considered is not enough to identify all parameters. Ongoing research attempts to define a cost function that can uniquely determine the parameters, using, for example, data mining techniques to find climate characteristics that are most sensitive to changes in the parameters.

4.2 Improving MCMC Efficiency for Computationally Intensive Models

MCMC methods typically require thousands of model evaluations, and it is challenging to apply them to problems where the model evaluation is computationally time consuming. One such case is the parameter estimation problem in climate models discussed above.

In a review paper on parameter estimation in climate modeling (Annan and Hargreaves 2007), MCMC was considered too computationally expensive for the task. In (Villagran et al. 2008), modern adaptive MCMC methods performed well with surrogate climate models. In paper III, it is shown that efficient adaptive MCMC can indeed be used to estimate the parameter distribution of a full climate model. However, the MCMC run with the required chain length took a long time to compute, even when the lowest realistic model resolution was used. For
extensive MCMC experimentation and higher model resolution, ways to speed up MCMC are needed. In paper IV, two techniques are presented for CPU intensive models: parallel adaptive MCMC and early rejection. These are discussed next.

### 4.2.1 Parallel Adaptive Chains

Here, a parallel chain version of the AM algorithm is presented. To parallelize AM, a simple mechanism called inter-chain adaptation is used, recently introduced in (Craiu et al. 2009). In inter-chain adaptation, one simply uses the samples generated by all parallel chains to perform proposal adaptation and the resulting proposal is used for all 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 straightforward to implement. The only difference to running independent parallel AM samplers is that each sampler applies and updates the same proposal covariance. The parallel chain implementation with AM adaptation is given as a flowchart in Figure 4.2. Note that also other adaptation schemes, such as the DRAM and SCAM methods discussed in Chapter 3, can be combined with inter-chain adaptation.

The parallel chain algorithm with DRAM adaptation (Haario et al. 2006) was implemented.
for the ECHAM5 climate model closure parameter estimation problem, see paper IV for details. The MCMC chain obtained with 7 parallel chains and a single chain are compared in Figure 4.3. One can clearly see the improvement in the ‘exploration speed’, especially for two out of the four parameters. From the practical point of view, the parallel algorithm has been crucial in our further MCMC experiments in terms of obtaining the results in a more reasonable time.

![Figure 4.3: MCMC chains from experiments with a single chain (red) and with 7 parallel chains (black). The figure is taken from paper IV.](image)

4.2.2 Early Rejection

In traditional MH based sampling, one first evaluates the whole model with a proposed parameter value and compares its posterior density to the density of the previous point. The idea in ER is simply to monitor the likelihood as the model simulation proceeds and stop the simulation as soon as one can conclude that the proposed parameter will result in a rejection.

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

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 $k$ measurements
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 \( k \). In this case, one can reject as soon as \( \frac{\pi_k(\theta_{i+1})}{\pi_k(\theta_i)} < r \). Thus, one can speed up the sampling simply by switching the order of the calculations: generate the random number \( r \) first, evaluate the likelihood part by part and check after each part if the parameter can be rejected.

The amount of calculation saved by ER depends on the problem and on the tuning of the proposal. In cases where the topology of the parameter posterior distribution is complicated (e.g. strongly nonlinear), the MH sampler, even if properly tuned, results in low acceptance rates and potentially large performance gains can be achieved through ER. Below, ER is demonstrated with a simple (almost Gaussian) posterior and a difficult, 'banana-shaped' posterior. Both cases are produced using the toy model of Section 3.2 by varying the time interval from which the measurements are obtained, see the original paper for details. In Figure 4.4, the parameter posteriors and tuned proposal covariances are illustrated for both cases. The histograms of measurement indices, where early rejection happens, are also given. In the strongly nonlinear case, many proposed parameter values miss the region of high probability and can be rejected early. In this case, ER saved roughly 50% of the computations. In the easier case, the proposal is a better approximation of the target and rejections can be typically done only in the end of the run, resulting in about 15% savings.

Thus, ER savings are not too significant with 'easy', approximatively Gaussian targets. However, one can argue that the whole MCMC approach may then not be really needed, since roughly the same results can be obtained using classical linear theory; the motive of MCMC is to be able sample 'difficult' posteriors, that otherwise are intractable. Loosely speaking, the more difficult the posterior is, the more helpful ER is. In the examples of paper IV, ER saved 10%-80% of CPU time.
4.2 Improving MCMC Efficiency for Computationally Intensive Models

Figure 4.4: Left: parameter distributions and tuned proposal covariances (95% ellipses) 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). The figure is taken from paper IV.
4. MCMC for Climate Models
After the posterior distribution of the parameters is obtained via MCMC, it is natural to ask how the results can be used further in other statistical analyses. One potential use is the design of the next experiments. In optimal experimental design, the goal is to use the current information about the parameters to guide the experimentation so that the parameter uncertainty is reduced as much as possible. The practical appeal of optimal design methods is that the model parameters can be estimated with fewer experiments, reducing the costs of experimentation.

In more practical terms, when measurements are obtained for parameter estimation, one has to set the design variables $x$ with which the measurements are done. For example, in chemical engineering this could mean setting the temperature and the pressure for the next experiment. Some measurements are more informative than others with respect to the parameters, which motivates the optimal experimental design problem: which values for the control variables give most information about the parameters? To be more precise, the task in optimal design is, after measuring $y$ and estimating $\theta$, to find a new experimental setting $x$ so that after measuring at $x$, the uncertainty in $\theta$ is minimized.

The optimal design theory is well established for linear models with additive Gaussian errors. For nonlinear models, the traditional approach is to linearize the model around a point estimate and use the linear theory. However, the classical approach deals with parameter covariance matrices and it is therefore not compatible with the output given by Bayesian estimation and MCMC. In this work, the problem of optimal experimental design, given a set of samples from the posterior distribution $\pi(\theta|y)$, is studied. The approach builds upon the concept of simulation-based optimal design, originally introduced in (Muller et al. 2004). Below, classical optimal design, Bayesian optimal design and the main ideas behind the proposed approach are summarized. For a detailed description, refer to paper V.

5.1 Classical Optimal Design

Let us start from the linear optimal design theory. In the case of linear models $y = D\theta + \varepsilon$ with Gaussian errors, the distribution of the least squares estimator is analytically known. More specifically, if $\varepsilon \sim N(0, \sigma^2 I)$, the least squares solution is $\hat{\theta} = (D^T D)^{-1} D^T y$, and we...
have
\[ \text{Cov}(\hat{\theta}) = \sigma^2 (D^T D)^{-1}. \]  
\[ (5.1) \]

Designing new measurements means adding new rows \( X \) to the existing design matrix \( D \), which gives the covariance
\[ C(X) \propto (D^T D + X^T X)^{-1}. \]  
\[ (5.2) \]

The classical ‘alphabetical’ design criteria minimize \( C(X) \) (or maximize its inverse) in different ways, based on the eigenvalues \( (\lambda_1, ..., \lambda_n) \) of the covariance matrix. For example, the D-optimal design, given by
\[ x^*_D = \arg\min_X \left( \prod_{i=1}^{n} \lambda_i \right) = \arg\min_X (\det(C(X))), \]  
\[ (5.3) \]
minimizes the volume of the confidence ellipsoid at \( \hat{\theta} \). An introduction to classical design criteria is given in (Atkinson et al. 2007).

With nonlinear models \( y = f(x, \theta) \) one can obtain a covariance estimate by linearizing the model around \( \hat{\theta} \): the matrices \( D \) and \( X \) are replaced by Jacobians \( J = \partial f(x, \theta)/\partial \theta \) evaluated at \( \theta = \hat{\theta} \). The linear approximation and the dependency on \( \hat{\theta} \) are major problems in the classical design approach. Moreover, if the parameter estimation problem is ill-posed in the sense that the covariance becomes singular, the classical criteria cannot be used. Thus, one can design measurements with the classical criteria only after the parameters can already be reasonably well estimated from data, and one has to assume that a good approximation of the true value of \( \theta \) is known. Next, the Bayesian optimal design is discussed, that can, in principle, solve these problems.

5.2 Bayesian Optimal Design

An introduction to a general Bayesian experimental design framework is given for example in (Chaloner and Verdinelli 1995). Bayesian optimal design is based on defining a utility function \( u(x, y, \theta) \) that is a measure of the utility of obtaining measurement \( y \) at point \( x \) with parameter value \( \theta \). Let us assume that we have already some information about model parameters, coded into a density \( p(\theta) \). Let \( p(y|\theta, x) \) denote the likelihood for observing a new measurement \( y \) at \( x \), given that the parameter value is \( \theta \). Bayesian optimal design solves the \( \hat{\theta} \) dependency problem by averaging over parameters \( \theta \) and possible measurements \( y \). The expected utility \( U(x) \) that one wishes to maximize w.r.t. \( x \) is defined as
\[ U(x) = \int \int p(y, \theta|x)u(x, y, \theta)dyd\theta = \int \int p(\theta)p(y|\theta, x)u(x, y, \theta)dyd\theta, \]  
\[ (5.4) \]
where \( p(y, \theta|x) = p(\theta)p(y|\theta, x) \) is the joint distribution of the parameters and new measurements. The simplest way to approximate the above integral over the parameter and observation spaces is to use direct Monte Carlo integration; sampling \( (\theta_i, y_i) \sim p(\theta, y|x), i = 1, ..., n \) and calculating
\[ \hat{U}(x) = \frac{1}{n} \sum_{i=1}^{n} u(x, \theta_i, y_i). \]  
\[ (5.5) \]
Naturally, the future measurements $y$ are unknown, but new measurements can be synthetically generated using the model and the current knowledge about the parameters. Parameters $\theta_i$ can be taken from the density $p(\theta)$, which can be created beforehand with MCMC, for example. Measurements $y_i$ can be simulated using the model, which is the reason why this approach is called simulation-based design.

The direct Monte Carlo integration is computationally expensive. In (Muller et al. 2004), the integration is done using MCMC by sampling from target

$$\pi(x, \theta, y) \propto p(y, \theta|x) u(x, \theta) = p(\theta)p(y|\theta, x)u(x, y, \theta)$$

(5.6)

which clearly admits $U(x)$ as its marginal. Thus, one can sample triples $(x, \theta, y)$ and read $U(x)$ from the resulting chain. Again, we often have $p(\theta)$ available as an MCMC chain and $y$ can be produced using the model.

Optimal design is traditionally viewed as an optimization problem. However, additional information about the sensitivity of the design problem can be obtained by running MCMC across designs and by looking at the whole design surface $U(x)$. For optimization, it is noted in (Muller et al. 2004) that the simulation-based MCMC sampling approach can be turned into an optimization algorithm by sampling from the target

$$\pi_J(x, \theta_1, ..., \theta_J, y_1, ..., y_J) = \prod_{i=1}^{J} p(\theta_i)p(y_i|\theta_i, x)u(x, y_i, \theta_i),$$

(5.7)

where $\theta_{1, J} = (\theta_1, ..., \theta_J)$ and $y_{1, J} = (y_1, ..., y_J)$ are $J$ identically distributed parameter and measurement vectors. The above target admits $U(x)^J$ as its marginal. Thus, increasing $J$ has an annealing effect on the target: the samples will be more concentrated around the modes of the target. In practice, one needs to draw $J$ random parameters from $p(\theta)$ and $J$ new measurements from $p(y|\theta, x)$ at each MCMC step.

In this work, the simulation-based design framework is used and developed further. In more detail, a general purpose utility function, based on calculating variances of the model response, is proposed, together with a way to carry out the computations efficiently.

### 5.3 Simulation-Based Design with Response Variances

How should one define the utility function using an existing MCMC chain from $p(\theta)$? An intuitive idea is that a measurement helps the most in experimental settings where the model response contains the largest uncertainty, given the current knowledge about $\theta$. In practice this amounts to searching for regions of high response variance. Thus, the utility for a single measurement is defined here simply as

$$U(x) = \text{Var}_{p(\theta)}(f(x, \cdot)) \approx \frac{1}{N_\theta-1} \sum_{i=1}^{N_\theta} (f(x, \theta_i) - \bar{f}(x, \cdot))^2$$

(5.8)

where $\theta_i, i = 1, ..., N_\theta$ are samples from the chain and $\bar{f}(x, \cdot)$ is the sample mean of the calculated model predictions. The sample statistics can be estimated simply by simulating with the model using different possible parameter values given in the MCMC chain of $p(\theta)$. 
It turns out that, in the linear case, maximizing the response variance is equivalent to classical D-optimal design, when the task is to design only one new measurement, see paper V for justification. In most real design problems, however, one has to design a whole series of measurements at once. How can the response variance criterion be extended to this setting? A simple sum of the utilities (5.8) at different design points is not enough, since the utility of a single measurement conditionally depends on the other measurements.

In the proposed design criterion, the parameter distribution $p(\theta)$ is updated after each single measurement. The actual measurements are, of course, unknown at the design stage, but one can simulate measurements with the model. The response variance criterion for our 'simulated sequential design' approach is written as

$$u(x, y, \theta) = \prod_{i=1}^n (1 + \sigma^{-2}\text{Var}_{p(\theta|y_{1:k-1})}(f(x_i, \theta))),$$

(5.9)

where the notation $y_{1:k} = (y_1, ..., y_k)$ means the first $k$ single measurements, $\sigma^2$ is the measurement error variance, and $\text{Var}_{p(\theta|y_{1:k})}(f(x, \theta))$ denotes the response variance after $k$ measurements (for $k = 0$, $p(\theta|y_{1:k}) = p(\theta)$ denotes the current posterior). The derivation of the criterion follows the connection between response variances and classical D-optimal design. The posterior updating is done using an importance weighting mechanism. For details about derivation and computation, see paper V.

The response variance design criteria can be computed even if the parameter estimation problem is ill-posed (Jacobians are singular). One can, for example, start only with upper and lower bounds for the parameters and start design optimization from the very first measurements. Classical design criteria can be used only after there is enough data to get non-singular Jacobians. Starting the optimal design 'early' can have a significant effect in the number of experiments needed to estimate the parameters reliably. This is demonstrated in the next example.

5.4 Example: Chemical Kinetics

This example is from (Berger et al. 2000), a benchmark project for estimation and design in chemical reaction kinetics. The rate $r$ of the reaction

$$\text{CO} + 2\text{H}_2 \leftrightarrow \text{MeOH}$$

(5.10)

is modelled as a function of temperature $T$ and partial pressures of the reactants, $p_C$, $p_H$ and $p_M$:

$$r = \frac{k_{\text{ref}} \exp\left(-\frac{E}{RT}\right) (p_Cp_H^2 - p_M/K_{eq})}{(1 + K_1p_C + K_3 \exp\left(-\frac{\Delta H_3}{RT}\right) + K_4p_H/p_C)p_Cp_H}$$

(5.11)

In this example, the unknown parameters are $\theta = [k_{\text{ref}}, E, K_1, K_3, K_4, \Delta H_3]$ and the experimental conditions $x = [T, p_C, p_H, p_M]$. In the original example, 27 measurements that follow a classical factorial design were given, and the task was to design the 28th measurement. Here experiments are designed sequentially, starting the optimization from the first measurement. Only a flat prior is assumed by giving lower and upper bounds for the parameters, obtained
from (Berger et al. 2000). For the design optimization, the annealing approach (equation 5.7) was used with the response variance criterion (equation 5.9), see the original paper for more details about the experiment.

As an example, the effect of the 5th measurement is illustrated in Figure 5.1. It is clear that the Jacobians are singular when the first experiments are designed, and classical criteria are unavailable.

After 8 optimized measurements the parameters are, roughly, as well identified as with the original 27 measurements given in (Berger et al. 2000). In Figure 5.2, the parameter posteriors given by 8 optimized measurements, by the 27 original measurements and by 8 measurements that follow a classical fractional factorial design are compared. The proposed method produces the smallest uncertainty with only 8 measurements.

![Figure 5.1: Two-dimensional marginals of the parameter posterior before (red) and after (black) the 5th measurement. The figure is taken from paper V.](image)

This example shows that starting the design optimization early can result in dramatic reduction in the number of experiments needed to estimate the parameters. Designing measurements in ill-posed problems becomes available through the presented sampling approach.
Figure 5.2: Comparison of posteriors with iteratively optimized 8 measurements (black), original 27 measurements (red) and 8 measurements that follow the classical factorial design (green). The figure is taken from paper V.
After collecting measurements and fitting the model, one can start using the model to study the phenomenon of interest. One common type of study, in addition to the experimental design problem discussed in the previous chapter, is model-based optimization. In model-based optimization, one searches for control variables that optimize a certain criterion using the model, for example maximize a product yield in a chemical process model. More formally, one wishes to optimize some criterion $c(x, \theta)$ with respect to control variables $x$. For example, in chemical reaction engineering $x$ could be the temperature of a reaction, $\theta$ might represent the reaction rate constants, and $c(x, \theta)$ could be the yield of a product given by the model with parameter value $\theta$.

The problem in the model-based approach is that one often does not know $\theta$ exactly (estimated from incomplete and noisy measurements). Traditionally, one plugs in a specific point estimate $\hat{\theta}$, optimizes $c(x, \hat{\theta})$ and ignores the uncertainty related to $\theta$. Recently, ways to incorporate $\theta$ in the optimization as a distribution have been proposed, but they consider only specific forms for the parameter distribution; see, e.g., (Ma and Braatz 2003; Rooney and Biegler 2003) and the review paper (Lee and Chen 2009). This chapter summarizes paper VI, which studies how parameter uncertainty, given in the form of a set of samples from the parameter posterior obtained by MCMC, can be incorporated into model-based process optimization.

In the Bayesian parameter estimation framework, $\theta$ is modeled as a random variable that has probability density $p(\theta|y)$, from which one can produce samples using e.g. MCMC. Following this interpretation, also the optimization criterion $c(x, \theta)$ is a distribution that can obtain a range of possible values at any point $x$. Instead of optimizing $c(x, \theta)$ for a specific fixed $\theta$, the Bayesian approach allows one to optimize a function of the distribution $c(x, \theta)$, for example maximize the expected product yield or the smallest possible product yield.

Below, some ‘distribution-based’ optimization criteria, and two ways to evaluate them, are introduced. First, the obvious (but computationally costly) direct Monte Carlo approach is discussed, where the distribution of $c(x, \theta)$ is directly simulated with different parameter samples given by MCMC parameter estimation. Then, a way to use MCMC integration and simulated annealing to optimize similar criteria is presented, based on the optimal design algorithm of paper V. Finally, an example case is presented, which shows that taking uncertainty into account in the optimization can have a large effect in the optimization results.
6.1 Direct Monte Carlo Sampling

An obvious candidate criterion for optimization is the expectation: one averages $c(x, \theta)$ over $\theta$ and maximizes the average $C(x)$, where

$$C(x) = \mathbb{E}_{p(\theta|y)}[c(x, \theta)] = \int c(x, \theta)p(\theta|y)d\theta. \tag{6.1}$$

This integral over the parameter space can be approximated using the existing MCMC samples from $p(\theta|y)$. The most obvious way is to pick a (large) number of samples $(\theta_1, \ldots, \theta_N)$ from the MCMC output and use a direct Monte Carlo approximation:

$$C(x) \approx \frac{1}{N} \sum_{i=1}^{N} c(x, \theta_i). \tag{6.2}$$

However, using the average alone as an optimization criterion does not take into account the variability in the criterion. It might be useful to solve a multi-objective optimization task instead, where the expectation is maximized and the variability is minimized. A simple way to implement this is to penalize large standard deviations in the objective function:

$$C(x) = \mathbb{E}_{p(\theta|y)}[c(x, \theta)] - \alpha \text{Std}_{p(\theta|y)}[c(x, \theta)], \tag{6.3}$$

where the tuning parameter $\alpha$ defines the weight given for the variability in the criterion. This robust mean criterion can also be computed easily using the direct Monte Carlo approximation using empirical mean and standard deviation formulas.

One can also optimize the ’worst case criterion’, where the worst possible value of $c(x, \theta)$ is maximized:

$$C(x) = \min_{\theta} c(x, \theta). \tag{6.4}$$

In this case, direct Monte Carlo approximation means calculating $c(x, \theta)$ with different possible values for $\theta$ and finding the minimum from the calculated samples.

The direct Monte Carlo approximation of the above criteria is simple to implement and include in different optimization routines. However, the approach is computationally challenging, since one might need a large number of points $\theta_i$ to evaluate $C(x)$ just once.

6.2 MCMC Sampling and Simulated Annealing

The idea used in paper VI, originally presented in (Muller et al. 2004) in an optimal design of experiments context, is to use MCMC sampling for approximating $C(x)$. For the mean criterion, this can be done by running MCMC with target density

$$\pi(x, \theta) \propto c(x, \theta)p(\theta|y), \tag{6.5}$$

assuming that $c(x, \theta)$ is non-negative and bounded for all $x$ and $\theta$. The above target density admits $C(x)$ as its marginal. The MCMC sampler is run in the joint space of $x$ and $\theta$, where parameters $\theta$ are proposed directly from the parameter posterior distribution and controls $x$ from a separate proposal distribution.
Thus, in addition to using MCMC to produce samples from $p(\theta|y)$, MCMC sampling is used to explore the ‘x-space’ and approximate $C(x)$ as well. This might seem contradictory, since we are dealing with an optimization problem instead of a sampling problem. However, the MCMC sampler can be turned into an optimization algorithm using the same simulated annealing technique as in the optimal design approach of paper VI. In this approach, one samples from an augmented target

$$\pi(x, \theta_1, ..., \theta_J) = \prod_{i=1}^J c(x, \theta_i)p(\theta_i|y)$$

that has the marginal distribution $C(x)^J$. In this case, one picks $(\theta_1, ..., \theta_J)$ at each MCMC iteration randomly from the existing parameter chain. Increasing $J$ concentrates the samples more tightly around the regions of high $C(x)$.

The MCMC approach can be used only if the criterion $C(x)$ can be written as a convenient integral, as the expectation above. For instance, the robust mean criterion in equation (6.3) cannot be expressed in such a way, and the MCMC sampling approach cannot be directly applied. However, a similar effect of penalizing large deviations in $c(x, \theta)$ can be obtained by defining the criterion as

$$C(x) = E_{p(\theta|y)}[c(x, \theta)] - \alpha E_{p(\theta, \theta'|y)}[|c(x, \theta) - c(x, \theta')|]$$

where $p(\theta, \theta'|y) \propto p(\theta|y)p(\theta'|y)$ is the joint distribution of two independent and identically distributed parameter vectors. It is straightforward to write this as an integral, which can be approximated by MCMC sampling from target

$$\pi(x, \theta, \theta') = (c(x, \theta) - \alpha |c(x, \theta) - c(x, \theta')|) p(\theta|y)p(\theta'|y).$$

Thus, at each MCMC iteration, one picks two independent random samples $\theta$ and $\theta'$ from the parameter distribution. Naturally, the annealed target, see equation (6.6) for expectation, can be used with the robust mean criterion as well.

The worst-case criterion in equation (6.4) cannot be written in an integral form. For this type of criteria, one has to use the direct Monte Carlo approximation.

### 6.3 Example: Simple Chemical Reaction

As an example, consider a simple chemical reaction $A \rightarrow B \rightarrow C$. Compound $A$ transforms into $B$ with rate $k_1(T)$ and $B$ into $C$ with rate $k_2(T)$. The reaction rates depend on temperature $T$. The model is written as an ODE system as

$$\frac{dA}{dt} = -k_1(T)A$$
$$\frac{dB}{dt} = k_1(T)A - k_2(T)B$$
$$\frac{dC}{dt} = k_2(T)B.$$

The temperature dependency of the reaction rates is defined using the Arrhenius equation

$$k_i(T) = a_i \exp \left( -\frac{E_i}{R} \left( \frac{1}{T} - \frac{1}{T_0} \right) \right)$$

(6.9)
where $T_0$ is a reference temperature and $R$ is the gas constant. In parameter estimation, one estimates $\theta = (a_1, E_1, a_2, E_2)$ by measuring $A$ and $B$. In Figure 6.1, the MCMC parameter estimation results are illustrated. Refer to the original article for parameter estimation details.

The optimization task here is to calculate the sampling time and the temperature that maximize the concentration of the intermediate product $B$. Thus, the control variables are $x = (t, T)$ and the criterion function is defined as

$$c(x, \theta) = B(t, T, \theta), \quad (6.10)$$

where $B(t, T, \theta)$ denotes $B$-component of the solution of the above ODE at time $t$ with temperature $T$ and parameter values $\theta$.

Since there are only two control variables in this case and the model is simple, one can demonstrate different optimization criteria by calculating direct Monte Carlo approximations on a 2D-grid of different values for the two control variables. The optimal process given by a point estimate (MAP estimate) is compared to the optimum given by the robust mean criterion in Figure 6.2.

Using the MAP estimate only in the optimization suggests that in order to get maximal product yield, the temperature should be set high and the product should be collected soon after the reaction starts. The optimum given by the robust mean criterion is different: it is better to put the temperature to a lower value and to wait for a bit longer before collecting the product. To illustrate the difference of these solutions, the distributions of model responses when using the point estimate only and when using the robust mean criterion are compared in Figure 6.3. The optimum value obtained in both ways is roughly the same, but there is more uncertainty in the solution given by the point estimate only. The distribution of the maximal product yields with different values for $\alpha$ are given in Figure 6.4.

From the MCMC output, one can study the sensitivity of the process optimum found with different methods, as in Figure 6.4, but the analysis does not tell how one method compares
6.3 Example: Simple Chemical Reaction

Figure 6.2: Criterion surfaces with the MAP estimate (left) and with the robust mean criterion with α = 0.5. Temperatures are in Celcius. The figure is modified from paper VI.

Figure 6.3: Distribution of model response at the MAP-optimal conditions (left) and at the optimal temperature calculated with the robust mean criterion (right). Two confidence envelopes, 50% (dark gray) and 99% (light gray) are plotted. The figure is taken from paper VI.
Figure 6.4: Distributions of product yield at MAP-optimal conditions (blue) and with the robust mean criterion (red and green). The figure is taken from paper VI.

to another in real life. In synthetic cases, however, the ‘true’ behavior of the system is known, and one can compare how the MCMC approach performs compared to the classical approach, where a fixed point estimate is used and the parameter uncertainty is neglected.

Here, such an experiment is performed as follows. First, synthetic data is generated, by adding noise to the model simulated with ‘true’ parameter values. Then, parameters are estimated using both least squares and MCMC, and the optimization is performed using the obtained estimates. Thus, in the least squares approach the obtained point estimate is fixed at the optimization stage, and in the MCMC approach the parameter uncertainty is taken into account. Finally, the obtained optimization results are plugged in to the true model (model with assumed true parameters) to see how large product yields were actually obtained with the two methods. The above procedure is repeated a number of times (here 400) to get statistics for the performance comparison.

The results of the comparison are illustrated in Figure 6.5. For the 400 test cases, the proposed approach that takes parameter uncertainty into account gives robust results, whereas using the point estimate only can lead to poor product yield. For instance, the product yield was below 60 in 33/400 cases when the point estimate was used, but only in 1/400 cases when the MCMC approach was applied. The standard optimization approach includes a high risk of a bad process design, which can be avoided by the proposed approach.

Note that taking the parameters into account as a distribution can yield different solutions for the process optimization problem, even if the parameter uncertainty is relatively small, as in this toy example. In real situations, one often cannot estimate parameters as well as in this example. The more uncertainty there is in parameter estimates, the more important it is to consider the optimized quantity as a distribution instead of a point estimate.
Figure 6.5: The optimal product yield (red) and the actual product yield obtained using the MCMC approach (green) and the classical way (blue) for 400 cases. The figure is taken from paper VI.
6. Model-Based Optimization
Besides model parameter estimation, where the estimation setting is static, one is often interested in estimating the dynamically changing state of the system. In many problems, the state of the system is not known and can be observed only partially. As time proceeds, new measurements become available, that can be used to update the state estimate. The state estimate and its uncertainty can be evolved in time using a mathematical model of the system. An example of a large state estimation problem is numerical weather prediction (NWP), where a weather model is used to propagate the state in time and observations are constantly used to update the state. The dynamical state estimation problems usually contain a real-time requirement: the current system state has to be updated fast enough to enable future predictions. In geosciences, dynamical state estimation is called data assimilation.

Here the state estimation problem is formulated as follows, using the notation of paper VII. At discrete times $k$, estimate the system state $x_k$ from observations $y_k$, when the model and observation equations are given as

$$x_k = M(x_{k-1}) + \varepsilon^p_k$$  \hspace{1cm} (7.1)$$

$$y_k = K(x_k) + \varepsilon^o_k.$$  \hspace{1cm} (7.2)

In the above system, $M$ is the ‘evolution model’ that evolves the state in time and $K$ is the observation model that maps the state to the observations. Error terms $\varepsilon^p_k$ and $\varepsilon^o_k$ represent the model error and the observation error, respectively.

State estimation problems are dynamical, where measurements are obtained in real-time and the state estimate needs to be updated after the measurements are obtained. This can be achieved by applying the Bayes’ formula sequentially. The prior is given by evolving the posterior of the model state from the previous time step using the model $M$ (prediction step). The obtained prior is updated with the likelihood of the measurement (update step) to get the posterior, which is evolved with the model and used as the prior in the next time step. Repeating this procedure allows ‘on-line’ estimation of model states.

In general terms, the sequential state estimation, also known as filtering, works as follows. The filtering methods aim at estimating the marginal distribution of the states $p(x_k|y_{1:k})$ given the measurements obtained until the current time. In the prediction step, the whole
distribution of the state is moved with the dynamical model to the next time step:

\[ p(x_k | y_{1:k-1}) = \int p(x_k | x_{k-1}) p(x_{k-1} | y_{1:k-1}) \, dx_{k-1}. \]  \tag{7.3} 

When the new observation \( y_k \) is obtained, the model state is updated using the Bayes’ rule, with (7.3) as the prior:

\[ p(x_k | y_{1:k}) \propto p(y_k | x_k) p(x_k | y_{1:k-1}). \]  \tag{7.4} 

This posterior is used inside integral (7.3) in the next prediction step. The term \( p(x_k | x_{k-1}) \) includes the evolution model and describes the probability of having state \( x_k \) at time \( k \), given that the state was \( x_{k-1} \) at the previous time step. The idea of sequential state estimation is illustrated in Figure 7.1. For more details about general filtering formulas, see (Särkkä 2006).

\[ y_{k-1} = K(x_{k-1}) + \epsilon_{k-1} \]

\[ y_k = K(x_k) + \epsilon_k \]

**Figure 7.1**: Two iterations of sequential state estimation at times \( k-1 \) and \( k \). The prior is given by the model, which is combined with the observation to get the posterior. The posterior is further propagated to be the prior for the next time point.

Different state estimation methods can be derived, depending on the assumptions of the form of the state distribution, the likelihood and the techniques used to evolve the uncertainty of the state in time. The fully Bayesian solution to the filtering problem is given by particle filters, also known as sequential Monte Carlo (SMC) methods. In particle filtering, the state estimate and its uncertainty is propagated in time simply by evolving samples (particles) from the state posterior with the nonlinear model \( M \). When new observations are obtained, the particles are resampled using importance sampling, where the weights for the particles are given by the likelihood. For an introduction to particle filters, see (Cappe et al. 2007).

The particle filtering methods run into difficulties in high-dimensional problems and in cases where the model evaluation is computationally costly. The sequential estimation problem can be made easier by assuming a specific form for the state distributions. For linear models and Gaussian likelihoods, the posterior of the state can be written down analytically. This approach leads to the Kalman filter (KF) and its nonlinear extension, the extended Kalman filter (EKF), see the next section for a short introduction. While KF and EKF work in much higher dimensional problems than particle filters, they are problematic when the dimension is very large, since they involve covariance matrices of size \( d \times d \) where \( d \) is the dimension of
7.1 Kalman Filter and Extended Kalman Filter

the state vector. Recently, a low-storage variational approach to approximate KF and EKF was proposed in (Auvinen et al. 2009), called the Variational Kalman Filter (VKF). In VKF, the large matrices in KF formulas were replaced with low-storage approximation provided by the quasi-Newton optimization method called limited memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS).

Ensemble Kalman filters (EnKF) - see (Evensen 2004) for an introduction - combine the particle filtering idea with the Gaussian assumption for the distribution of the states. In EnKF, an ensemble of states is propagated using the model. EnKF methods essentially replace the covariances in the KF formulas by the empirical covariance estimates calculated from the ensemble.

In this thesis, the focus is on high-dimensional state estimation problems. In paper VII, the VKF idea is generalized to ensemble filtering. Covariances are estimated using the LBFGS method and new ensembles can be efficiently generated from the posterior covariance approximation. All computations can be performed without explicitly constructing full covariance matrices. The method is summarized in Section 7.4, after recalling how KF, EKF, VKF and EnKF methods work.

7.1 Kalman Filter and Extended Kalman Filter

The Kalman filter and the extended Kalman filter (Kalman 1960) are among the most used state estimation methods. The KF is meant for cases where the errors are Gaussian and the model and observation operators are linear: 

\[ M(x_k) = M_k x_{k-1} \]  
\[ K(x_k) = K_k x_{k-1} \]

Since the model structure is linear and Gaussian in KF, the sequential application of the Bayes’ formula can be written down analytically. After some manipulation, the prediction and update steps of the KF can be written as follows.

i. Prediction: move the state estimate \( x_{k-1}^{est} \) and its covariance \( C_{k-1}^{est} \) in time

(a) Compute \( x_k^p = M_k x_{k-1}^{est} \).

(b) Compute \( C_k^p = M_k C_{k-1}^{est} M_k^T + C_{\epsilon k}^p \).

ii. Update: combine the prior \( x_k^p \) with observations \( y_k \)

(a) Compute the Kalman gain \( G_k = C_k^p K_k^T C_{k-1}^p + C_{\epsilon k}^p \)^{-1}.

(b) Compute the state estimate \( x_k^{est} = x_k^p + G_k(y_k - K_k x_k^p) \).

(c) Compute the covariance estimate \( C_k^{est} = C_k^p - G_k K_k C_k^p \).

The extended Kalman filter directly uses the Kalman filter formulas in the nonlinear case by replacing the nonlinear model and observation operators in the covariance computations with linearizations: 

\[ M_k = \partial M(x_{k-1}^{est})/\partial x \] and \[ K_k = \partial K(x_{k-1}^{est})/\partial x \].

In practice, the KF and EKF methods suffer from computational problems, when the dimension of the state space is large. First of all, as the dimension gets larger, the covariance matrices soon become too large to fit to the memory of a computer. Secondly, in nonlinear problems, linearizing the model can be difficult. Moreover, the method involves a matrix inversion that can be difficult if the number of observations is large.
7.2 Variational Kalman Filter

In the variational formulation of the Kalman filter, the state estimation at step \( k \) is viewed as an optimization problem, where the quadratic function (negative log-posterior)

\[
- \log p(x|y_k) = \frac{1}{2} (x - x^p_k)^T (C^p_k)^{-1} (x - x^p_k) + \frac{1}{2} (y_k - \mathcal{K}(x))^T (C^\varepsilon_k)^{-1} (y_k - \mathcal{K}(x))
\]  

(7.5)

is minimized with respect to \( x \). In the VKF method, introduced in (Auvinen et al. 2009), the minimization is done with the LBFGS optimization method, that produces both the state estimate and a low-storage approximation of the covariance (inverse Hessian at the minimizer).

In the VKF algorithm, the inverse of the prior covariance \( C^p_k \) is also approximated using LBFGS by setting up an auxiliary optimization problem (with a trivial solution), given as

\[
\arg \min_u \frac{1}{2} u^T C^p_k u.
\]  

(7.6)

Thus, the LBFGS optimization routine provides low-storage approximation for both \( (C^p_k)^{-1} \) and \( C^\text{est}_k \). All computations with the covariances can be carried out efficiently using the implicit low-storage representation, without forming the full matrices. The VKF method for nonlinear problems is given as an algorithm below:

i. **Prediction**: move the state estimate and covariance in time

(a) Compute \( x^p_k = \mathcal{M}(x^{\text{est}}_{k-1}) \).

(b) Define \( C^p_k = M_k B^p_{k-1} M^T_k + C^\varepsilon_k \).

(c) Apply LBFGS to (7.6) to get an approximation \( B^*_k \) of \( (C^p_k)^{-1} \).

ii. **Update**: combine the prior with observations

(a) Minimize expression (7.5) using LBFGS to get the state estimate \( x^{\text{est}}_k \) and covariance estimate \( B^*_k \).

iii. Set \( k \rightarrow k + 1 \) and go to step i.

Note that while VKF can solve the storage problem related to EKF, it requires a way to evolve the covariance in time (step 1(b) in the algorithm above). Propagating the covariance using a direct linearization, as in EKF, is infeasible in high dimensions. In VKF, covariance propagation is done using the so called tangent linear and adjoint codes, that implement differentiation in the ‘code level’: each line in the model code is differentiated separately, and the chain rule is applied to compose linearization from the individual derivatives. This is a standard technique in variational data assimilation, see e.g. (Le Dimet and Talagrand 1986) and (Giering and Kaminski 1998). These codes must be prepared separately for every model and their construction is laborious, although automatic code generators have been recently developed, see e.g. (Cheng et al. 2009). In ensemble filters, tangent linear and adjoint codes are not needed.


7.3 Ensemble Kalman Filter

In ensemble filtering, the uncertainty in the state estimate $x_k$ is represented as $N$ samples instead of a covariance matrix, here denoted as $s_k = (s_{k,1}, s_{k,2}, ..., s_{k,N})$. The first ensemble filtering method was the ensemble Kalman filter (EnKF), introduced in (Evensen 1994) and implemented in operational numerical weather prediction e.g. in (Houtekamer and Mitchell 1998). The ensemble Kalman filter essentially replaces the state covariance matrices in EKF with the sample covariance calculated from the ensemble. The sample covariance can be written as $\text{Cov}(s_k) = X_k^T X_k$, where

$$X_k = \left( (s_{k,1} - \bar{s}_k), (s_{k,2} - \bar{s}_k), ..., (s_{k,N} - \bar{s}_k) \right) / \sqrt{N - 1}.$$  \hspace{1cm} (7.7)

The sample mean is denoted by $\bar{s}_k$. Using this notation, the EnKF algorithm can be formulated as follows:

i. **Prediction:** move the state estimate and covariance in time

(a) Move ensemble forward and perturb members with model error: $s^p_{k,i} = M(s_{k-1,i}) + \epsilon^p_{k,i}$, $i = 1, ..., N$.

(b) Calculate sample mean $\bar{s}^p_k$ and covariance $C^p_k = X_k^T X_k$.

ii. **Update:** combine the prior with observations

(a) Compute the Kalman gain $G_k = C^p_k K^T_k (K_k C^p_k K^T_k + C^\epsilon_k)^{-1}$.

(b) Update ensemble members $s^\text{est}_{k,i} = s^p_{k,i} + G_k (y_k - K_k s^p_{k,i} + \epsilon^o_{k,i})$.

(c) Calculate state estimate as the sample mean: $x^\text{est}_k = \bar{s}^\text{est}_k$.

In the above algorithm, vectors $\epsilon^p_{k,i}$ and $\epsilon^o_{k,i}$ are realizations of the model and observation errors $\epsilon^p_k$ and $\epsilon^o_k$, respectively.

The ensemble Kalman Filter is very simple to implement and it does not require tangent linear and adjoint codes. However, EnKF has various problems and numerous variants have been developed to overcome these issues, see e.g. (Houtekamer and Mitchell 1998), (Zupanski 2005), (Anderson 2001), (Etherton and Bishop 2004). Most notably, EnKF suffers from errors brought by the random perturbation of model states and observations. The variational ensemble Kalman filter (VEnKF), discussed in the next Section, can overcome many issues in the standard EnKF.

7.4 Variational Ensemble Kalman Filter

In paper VII, a method called the Variational Ensemble Kalman Filter (VEnKF) is developed. VEnKF follows the VKF ideas and shows how they can be implemented in the ensemble filtering context. As in VKF, the state estimation in VEnKF is based on minimizing the cost function in equation (7.5). The prior covariance needed in the cost function is defined here as

$$C^p_k = \text{Cov} (M(x^\text{est}_{k-1}) + \epsilon^p_k) = \text{Cov} (M(x^\text{est}_{k-1})) + \text{Cov} (\epsilon^p_k) = X_k^T X_k + C^\epsilon_k.$$  \hspace{1cm} (7.8)
Note that the above formula contains the common assumption, that the model error and model response are uncorrelated. In VEnKF, the sample covariance $X_kX_k^T$ is calculated using the state estimate evolved from the previous time as the expectation, instead of the sample mean used in EnKF. Thus, we define

$$X_k = ((s_{k,1} - x_k^p), (s_{k,2} - x_k^p), ..., (s_{k,N} - x_k^p))/\sqrt{N},$$

(7.9)

where $x_k^p = M(x_{est,k} - 1)$ and $s_{k,i} = M(s_{est,k-1,i})$. Note that now the ensemble members do not contain random perturbations; the model error is added directly in equation (7.8), as in the standard KF.

The inverse of the prior covariance $C_p^k = X_kX_k^T + C_{\varepsilon}^k$, needed when evaluating the cost function (7.5), can be obtained in two ways. Following VKF derivation, one can approximate the inverse by applying LBFGS to the artificial optimization problem

$$\min_u \frac{1}{2} u^T (X_kX_k^T + C_{\varepsilon}^k) u.$$  

(7.10)

An attractive alternative way to proceed is to calculate the inverse of the prior covariance using the Sherman-Morrison-Woodbury (SMW) matrix inversion formula, see e.g. (Higham 2002).

The inverse of the prior covariance can be written as

$$\begin{aligned}
(C_p^k)^{-1} &= (X_kX_k^T + C_{\varepsilon}^k)^{-1} \\
&= C_{\varepsilon}^{-1} - C_{\varepsilon}^{-1}X_k(I + X_k^T C_k^{-1} X_k)^{-1}X_k^T C_{\varepsilon}^{-1}.
\end{aligned}$$

(7.11)

(7.12)

This formulation of the inverse of the prior covariance can be directly inserted into equation (7.5) when it is minimized. With this formulation, one needs to invert only $N \times N$ matrices instead of $d \times d$. The computation of the quadratic expression $(x - x_k^p)^T (C_p^k)^{-1} (x - x_k^p)$ can be organized so that no full matrices of size $d \times d$ have to be stored. With this formulation, the first LBFGS approximation can be avoided and the prior can be included ‘exactly’ in the second optimization. Note that in VKF this is not possible, and the auxiliary optimization task (7.6) needs to be solved.

When LBFGS optimization is applied to minimize function (7.5), an estimate $x_{est}^k$ and a low-storage approximation for the covariance $C_{est}^k$ are obtained. After that, a new ensemble of state vectors from $N(x_{est}^k, C_{est}^k)$ is needed. Typically, sampling from a multivariate normal distribution is done via the Cholesky decomposition of the covariance matrix, which is impossible in this case, since one cannot even store the full matrix $C_{est}^k$ and it is available only in the LBFGS form. However, samples can be drawn directly from the LBFGS covariance representation in an efficient manner, since the LBFGS representation for $C_{est}^k$ can be written in the form

$$C_{est}^k = B_0B_0^T + \sum_{i=1}^n b_ib_i^T,$$

(7.13)

where $B_0$ is a $d \times d$ matrix and $b_i$ are $d \times 1$ vectors. From this representation one can produce a zero mean random vector $r \sim N(0, C_{est}^k)$ simply by calculating

$$r = B_0x + \sum_{i=1}^n \omega_i b_i,$$

(7.14)
7.5 Example: Lorenz 95

where \( z \sim N(0, I) \) and \( \omega_i \sim N(0, 1) \). The matrix \( B_0 \) does not have to be stored explicitly, since the product \( B_0z \) can be computed implicitly using the stored LBFGS vectors. See the appendix of paper VII for details about constructing \( B_0 \) and \( b_i \). This efficient sampling from low-memory LBFGS covariance approximations is potentially useful in many other contexts besides VEnKF, for example in quantifying the uncertainty in inverse problems, where the dimension usually grows large due to dense discretization.

Below, the VEnKF is presented as an algorithm. After setting the initial guesses for the state and its covariance to \( x^0_{\text{est}} \) and \( C^0_{\text{est}} \) respectively, and setting \( k = 1 \), the algorithm is written as follows:

i. **Prediction**: move the ensemble forward and build the prior:
   (a) Compute prior center point \( x^p_k = M(x^0_{\text{est}, k-1}) \).
   (b) Compute prior ensemble \( s^p_{k,i} = M(s^0_{\text{est}, (k-1), i}) \), \( i = 1, \ldots, N \).
   (c) Define \( (C^p_k)^{-1} \) using SMW formula (7.12)
      (Alternatively: Apply LBFGS to (7.10) to get \( (C^p_k)^{-1} \)).

ii. **Update**: calculate the posterior estimate and generate the new ensemble:
   (a) Apply LBFGS to minimize (7.5) to get \( x^\text{est}_k \) and \( C^\text{est}_k \).
   (b) Sample new ensemble \( s^\text{est}_k,i \sim N(x^\text{est}_k, C^\text{est}_k) \) using (7.14).

iii. Set \( k \rightarrow k + 1 \) and go to step i.

#### 7.5 Example: Lorenz 95

In this example, the well-known nonlinear and chaotic Lorenz’95 model, introduced in (Lorenz 1996) and further analyzed in (Lorenz 1998), is considered. The model shares many characteristics with realistic atmospheric models and it is often used as a low-order test case for weather forecasting schemes. A 40-dimensional version of the model is used, given as an ODE system as

\[
\frac{dx_i}{dt} = (x_{i+1} - x_{i-2})x_{i-1} - x_i + 8, \quad i = 1, 2, \ldots, 40. \tag{7.15}
\]

The state variables are periodic: \( x_{-1} = x_{39}, x_0 = x_40, \) and \( x_{41} = x_1 \). Out of the 40 model states, measurements are obtained from 24 states. The observation operator is defined (following (Auvinen et al. 2009)) as \( K(x) = Kx \), where

\[
K_{rp} = \begin{cases} 
1, & (r,p) \in \{(3j + i, 5j + i + 2)\} \\
0 & \text{otherwise}
\end{cases} \tag{7.16}
\]

where \( i = 1, 2, 3 \) and \( j = 0, 1, \ldots, 7 \). Thus, the last three states in every set of five are observed. To generate data, Gaussian noise to the model solution is added. For details of the experiment, see paper VII.
Figure 7.2: Comparison of EnKF (red), VEnKF (black) and EKF (green) with ensemble sizes $N = (10, 15, 20, 40)$ in the Lorenz 95 example. Increasing ensemble size leads to monotonically decreasing error levels for both EnKF and VEnKF. The figure is taken from paper VII.
Experiments are run with varying ensemble size $N$. In Figure 7.2, the performance of EKF, EnKF and VEnKF is compared with $N = (10, 15, 20, 40)$. From the results it is clear that VEnKF works better, when the ensemble size is small. When the ensemble size gets large, VEnKF and EnKF performances approach each other.

In the original paper, a high-dimensional linear problem was also studied. In that case, the difference between VEnKF and EnKF was even more dramatic, and the KF could not be run anymore due to memory issues. Thus, VEnKF seems to be a promising approach for high-dimensional ensemble filtering. Testing the performance of VEnKF in real problems remains a topic of further research.
When parameters of mathematical models are estimated from incomplete and noisy data, it is important to quantify the uncertainty in the estimates. The introduction of MCMC methods to carry out the computations has made Bayesian parameter estimation an increasingly popular tool for statistical analysis of mathematical models. MCMC methods give the solution to the parameter estimation problem as samples from the distribution of the model parameters. The uncertainty analysis can be easily extended to any function of the model parameters using Monte Carlo simulation. In this thesis, the flexibility of the MCMC sampling approach is demonstrated in a chemical reaction model and a frog population analysis.

MCMC methods are CPU intensive, and applying them to computationally demanding models requires special attention. In this thesis, MCMC was applied to the closure parameter estimation problem in climate modeling. The MCMC methodology was developed further to make the basic algorithms more efficient using parallel chain and early rejection techniques. With the presented improvements, MCMC methods can be applied more easily to larger models. In the climate model case, the proposed methods have been shown to be crucial in obtaining parameter estimation results in reasonable time.

In addition to applying and developing MCMC methodology, this thesis considers how the output of the estimation can be used as input in optimal experimental design. Classical experimental design methods are not compatible with the output of MCMC methods. The Bayesian simulation-based optimal design methodology, which utilizes MCMC output, is developed further in this thesis, showing that the approach can significantly reduce the number of experiments needed to obtain accurate parameter estimates.

In model-based optimization, one optimizes a certain quantity using the model, e.g., a product yield in a chemical reaction system. Traditionally, the model is fixed at the optimization stage by fixing a point estimate of model parameters. However, if the parameters are considered as a distribution, the optimized quantity is also a distribution, and a stochastic optimization task follows. In this thesis, potential ways to solve the problem are discussed. Test problems show that taking the uncertainty into account can have a large positive effect in the optimization results.

Instead of static model parameters, the quantity of interest is often the model state itself, which has to be estimated 'on-line' as the measurements become available. The theory and
methods are rather well-established for small scale problems. In this work, the problems arising in high-dimensional problems are considered. A novel hybrid state estimation method is presented, that combines components from deterministic variational methods and random sampling based ensemble methods. In test problems, the performance of the method is promising. The most attractive feature of the method is its easy application: it can be implemented using the forward model only, without the often laborious construction of linearization codes. Components of the proposed method, especially the efficient generation of random vectors from high-dimensional normal distributions, are potentially useful for static estimation problems as well.

To sum up, the thesis includes applications and methodological development of uncertainty quantification in both low and high-dimensional cases. Moreover, novel ways to use the output of the methods in other common analyses are presented. The developed methods are still prototypes, and future research will show how they perform in real-life applications.


62


Kinetics of Neopentyl Glycol Esterification with Different Carboxylic Acids


© 2009 American Chemical Society. Reprinted, with permission.
KINETICS, CATALYSIS, AND REACTION ENGINEERING

Kinetics of Neopentyl Glycol Esterification with Different Carboxylic Acids

Kari Vahteristo,* Sylvie Maury, Arto Laari, Antti Solonen, Heikki Haario, and Salme Koskimies†
Lappeenranta University of Technology, Department of Chemical Technology; P.O. Box 20, FIN-53851 Lappeenranta, Finland

The aim of this work was to study the reaction kinetics of neopentyl glycol esterification with different carboxylic acids (propionic acid, isobutyric acid and 2-ethylhexanoic acid). The experiments were carried out in a batch reactor as reactive distillation removing continuous one reaction product (water) by purging nitrogen through the reactor. Commercial heterogeneous gelular (Dowex 50WX2) and macroreticular (Amberlyst 15) type resins and homogeneous para-toluene sulfonic acid were compared as catalysts. The gelular type resin was dried before loading into the reactor. When performing esterification of neopentyl glycol with different acids the observed formation of monoester was clearly much faster than the consecutive esterification of monoester to diester. A disproportionation reaction was also found to take place. The reliability of the estimated Arrhenius parameters was analyzed using Bayesian inference and Markov chain Monte Carlo methods (MCMC). MCMC analysis was also used to study uncertainties in the model predictions and in some additional parameters, such as reaction rate constants calculated from the estimated parameters. This cannot be done by classical regression analysis.

1. Introduction

Latex paints have low solvent emissions, and they form films that have excellent physical and chemical properties.2,3 Film formation is a rather complicated process, and many of the latexes used in paints do not have the required desirable properties. This problem is normally solved by adding a nonvolatile plasticizer which unfortunately has the effect of making films soft and tacky causing dirt accumulation and blocking problems during drying. To overcome these problems a fugitive coalescing agent is added to the latex. The coalescing agent dissolves in the latex particles, acts as a plasticizer, and increases free-volume availability, hence permitting film formation at a lower temperature.1 After the film has formed, the coalescing agent diffuses slowly to the surface and evaporates. Generally, coalescing agents are mixtures of different components and not a pure compound.5

It has been reported that certain esters may function as coalescents.2 These esters (from 8 to 14 carbon atoms in the molecule) are hydrolytically stable, and they are particularly useful in alkaline coating formulations. Suitable esters can be produced by reacting different glycols with different carboxylic acids. These reversible, consecutive second-order esterification reactions have been widely researched.4–7 For example, an esterification reaction of ethylene glycol with acetic acid has been used as a model reaction for reactive distillation.8 Generally, reaction equations for these esterification mechanisms can be written as

\[
\text{glycol} + \text{acid} \rightleftharpoons \text{monoester} + \text{water} \quad (R1)
\]

\[
\text{monoester} + \text{acid} \rightleftharpoons \text{diester} + \text{water} \quad (R2)
\]

The equilibrium constants of these esterification reactions are typically about 1–10.9 Water is usually removed during the process to shift the reaction equilibrium toward the products. The disproportionation reaction has often been ignored when this reaction mechanism has been examined.10 The reaction equation for disproportionation is

\[
\text{glycol} + \text{diester} \rightleftharpoons 2 \text{monoester} \quad (R3)
\]

This reaction can be very unfavorable because it also takes place during separation, reducing the amount of monoester which is normally the desired product. When sulfonated styrene–divinylbenzene copolymers became available, they were used as catalysts for acetylation of alcohols with acetic acid.11 These resins can retain their activity for a long period of time; some samples of resins can be used in 50 esterification cycles without regeneration.12 Mineral acids have usually been used to catalyze esterification reactions, but nowadays ion exchange resins have gained prominence because they are easy to remove, noncorrosive, and they function in rather mild reaction conditions.13 Cation-exchange resins, as Brønsted acids, are sulfonic acid cation exchangers in the hydrogen form. These resins can be divided into two groups having major structural differences: gel resins and macroreticular resins.14 Gel resins can only function effectively as catalysts in a swelling medium, but macroreticular resins can function also in nonswelling solvents. Gel resins are characterized by a divinylbenzene content generally below 12%.14 The divinylbenzene content of macroreticular resins varies from 18% (Lewatit SPC 118) to >50% (Amberlyst XN-1010).14 The product selectivity for these catalysts is often higher than that obtained by dissolved electrolyte catalysts.14,15 Activities higher than soluble para-toluensulfonic acid have been reported for the addition of alcohols to olefins using ion-exchange resin catalysts.14,16

The catalyst application of ion-exchange resins in different reactions has been thoroughly investigated,11,14 and the esterification of carboxylic acid with olefins in the presence of cation-exchange resins has considerable academic and industrial
The heterogeneously catalyzed esterification reaction of succinic acid with methanol over a sulfonic ion-exchange resin (Relite CFS) catalyst has been investigated in order to simulate the preliminary esterification step in the biodiesel production process. The use of ion-exchange resins both as a catalyst as well as a selective sorbent in a continuous chromatographic reactor has been widely studied. The applicability as a catalyst for fuel oxygenation agent production has also been thoroughly researched, and nowadays MTBE production is catalyzed using a strong acidic macrotiregular ion-exchange resin. The heterogeneously catalyzed esterification reaction of succinic acid with ethanol over macrotiregular resin (Amberlyst 15) has been researched in order to get a kinetic model for the design and simulation of processes such as reactive distillation for diethyl succinate formation. These organic acid esters produced by the reaction of organic acids and alcohols can be entirely biorenewable.

The aim of this work is to study the reaction kinetics of neopentyl glycol (NPG) with different carboxylic acids (ACID). Commercial heterogeneous gelular (Dowex 50WX2) and macrotiregular (Amberlyst 15) type resins and homogencous para-toluenesulfonic acid (TSA) are compared as catalysts. The purpose of this work is also to compare the kinetic results for different acids and catalysts. The Markov chain Monte Carlo (MCMC) method is applied to calculate the reliability of the obtained results. The MCMC method has been rather seldom used in chemical kinetics to analyze the reliabilities of estimated parameters. This method has been earlier applied in the estimation of Arrhenius parameters for the esterification reaction of neopentyl glycol (NPG) with propionic acid (PA).

2. Experimental Section

Experiments (Table 1, Vahteristo et al.) were performed in a 650 mL glass batch reactor (OM Lasilait Oy, DN100) with a glass cooler (length about 0.33 m) with a spiral cooling coil, tubes for nitrogen (AGA) flow (length about 0.25 m), and a thermocouple (LAUDA PT-100–70). Reactants were mixed using a stirrer (IKA EUROSTAR power control-visc.) and heated with a thermostat (LAUDA C6 CP). Experiments were carried out under nearly isothermal conditions. The temperature variation was less than 0.3 K during each experiment. The nitrogen volume flow was controlled using a mass flow controller (Brooks 5850E SN). Nitrogen was preheated to the reactor temperature in a glass tube through which it was fed into the reactor. A detailed description of the reactor setups for esterification and disproportionation reactions is presented in an earlier publication. At the beginning of the esterification batch experiment, the reactants were weighed, mixed, and poured into the reactor. Mixing (500 rpm) and heating were started with the temperature being controlled with a thermostat. When the set temperature was reached, the weighed catalyst (1 g, 0.29 wt %) was put into the reactor and timing was started. The water produced was distilled out from the reactor by means of nitrogen flow through a chiller that was placed on the reactor vent line to condense the water. The mass of the condensed water was measured.

Table 1. Esterification and Disproportionation Experiments

<table>
<thead>
<tr>
<th>Reactant, mol/mol catalyst</th>
<th>Temperature (°C)</th>
<th>Run</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPG, 1/1</td>
<td>373</td>
<td>ES1</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>383</td>
<td>E8</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>393</td>
<td>E3</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>383</td>
<td>D1</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>383</td>
<td>D2</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>408</td>
<td>E6</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>563</td>
<td>E7</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>373</td>
<td>E8</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>383</td>
<td>E9</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>393</td>
<td>E10</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>408</td>
<td>E11</td>
</tr>
<tr>
<td>MEIBA Dowex</td>
<td>393</td>
<td>D9</td>
</tr>
<tr>
<td>MEIBA Dowex</td>
<td>423</td>
<td>D9</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>383</td>
<td>E12</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>393</td>
<td>E13</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>408</td>
<td>E14</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>423</td>
<td>E15</td>
</tr>
<tr>
<td>MEIBA sodium acetate</td>
<td>453</td>
<td>D9</td>
</tr>
<tr>
<td>MEIBA sodium acetate</td>
<td>453</td>
<td>D9</td>
</tr>
<tr>
<td>MEIBA Dowex</td>
<td>393</td>
<td>D9</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>383</td>
<td>E16</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>403</td>
<td>E17</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>423</td>
<td>E18</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>423</td>
<td>E19</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>383</td>
<td>E20</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>403</td>
<td>E21</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>413</td>
<td>E22</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>423</td>
<td>E23</td>
</tr>
<tr>
<td>NPG, 1/2</td>
<td>423</td>
<td>E24</td>
</tr>
</tbody>
</table>

Table 2. Comparison of the Properties of Resins

<table>
<thead>
<tr>
<th>Type</th>
<th>Dowex 50WX2</th>
<th>Amberlyst 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>strong acidic cation exchange</td>
<td>strongly acidic cation exchange</td>
</tr>
<tr>
<td>Functional structure</td>
<td>nuclear sulfonic acid</td>
<td>$\text{SO}_3^-$H$^+$</td>
</tr>
<tr>
<td>Surface area, m$^2$ g$^{-1}$</td>
<td>loss of drying (110 °C)</td>
<td>loss of drying (110 °C)</td>
</tr>
<tr>
<td>Moisture content</td>
<td>≈ 80%</td>
<td>≈ 5%</td>
</tr>
<tr>
<td>Ion exchange capacity</td>
<td>mg/g</td>
<td>4.6</td>
</tr>
<tr>
<td>MGD/kg</td>
<td>1.7</td>
<td>1.8</td>
</tr>
<tr>
<td>Porosity, %</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Maximum operating temperature °C</td>
<td>good up to 150</td>
<td>120</td>
</tr>
</tbody>
</table>

The kinetics of the esterification reaction of oleic acid with methanol over a sulfonic ion-exchange resin (Relite CFS) catalyst has been investigated in order to simulate the preliminary esterification step in the biodiesel production process. The use of ion-exchange resins both as a catalyst as well as a selective sorbent in a continuous chromatographic reactor has been widely studied. Their applicability as a catalyst for fuel oxygenation agent production has also been thoroughly researched, and nowadays MTBE production is catalyzed using a strong acidic macrotiregular ion-exchange resin. The heterogeneously catalyzed esterification reaction of succinic acid with methanol over macrotiregular resin (Amberlyst 15) has been researched in order to get a kinetic model for the design and simulation of processes such as reactive distillation for diethyl succinate formation. These organic acid esters produced by the reaction of organic acids and alcohols can be entirely biorenewable.

The aim of this work is to study the reaction kinetics of neopentyl glycol (NPG) with different carboxylic acids (ACID). Commercial heterogeneous gelular (Dowex 50WX2) and macrotiregular (Amberlyst 15) type resins and homogencous para-toluenesulfonic acid, a gelular type resin (Dowex 50 WX 2, Fluka Chemicals) and a macrotiregular type resin (Amberlyst 15, Fluka Chemicals) were used as catalysts. The gelular type resin was dried at a temperature of 383 K in order to get the correct catalyst mass because the loss on drying was approximately 80%. For comparison purposes some additional experiments were performed using a wet gelular type resin. The results were the same as for the dried resin. Other properties of the resin catalysts are given in Table 2.

Samples were diluted with ethylacetate (>99%, Fluka) and analyzed with a gas–liquid chromatograph (GLC) (HP 8590 SERIES II PLUS). Conditions during the analysis were initial temperature, 50 °C; initial time, 3 min; heating rate, 5 °C/min to final temperature 280 °C; injector temperature, 280 °C; and FID-detector, temperature 300 °C. The injector was split/splitless and the split ratio was 1:100. The peaks of NPG, PA, IBA, EHA, and their corresponding monoesters and diesters were identified by comparing with pure components, which were also used for the calculation of the response factors. Some samples

<table>
<thead>
<tr>
<th>Reactant</th>
<th>Temperature (°C)</th>
<th>Run</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPG, 1/1</td>
<td>373</td>
<td>ES1</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>383</td>
<td>E8</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>393</td>
<td>E3</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>383</td>
<td>D1</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>383</td>
<td>D2</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>408</td>
<td>E6</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>563</td>
<td>E7</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>373</td>
<td>E8</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>383</td>
<td>E9</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>393</td>
<td>E10</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>408</td>
<td>E11</td>
</tr>
<tr>
<td>MEIBA Dowex</td>
<td>393</td>
<td>D9</td>
</tr>
<tr>
<td>MEIBA Dowex</td>
<td>423</td>
<td>D9</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>383</td>
<td>E12</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>393</td>
<td>E13</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>408</td>
<td>E14</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>423</td>
<td>E15</td>
</tr>
<tr>
<td>MEIBA sodium acetate</td>
<td>453</td>
<td>D9</td>
</tr>
<tr>
<td>MEIBA sodium acetate</td>
<td>453</td>
<td>D9</td>
</tr>
<tr>
<td>MEIBA Dowex</td>
<td>393</td>
<td>D9</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>383</td>
<td>E16</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>403</td>
<td>E17</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>423</td>
<td>E18</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>423</td>
<td>E19</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>383</td>
<td>E20</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>403</td>
<td>E21</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>413</td>
<td>E22</td>
</tr>
<tr>
<td>NPG, 1/1</td>
<td>423</td>
<td>E23</td>
</tr>
<tr>
<td>NPG, 1/2</td>
<td>423</td>
<td>E24</td>
</tr>
</tbody>
</table>
were analyzed before dilution with a thermal conductive detector (TCD) in order to clarify the amount of water in the product. When using a flame ionization detector (FID) the column was 30 m cross-linked methyl siloxane (HP-1MS, 19091S-933) and with TCD it was 60 m cross-linked methyl silicone gum (HP-1, 19091Z-236).

3. Modeling and results

3.1. Product Mixture. In the esterification of neopentyl glycol with different acids, the observed products were neopentyl glycol monoiso-butryate (MEIBA) and neopentyl glycol di-isobutyrate (DEIBA) when using isobutyric acid (IBA); neopentyl glycol monopropionate (MEPA) and neopentyl glycol dipropionate (DEPA) when using propionic acid (PA); and neopentyl glycol mono-2-ethylhexanate (MEEHA) and neopentyl glycol di-2-ethylhexanate (DEEHA) when using 2-ethylhexanoic acid (EHA). Of course, every reaction produced also water (W). Only traces of other products were detected in the reaction mixture.

3.2. Kinetic Model. Three reactions take place simultaneously when performing esterification of NPG with PA, IBA, or EHA. The reaction scheme for the esterification of NPG with IBA to produce MEIBA and W. MEIBA can react further with IBA to produce DEIBA and W. The third reaction is a disproportionation equilibrium reaction between DEIBA and NPG to produce 2 moles of MEIBA. The existence of the disproportionation reaction was experimentally observed when MEIBA and MEPA\textsuperscript{10} were used as reactants (Table 1 and 3). Experiments using DEPA and NPG as reactants were also performed.\textsuperscript{10} Disproportionation reaction experiments with MEEHA and DEEHA were not performed because the separation of these compounds from the product mixture was problematic.

Using a batch reactor model and applying the reaction scheme in Figure 1, the production rate for each component can be calculated from eqs 1–5.

\[
\frac{\text{d}C_{\text{NPG}}}{\text{d}t} = -k_1C_{\text{NPG}}C_{\text{ACID}} + k_2C_{\text{ME}}C_{\text{W}} - k_3C_{\text{DE}}C_{\text{NPG}} + k_4C_{\text{ME}}^2\quad (1)
\]

\[
\frac{\text{d}C_{\text{ACID}}}{\text{d}t} = -k_1C_{\text{NPG}}C_{\text{ACID}} + k_2C_{\text{ME}}C_{\text{W}} - k_3C_{\text{ACID}}C_{\text{ME}} + k_4C_{\text{ACID}}^2\quad (2)
\]

\[
\frac{\text{d}C_{\text{ME}}}{\text{d}t} = k_1C_{\text{ME}}C_{\text{ACID}} - k_2C_{\text{ME}}C_{\text{W}} - k_3C_{\text{ACID}}C_{\text{ME}} + k_4C_{\text{ACID}}C_{\text{ME}}^2 + 2k_5C_{\text{DE}}C_{\text{NPG}} - 2k_6C_{\text{ME}}^2\quad (3)
\]

\[
\frac{\text{d}C_{\text{DE}}}{\text{d}t} = k_1C_{\text{ME}}C_{\text{ACID}} - k_2C_{\text{DE}}C_{\text{W}} - k_3C_{\text{ACID}}C_{\text{ME}} + k_4C_{\text{ACID}}C_{\text{ME}}^2 + k_5C_{\text{ME}}^2\quad (4)
\]

\[
\frac{\text{d}C_{\text{W}}}{\text{d}t} = k_1C_{\text{NPG}}C_{\text{ACID}} - k_2C_{\text{DE}}C_{\text{W}} - k_3C_{\text{ME}}C_{\text{ACID}} + k_4C_{\text{ME}}C_{\text{ACID}} - k_5C_{\text{DE}}C_{\text{W}}\quad (5)
\]

where \( r \) is the production rate, \( C \) is the concentration, \( k \) is the apparent reaction rate constant, and \( t \) is time. ACID refers to PA, IBA, or EHA. ME and DE refer to the corresponding monoester or diester.

As a result of continuous water removal the amount of the reaction mixture decreases during the experiments. The molar amount of NPG is obtained from

\[
\text{m}_{\text{NPG}} = C_{\text{NPG}}m_{\text{mix}}\quad (6)
\]

Here \( m_{\text{NPG}} \) is the molar amount of NPG and \( m_{\text{mix}} \) is the weight of the reaction mixture. If eq 6 is differentiated

\[
\frac{\text{d}m_{\text{NPG}}}{\text{d}t} = \frac{\text{d}C_{\text{NPG}}}{\text{d}t} \quad (7)
\]

and both parts of eq 7 are divided by the time differential, eq 8 is obtained.

\[
\frac{\text{d}m_{\text{NPG}}}{\text{d}t} = \frac{\text{d}C_{\text{NPG}}}{\text{d}t} + C_{\text{NPG}}\frac{\text{d}m_{\text{mix}}}{\text{d}t}\quad (8)
\]

Assuming that the concentration of water in the reactor is negligible, eq 1 for the molar production rate of NPG can be rewritten as

\[
\frac{\text{d}C_{\text{NPG}}}{\text{d}t} = -k_1C_{\text{NPG}}C_{\text{ACID}} - k_2C_{\text{DE}}C_{\text{NPG}} + k_4C_{\text{ME}}^2\quad (9)
\]

Multiplying eq 9 by \( m_{\text{mix}} \) obtains

\[
\frac{\text{d}C_{\text{NPG}}}{\text{d}t} = (-k_1C_{\text{NPG}}C_{\text{ACID}} - k_2C_{\text{DE}}C_{\text{NPG}} + k_4C_{\text{ME}}^2) m_{\text{mix}}\quad (10)
\]

Combining eqs 10 and 8 gives

\[
m_{\text{mix}} \frac{\text{d}C_{\text{NPG}}}{\text{d}t} + C_{\text{NPG}} \frac{\text{d}m_{\text{mix}}}{\text{d}t} = (-k_1C_{\text{NPG}}C_{\text{ACID}} - k_2C_{\text{DE}}C_{\text{NPG}} + k_4C_{\text{ME}}^2) m_{\text{mix}}\quad (11)
\]

After rearrangement

\[
\frac{\text{d}C_{\text{NPG}}}{\text{d}t} = -k_1C_{\text{NPG}}C_{\text{ACID}} - k_2C_{\text{DE}}C_{\text{NPG}} + k_4C_{\text{ME}}^2 - \frac{C_{\text{NPG}}\frac{\text{d}m_{\text{mix}}}{\text{d}t}}{m_{\text{mix}}}\quad (12)
\]

By adapting eqs 6, 7, 10, and 12 for each component, eqs 1–4 can be rewritten as

\[
\frac{\text{d}C_{\text{NPG}}}{\text{d}t} = -k_1C_{\text{NPG}}C_{\text{ACID}} - k_2C_{\text{DE}}C_{\text{NPG}} + k_4C_{\text{ME}}^2 - \frac{C_{\text{NPG}}\frac{\text{d}m_{\text{mix}}}{\text{d}t}}{m_{\text{mix}}}\quad (13)
\]
M

where the parametrizations of the model that, statistically, fit the data can be estimated from the data, as shown by the MCMC analysis (Figure 3). Section 3.3 gives details about the MCMC technique.

The temperature dependence of the reaction rate coefficients is expressed by the Arrhenius law.

\[ k = A \exp\left(-\frac{E}{RT}\right) \]

The kinetic model formed by eqs 13–19 has seven parameters, which are \( A_{1,\text{mean}}, A_{2,\text{mean}}, A_{3,\text{mean}}, E_1, E_2, E_3, \) and \( K_3 \). These parameters have to be estimated on the basis of experimental data.

### 3.3. Parameter Estimation and MCMC Analysis

The equations for the reaction kinetic model form a system of ordinary differential equations (eqs 13–18 and 20) which can be integrated starting from the initial conditions. The reaction kinetic parameters were first estimated with standard least-squares fitting by minimizing the squared difference between the measured and calculated concentrations. The estimated reaction kinetic parameters and traditional error estimates are presented in Table 3. In almost all series of experiments the coefficient of determination was over 97%, indicating good agreement between the measured and calculated concentrations. In the esterification of NPG with PA using TSA as a catalyst (runs ES4, ES5, and DP1), the coefficient of determination was only 85.9%. The reason for this is that in this case experiments were performed only at two temperatures.

To evaluate the accuracy of the estimated parameters (eq 21) in a nonlinear multiparameter model, it is important to consider possible cross-correlation and identifiability of the parameters. Classical statistical analysis (see Table 3) that gives the optimal parameter values, error estimates for them, and correlations between them is approximate (based on linearization of the model) and may sometimes be quite misleading. Moreover, the question of the reliability of the model predictions is left open, that is, how is the uncertainty in model parameters reflected to the model response (Figure 4). Both these problems may be properly treated by Markov chain Monte Carlo (MCMC) methods. Using MCMC methods, the estimation of model parameters and predictions are performed according to a Bayesian paradigm. All uncertainties in the data as well as the modeling results are treated as random variables that have statistical distributions. Instead of a single fit to the data, “all” the parametrizations of the model that, statistically, fit the data

\[ r_{\text{ACID}} = \frac{dC_{\text{ACID}}}{dt} = -k_1C_{\text{NPG}}C_{\text{ACID}} - k_2C_{\text{ME}}C_{\text{ACID}} - \frac{C_{\text{ACID}}\, dm_{\text{mix}}}{m_{\text{mix}}\, dt} \]  

(14)

\[ r_{\text{ME}} = \frac{dC_{\text{ME}}}{dt} = k_1C_{\text{NPG}}C_{\text{ACID}} - k_2C_{\text{ME}}C_{\text{ACID}} + 2k_3C_{\text{NPG}}C_{\text{DE}} - 2k_3C_{\text{ME}} = \frac{C_{\text{ME}}\, dm_{\text{mix}}}{m_{\text{mix}}\, dt} \]  

(15)

\[ r_{\text{DE}} = \frac{dC_{\text{DE}}}{dt} = k_1C_{\text{ME}}C_{\text{ACID}} - k_2C_{\text{ME}}C_{\text{ACID}} + k_3C_{\text{ME}}^2 = \frac{C_{\text{DE}}\, dm_{\text{mix}}}{m_{\text{mix}}\, dt} \]  

(16)

Multiplying eq 5 by \( m_{\text{mix}} \) gives eqs 17 and 18. The concentration of water in the reactor is assumed to be negligible.

\[
  r_w = \frac{dm_w}{dt} = (k_3C_{\text{NPG}}C_{\text{ACID}} + k_3C_{\text{ME}}C_{\text{ACID}})m_{\text{min}}
\]

(17)

\[
  \frac{dm_{\text{mix}}}{dt} = -M_w \frac{dm_w}{dt} = -r_w m_{\text{mix}} M_w
\]

(18)

where \( M_w \) is the molar mass of water. Then eq 18 can be used to evaluate the decrease of the mass of the reaction mixture in eqs 13–16. Using eq 18 eq 13 can be rewritten as

\[
  r_{\text{NPG}} = \frac{dC_{\text{NPG}}}{dt} = -k_1C_{\text{NPG}}C_{\text{ACID}} - k_2C_{\text{NPG}}C_{\text{DE}} + k_3C_{\text{NPG}}C_{\text{ME}} + \frac{r_w M_w m_{\text{mix}}}{m_{\text{mix}}} \]

(19)

Equations 14–16 can be rewritten similarly.

The equilibrium constant for the disproportionation reaction is defined as

\[
  K_3 = \frac{k_3}{k_1}\]

(20)

The disproportionation reaction was neglected if experimental data for the reaction of ME or DE with NPG was not available. In this case the Arrhenius law parameters for this reaction cannot be integrated starting from the initial conditions. The reaction kinetic parameters were estimated with standard least-squares fitting by minimizing the squared difference between the measured and calculated concentrations. The estimated reaction kinetic parameters and traditional error estimates are presented in Table 3. In almost all series of experiments the coefficient of determination was over 97%, indicating good agreement between the measured and calculated concentrations. In the esterification of NPG with PA using TSA as a catalyst (runs ES4, ES5, and DP1), the coefficient of determination was only 85.9%. The reason for this is that in this case experiments were performed only at two temperatures.

To evaluate the accuracy of the estimated parameters (eq 21) in a nonlinear multiparameter model, it is important to consider possible cross-correlation and identifiability of the parameters. Classical statistical analysis (see Table 3) that gives the optimal parameter values, error estimates for them, and correlations between them is approximate (based on linearization of the model) and may sometimes be quite misleading. Moreover, the question of the reliability of the model predictions is left open, that is, how is the uncertainty in model parameters reflected to the model response (Figure 4). Both these problems may be properly treated by Markov chain Monte Carlo (MCMC) methods. Using MCMC methods, the estimation of model parameters and predictions are performed according to a Bayesian paradigm. All uncertainties in the data as well as the modeling results are treated as random variables that have statistical distributions. Instead of a single fit to the data, “all” the parametrizations of the model that, statistically, fit the data.

![Figure 1. Reaction scheme for the esterification of NPG with IBA.](image-url)
FORTRAN 90 software package MODEST27 was used for both ES7 used as the reactant acid and Dowex as the catalyst (runs from the MCMC analysis are shown for the case when IBA is apparent reaction rates and their ratios (Figures 5 and 6).

parameters. This property was used to study the uncertainties for uncertainty estimates for any function of the unknown parameters is generated using available prior information (e.g., statistical knowledge of the observation noise. Computationally, the distribution is generated using the Markov chain Monte Carlo (MCMC) sampling approach. Up-to-date adaptive computational schemes are employed in order to make the simulations as effective as possible. 25,26 A FORTRAN 90 software package MODEST27 was used for both the least-squares and the MCMC estimation. The referred methods are also implemented in a MATLAB package.26,28

The MCMC output can additionally be used to construct uncertainty estimates for any function of the unknown parameters. This property was used to study the uncertainties for apparent reaction rates and their ratios (Figures 5 and 6).

The results of the MCMC analysis for the Arrhenius law parameters are depicted in Figures 2–3. In Figure 2 the results from the MCMC analysis are shown for the case when IBA is used as the reactant acid and Dowex as the catalyst (runs ES7–11, DP3–4). The algorithm has produced a sample of size 50000 from the full distribution of the parameters, from which 1D and 2D marginal distributions are presented. The results show that the estimated parameters of all reactions are rather well identified - the distributions are centered around the least-squares estimates. Figure 3 presents the corresponding analysis when using EHA and Dowex (runs ES20–24). When considering the esterification reactions (reactions 1 and 2 in Figure 1) the estimated parameters are rather well identified. However, the disproportionation reaction (reaction 3 in Figure 1) is poorly identified because of the lack of disproportionation experiments. It is for this reason that the disproportionation reaction is ignored if disproportionation experiments were not performed (Table 3).

### 3.4. Comparison of the Esterification Rate Constants

In esterification of NPG with PA, IBA or EHA the formation of the corresponding monoester was clearly much faster than the consecutive esterification of the monoester to diester (Figure 4, ES9). In Figure 5a,b the apparent reaction rate constants and their uncertainties for the corresponding reactions (k₁ and k₂) are depicted as a function of the inversed reaction temperature (1/T) using the fitted Arrhenius law parameters from the MCMC analysis. For IBA the ratio of the apparent reaction rate constants for the corresponding reactions (k₁/k₂) was about 5.2 (standard deviation 17%) at 383 K when using Dowex (0.27 wt %) as a catalyst (Figure 5c). For PA with a smaller molecule size the ratio was 2.8 (std 5%) and for EHA with larger molecule size the ratio was 9.3 (std 11%). These ratios are quite unreliable because the further esterification of monoester to diester (k₂) is a secondary and minor reaction. Nevertheless, it gives a clue to how much more difficult it is to produce diesters from monoesters of large molecules than it is to produce diesters from small molecules. Of course, when the size of the acid molecule is increased the apparent reaction rate constant of NPG esterification to monoester (k₁) decreases (Figure 5a). For PA the apparent reaction rate constant was 4.35 × 10⁻³ kg mol⁻¹

<p>| Table 3. Estimated Reaction Kinetic Parameters Using Different Acids and Catalysts |
|---------------------------------|-----------------|----------------|-----------------|-----------------|</p>
<table>
<thead>
<tr>
<th>acid</th>
<th>catalyst</th>
<th>( r_{	ext{max}} (\text{K}) )</th>
<th>parameter</th>
<th>estimated value</th>
<th>standard error (%)</th>
<th>coefficient of determination (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA</td>
<td>nothing</td>
<td>400</td>
<td>( A_{\text{IBA}} )</td>
<td>0.827 × 10⁻³ kg mol⁻¹</td>
<td>3.5</td>
<td>99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( A_{\text{IBA}} )</td>
<td>0.335 × 10⁻³ kg mol⁻¹</td>
<td>15.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( E_1 )</td>
<td>4.48 × 10⁴ J mol⁻¹</td>
<td>5.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( E_2 )</td>
<td>2.40 × 10⁴ J mol⁻¹</td>
<td>53.3</td>
<td></td>
</tr>
<tr>
<td>PA</td>
<td>Dowex 50WX2</td>
<td>379</td>
<td>( A_{\text{IBA}} )</td>
<td>10.8 × 10⁻³ kg mol⁻¹</td>
<td>7.9</td>
<td>85.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( A_{\text{IBA}} )</td>
<td>1.47 × 10⁻³ kg mol⁻¹</td>
<td>45.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( E_1 )</td>
<td>6.6 × 10⁴ J mol⁻¹</td>
<td>211</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( E_2 )</td>
<td>5.59 × 10⁴ J mol⁻¹</td>
<td>3.16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( K_1 )</td>
<td>3.99 × 10⁴ J mol⁻¹</td>
<td>26.6</td>
<td></td>
</tr>
<tr>
<td>PA</td>
<td>Dowex 50WX2</td>
<td>400</td>
<td>( A_{\text{IBA}} )</td>
<td>9.3 × 10⁻³ kg mol⁻¹</td>
<td>3.6</td>
<td>97.18</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( A_{\text{IBA}} )</td>
<td>3.25 × 10⁻³ kg mol⁻¹</td>
<td>3.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( A_{\text{IBA}} )</td>
<td>0.707 × 10⁻³ kg mol⁻¹</td>
<td>4.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( E_1 )</td>
<td>5.74 × 10⁴ J mol⁻¹</td>
<td>4.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( E_2 )</td>
<td>5.53 × 10⁴ J mol⁻¹</td>
<td>4.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( K_1 )</td>
<td>4.68 × 10⁴ J mol⁻¹</td>
<td>8.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( K_2 )</td>
<td>6.33</td>
<td>4.6</td>
<td></td>
</tr>
<tr>
<td>IBA</td>
<td>Amberlyst 15</td>
<td>400</td>
<td>( A_{\text{IBA}} )</td>
<td>1.86 × 10⁻³ kg mol⁻¹</td>
<td>1.7</td>
<td>98.63</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( A_{\text{IBA}} )</td>
<td>0.407 × 10⁻³ kg mol⁻¹</td>
<td>6.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( A_{\text{IBA}} )</td>
<td>5.83 × 10⁴ J mol⁻¹</td>
<td>2.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( A_{\text{IBA}} )</td>
<td>5.63 × 10⁴ J mol⁻¹</td>
<td>9.0</td>
<td></td>
</tr>
<tr>
<td>IBA</td>
<td>Dowex 50WX2</td>
<td>379</td>
<td>( A_{\text{IBA}} )</td>
<td>1.66 × 10⁻³ kg mol⁻¹</td>
<td>1.2</td>
<td>99.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( A_{\text{IBA}} )</td>
<td>0.324 × 10⁻³ kg mol⁻¹</td>
<td>5.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( A_{\text{IBA}} )</td>
<td>0.16 × 10⁻³ kg mol⁻¹</td>
<td>12.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( E_1 )</td>
<td>5.51 × 10⁴ J mol⁻¹</td>
<td>1.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( E_2 )</td>
<td>5.08 × 10⁴ J mol⁻¹</td>
<td>7.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( E_3 )</td>
<td>3.46 × 10⁴ J mol⁻¹</td>
<td>8.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( K_1 )</td>
<td>3.33</td>
<td>8.0</td>
<td></td>
</tr>
<tr>
<td>EHA</td>
<td>TSA</td>
<td>400</td>
<td>( A_{\text{IBA}} )</td>
<td>1.74 × 10⁻³ kg mol⁻¹</td>
<td>1.6</td>
<td>98.62</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( A_{\text{IBA}} )</td>
<td>0.292 × 10⁻³ kg mol⁻¹</td>
<td>8.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( A_{\text{IBA}} )</td>
<td>5.40 × 10⁴ J mol⁻¹</td>
<td>2.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( A_{\text{IBA}} )</td>
<td>6.35 × 10⁴ J mol⁻¹</td>
<td>9.4</td>
<td></td>
</tr>
<tr>
<td>EHA</td>
<td>Dowex 50WX2</td>
<td>400</td>
<td>( A_{\text{IBA}} )</td>
<td>0.471 × 10⁻³ kg mol⁻¹</td>
<td>1.1</td>
<td>99.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( A_{\text{IBA}} )</td>
<td>0.06 × 10⁻³ kg mol⁻¹</td>
<td>12.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( A_{\text{IBA}} )</td>
<td>6.87 × 10⁴ J mol⁻¹</td>
<td>1.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( A_{\text{IBA}} )</td>
<td>8.12 × 10⁴ J mol⁻¹</td>
<td>10.8</td>
<td></td>
</tr>
</tbody>
</table>
min\(^{-1}\) (std 2%) and for IBA and EHA it was 2.01 \(\times\) 10\(^{-3}\) (std 1%) and 0.19 \(\times\) 10\(^{-3}\) (std 2%) kg mol\(^{-1}\) min\(^{-1}\) over Dowex (0.27 wt %) at 383 K, respectively.

The esterification reactions proceeded also without a catalyst (Figure 5). The apparent reaction rate constants for the esterification of NPG with PA to MEPA and for the esterification of MEPA with PA to DEPA were 0.45 \(\times\) 10\(^{-3}\) (std 2%) and 0.24 \(\times\) 10\(^{-3}\) (std 11%) kg mol\(^{-1}\) min\(^{-1}\) at 383 K without a catalyst (Table 3). When using Dowex, the apparent reaction rate constants were 4.35 \(\times\) 10\(^{-3}\) (std 11%) kg mol\(^{-1}\) min\(^{-1}\) at 383 K without a catalyst (Table 3). When using Dowex, the apparent reaction rate constants were 5.79 \(\times\) 10\(^{-3}\) and 1.02 \(\times\) 10\(^{-3}\) kg mol\(^{-1}\) min\(^{-1}\), respectively. The gellular type resin (Dowex) was much more active than the macroreticular type resin (Amb) (Figure 6) although their ion-exchange capacities are about the same (Table 2). Due to its homogeneous nature, the most active catalyst was TSA which would be expected. For the esterification of NPG with EHA at 383 K the apparent reaction rate constant to monoester (MEEHA) was 0.85 \(\times\) 10\(^{-3}\) (std 3%) kg mol\(^{-1}\) min\(^{-1}\).
Figure 4. Comparison of the measured and possible predicted concentrations (95% confidence intervals) given by MCMC in the esterification of NPG with IBA at 383 K over Dowex (run ES 9). The dark gray area represents the possible model predictions with parameter values given by MCMC. The light gray area represents the 95% confidence band for measurements (model uncertainty and experimental error).

Figure 5. (a–c) Comparison of different carboxylic acids using apparent reaction rate constants of esterification (runs ES1-3, Vahteristo et al.,10 ES7-11, DP3-4, ES20-24). Uncertainty estimates are calculated using the MCMC output.
when using TSA as the catalyst and $0.19 \times 10^{-3}$ (std 2%) kg mol$^{-1}$ min$^{-1}$ when using Dowex (Figure 6a).

As presented earlier, it is possible to produce uncertainty estimates with MCMC for any functions of unknown parameters. The standard deviations reported above are calculated from the MCMC sample using this property of MCMC analysis. The uncertainties for the apparent reaction rates and reaction rate ratios are plotted in Figures 5a–c and 6a–c using different possible parameter values given by MCMC. This type of uncertainty estimate cannot be produced by traditional regression analysis.

3.5. Comparison of the Disproportionation Rate Constants. The disproportionation reaction was a minor reaction in the reaction system. The existence of the disproportionation reaction could, however, be experimentally verified using NPG and DEPA, MEPA, and MEIBA as reactants (Figure 7). It can be assumed that the disproportionation reaction also proceeded without a catalyst because the apparent reaction rate constant of MEIBA dissociation ($k_{2}$) was $0.015 \times 10^{-3}$ kg mol$^{-1}$ min$^{-1}$ when 159 ppm of sodium acetate was used as a catalyst at 180 °C (Table 1, DP5). The reaction rate constant increased to $0.044 \times 10^{-3}$ kg mol$^{-1}$ min$^{-1}$ when the amount of sodium acetate was increased to 1590 ppm (Table 1, DP6). When using a disproportionation catalyst like TSA, the apparent reaction rate constant of MEIBA dissociation was as high as $0.181 \times 10^{-3}$ kg mol$^{-1}$ min$^{-1}$ (Table 1, DP7) even at a temperature as low as 120 °C. Dowex also catalyzed disproportionation because $k_{2}$ was found to be $0.076 \times 10^{-3}$ kg mol$^{-1}$ min$^{-1}$ (Table 1, DP3).

When performing disproportionation of DEPA with NPG, the apparent reaction rate constant of the disproportionation reaction

4. Conclusions and Discussion

The kinetics of NPG esterification with different carboxylic acids was studied experimentally. The kinetic parameters, pre-exponential factors, and activation energies were estimated for
the pseudohomogeneous model and their reliability was studied using the MCMC method in addition to classical regression analysis.

The two major reactions in the reaction mechanism were the esterification reactions of NPG and ME

\[ \text{NPG} + \text{ACID} \rightarrow \text{ME} + W \]

\[ \text{ME} + \text{ACID} \rightarrow \text{DE} + W \]

An equilibrium reaction was also discovered, which was the disproportionation reaction

\[ \text{NPG} + \text{DE} \rightleftharpoons 2\text{ME} \]

The esterification experiments can be explained without disproportionation, but experimentally disproportionation certainly takes place.

The sampled parameter values (Figures 2 and 3) produced by MCMC, were used to create distributions for the computed response values to describe how the uncertainties in the estimated parameters affect the fitted concentrations (Figures 4 and 9). The fits were relatively good in most experiments and uncertainties were small (Figure 4). When performing esterification of NPG with PA over TSA, the level of uncertainty was higher, partly because experiments were done only at two temperatures (Figure 9).

MCMC was used to calculate uncertainty estimates for the apparent reaction rate constants and their ratios (Figures 5, 6, and 8) at different temperatures. MCMC turns out to be especially useful here—uncertainties related to model predictions and other nonlinear functions of estimated parameters cannot be handled with standard regression techniques.

The rate of esterification has been found by several authors to decrease with increasing cross-linking in the resin particles.13,29 This agrees well with the current results as the esterification activity was clearly higher when using a gelular type resin (Dowex) than when using a macroreticular type resin (Amb) (Figure 6). Cross-linkage-%DVB of Amb was as high as 20–25 whereas cross-linkage-%DVB of Dowex was only 2. The ratio of the apparent reaction rate constants for the resins \( k_1(\text{Dowex})/ k_1(\text{Amb}) \) was over 2 when performing esterification of NPG with IBA, although their ion-exchange capacities are about the same (Table 2).
It has been claimed that cross-linking in the macrorotectic resin particle has only little effect on the reaction rates. If this is the case, the increased reaction rate with Dowex can be explained by its swelling ability, which increases porosity and makes available the active nuclear sulfonic acid groups residing in the pores of the resin. The self-catalyzed reaction rate evidently contributes to esterification at low catalyst loadings when comparing the self-catalyzed reaction rate with that of the ion-exchange resin catalyzed. Catalyzed (Dowex) and self-catalyzed esterification reaction rate constants of PA and IBA differed clearly, but those of EHA do not necessarily differ so much because of its larger size. The activation energies for the esterification of NPG with PA, IBA, or EHA were 57.4 (std 4.8%), 55.1 (std 2%), or 68.7 (std 2%) kJ mol⁻¹ over Dowex 50 WX 2 (Table 3). This agrees well with the results obtained for the esterification of n-butanol with benzoic acid over the same catalyst, where the activation energy was about 75 kJ mol⁻¹.\footnote{Wicks, Z. W. Coatings. In Kirk-Othmer Encyclopedia of Chemical Technology; Kroschwitz, J. I., Howe-Grant, M., Eds.; John Wiley & Sons: New York, 1993, pp 669–746.}

Acknowledgment

TEKES (Finnish Funding Agency for Technology and Innovation, Finland) and Neste Oy are gratefully acknowledged for financial support.

Nomenclature

A = frequency factor, kg mol⁻¹ min⁻¹
Amb = Amberlyst 15
C = concentration, mol kg⁻¹
DDEHA = neopentyl glycol di-2-ethylhexanate
DEIBA = neopentyl glycol di-isobutyrate
DEPA = neopentyl glycol dipropionate
Dowex = Dowex 50WX2
DVB = divinylbenzene
EHA = 2-ethylhexanoic acid
IBA = isobutyric acid
Kₚ = apparent reaction rate constant, kg mol⁻¹ min⁻¹
Kₑq = equilibrium constant, -
me = weight of the reaction mixture, kg
MEEHA = neopentyl glycol mono-2-ethylhexanate
MEIBA = neopentyl glycol monoisobutyrate
MEPA = neopentyl glycol monopropionate
NPG = neopentyl glycol
PA = propionic acid
TSA = para-toluene sulfonic acid
r = reaction rate, mol min⁻¹ kg⁻¹
T = temperature, K
Tavg = average temperature, K
W = water

Subscripts and Superscripts

n, m = reactions of the model, see Figure 1
ACID = carboxylic acid (PA, IBA or EHA)
DE = diester (DEPA, DEIBA or DDEHA)
ME = monoester (MEPA, MEIBA or MEEHA)
NPG = neopentyl glycol
W = water

Literature Cited


Received for review October 20, 2008
Revised manuscript received April 23, 2009
Accepted May 6, 2009
Frog population viability under past, present and future climate conditions: a Bayesian state-space approach

Title: Frog population viability under past, present and future climate conditions: a Bayesian state-space approach

Running title: Climate effects on a montane amphibian

Authors: Rebecca McCaffery¹, Antti Solonen²,³, and Elizabeth Crone¹,⁴

Institutional affiliation:
¹Ecosystem and Conservation Sciences
College of Forestry and Conservation
University of Montana
Missoula, MT 59812
USA

²Department of Mathematics and Physics
Lappeenranta University of Technology
P. O. Box 20
FI-53851 Lappeenranta
Finland

³Finnish Meteorological Institute
P.O. Box 503
FI-00101 Helsinki
Finland

⁴Current affiliation:
Harvard Forest
Harvard University
324 N Main Street
Petersham MA 01366
USA
Summary

1. Worldwide extinctions of amphibians are at the forefront of the biodiversity crisis, with climate change figuring prominently as a potential driver of continued amphibian decline. As in other taxa, changes in both the mean and variability of climate conditions may affect amphibian populations in complex, unpredictable ways. In western North America, climate models predict a reduced duration and extent of mountain snowpack and increased variability in precipitation, which may have consequences for amphibians inhabiting montane ecosystems.

2. We used Bayesian capture-recapture methods to estimate vital rates in a high elevation population of the Columbia spotted frog (*Rana luteiventris*) over ten years, and related these rates to interannual variation in peak snowpack. Then, we forecasted frog population growth and viability under a range of scenarios with varying levels of change in mean and variance in snowpack.

3. Population models based on snowpack during our study period predicted a strongly declining population. However, snowpack data from the recent past predicted a viable population. This difference was mostly due to changes in snowpack mean, not variance. Similarly, over a range of future scenarios, changes in mean snowpack had a greater effect on viability than changes in the variance of snowpack, with forecasts generally predicting an increase in population viability.

4. Longevity and heterogeneous habitat may contribute to the potential for this amphibian species to be resilient to increased climatic variation, and shorter-lived species inhabiting homogeneous ecosystems may be more susceptible to increased variability in climate conditions.

5. Although mean conditions were more important to viability than variance, for a given mean snowpack depth, increases in variability could change a population from increasing to decreasing. Therefore, both mean and variance in snowpack conditions can influence predictions of population viability, and the influence of changing climate variability on populations should be accounted for in predictive models. The Bayesian modeling framework allows for the explicit characterization of uncertainty in parameter estimates and ecological forecasts, and thus provides a natural approach for examining relative contributions of mean and variability in climatic variables to population dynamics.

**Key-words:** Bayesian state space model; capture-recapture analysis; climate change; Columbia spotted frog; population viability analysis; *Rana luteiventris*; Rocky Mountains, USA; snow water equivalence
Introduction

Climate change has emerged as one of the greatest threats to global biodiversity, and ecologists and conservation biologists will be increasingly called upon to determine how a changing climate will affect individual species. Studies of climate change impacts have involved tracking changes in phenology (Root et al., 2003), distribution (Walther et al., 2002), survival (Wang et al., 2002, e.g. Post and Stenseth, 1998), and interspecific interactions (e.g. Kaeriyama et al., 2004) in relation to climatic variables. Documenting changes in phenology and distribution is important, but we ultimately want to know whether changes in climate will affect the viability of plant and animal populations. Predicting changes in population viability is a fundamental challenge, because knowledge of demographic vital rates (i.e. survival, growth, and fecundity) is required, and the demographic research needed to quantify these parameters over appropriate time scales is logistically daunting. However, such studies are necessary to determine the influence of various climatic factors on individual vital rates, population growth rate, and risk of extinction.

Most attention in the climate literature has been focused on change in mean climate conditions (Knowles et al., 2006, Stewart et al., 2004, Mote et al., 2005, Payne et al., 2004), and some authors have begun to address how changes in mean climate conditions might affect wild populations (Brodie and Post, 2010, Post and Stenseth, 1998, Wang et al., 2002, Scherer et al., 2008, McCaffrey and Maxell, 2010). However, climate models also predict an increase in climatic variability (Easterling et al., 2000, MacCracken et al., 2001, Raisanen, 2002, Watterson, 2005), and fewer researchers have examined how changes in variability of climate across years might affect population viability and species distributions (but see van de Pol et al., 2010, Colchero et al., 2009, Zimmermann et al., 2009, Jonzen et al., 2010). In one of the few studies to look at variance in climate indices, Zimmerman et al. (2009) found that predictions of spatial distributions of tree species in Switzerland were improved by including climatic extremes in their models. In contrast, studies of three long-lived species found that after other factors were taken into account, mean changes in drought, precipitation, and winter temperature, respectively, were more important for population viability than changes in the interannual variation of those variables (Colchero et al., 2009, Jonzen et al., 2010, van de Pol et al., 2010). Determining the extent to which changes in climatic variation affect our predictions of population viability, relative to changes in the mean, is key to developing meaningful forecasts of climate change on individual species.

In this paper, we assess the importance of changes in the mean and variance of snowpack for *Rana luteiventris* Thompson, 1913, a frog species that inhabits the Rocky Mountains of western North America. Understanding the impact of climate change on amphibians is an issue of particular urgency; a third of all amphibian species are considered threatened, more than twice as many amphibians are critically endangered compared to mammals or birds (Stuart et al., 2004), and amphibian decline is considered evidence that the sixth mass extinction in the Earth’s history is underway (Wake and Vredenburg, 2008). Habitat destruction and disease are considered the principal causes of amphibian declines, but there is little doubt that climate change can and will interact with these factors to influence the rate and extent of declines (Corn, 2005). However, we know relatively little about how specific changes in climate may affect the demography of individual amphibian populations. Demographic studies of amphibians are essential to determine which life stages are affected by various stressors, and how these impacts combine to affect population growth rates and viability (Biek et al., 2002, Govindarajulu et al., 2005, Vonesh and De La Cruz, 2002).

Our study of Columbia spotted frog (*Rana luteiventris*) population viability is based on ten years of demographic data from a high elevation population in the US Rocky Mountains (McCaffrey and Maxell, 2010). From a previous analysis examining relationships between survival, growth, and fecundity rates and a suite of winter and summer climate variables, we found that juvenile and adult survival rates were inversely correlated with peak snowpack (McCaffrey and Maxell, 2010). Recent snowpack in western mountains has been reduced in volume and duration, a trend predicted to continue (Stewart et al., 2004), and one that has direct consequences for the hydrologic and temperature regimes of montane wetlands (Adam et al., 2009, McMenamin et al., 2008). This trend should increase *R. luteiventris* population viability. However, snowpack is also predicted to become more variable, the consequences of which are unknown.

To examine how changes in the mean and variability of peak annual snowpack might affect population viability, we estimated *R. luteiventris* vital rates using Bayesian capture-recapture methods, established relationships between vital rates and snowpack, and simulated population growth under current, historical, and possible future climate scenarios. Bayesian statistical methods are useful for examining relationships between...
climate variables and life history parameters, because they provide a framework for extending uncertainty in parameter estimates to derived statistics such as stochastic population growth rates and extinction risks (Clark, 2005, Ellison, 2004, Wade, 2000). Parameters are expressed as probability distributions rather than point estimates, which allows a more explicit representation of uncertainty in the estimates; uncertainty can then be translated into ecological forecasts by integrating (or sampling) over the distributions of each empirically estimated parameter. In this study, we use probability distributions of vital rates in relation to climate variables to simulate the distribution of population growth rates and extinction risk under past and future climate scenarios. We then compare the sensitivity of population viability metrics to changes in mean snowpack versus variance in snowpack across these scenarios for this frog population with highly variable population dynamics.

Materials and methods

Study area and climate data

We conducted demographic monitoring of a Columbia spotted frog population in the Bitterroot Mountains of western Montana, USA from 2000-2009. Our study population was located at 2200 m in the Little Rock Creek drainage of the Selway-Bitterroot Wilderness. This basin is delineated by glacial headwalls, and the frog population here is isolated from other nearby populations (Funk et al., 2005). For a complete description of the study site, see McCaffrey and Maxell (2010). We obtained precipitation data from the Twin Lakes SNOTEL site, located approximately 18 km northwest of our study site at 1950 m elevation (www.wcc.nrcs.usda.gov/snow). We recorded the peak snow-water equivalence (hereafter “SWE”) for each year of our study. SWE is the depth of water that would result if the snowpack were melted instantaneously, and is a commonly used metric of snowpack that takes into account both the depth and the density of the snow. We also examined the mean and variance in peak SWE recorded at this site historically (1970-2000).

Field methods

In the spring, we searched all standing water bodies for egg masses one to two times a week for the duration of the breeding season. Egg masses are conspicuous and distinct from one another, and are found attached to emergent vegetation in shallow water along pond shorelines. We determined the total number of egg masses deposited annually across the basin. From 2000 to 2003, we estimated clutch size and variance in a range of egg masses across all breeding ponds using volumetric displacement (Morris and Tanner, 1969, Corn and Livo, 1989, Werner et al., 1999). To calculate the total number of eggs produced annually in each pond, we multiplied the number of egg masses for each pond by an average of 812 eggs/mass, the mean clutch size estimated from our volumetric displacement calculations. Egg mass counts were used to estimate breeding probability, and total egg abundance was used to estimate survival from egg to one year (see below).

We captured frogs in mid-summer following a robust sampling design (Pollock, 1982). We monitored three female life stages: juveniles were frogs that were too young to be sexed; subadults were frogs that could be identified as female (>50 mm with no nuptial pads on the thumbs), but were smaller than the smallest documented breeding female; and adults were frogs large enough to breed (>62 mm). Each year, we systematically surveyed all the ponds and lake shores in the basin and captured animals by hand or net. Juveniles and adults are commonly found basking on rocks or logs near shorelines or swimming in shallow water, and the species is highly aquatic. Animals were individually marked by clipping unique combinations of toes using an alphanumeric coding system (Waichman, 1992), and were measured to determine life stage. To satisfy the assumptions of the robust design, we captured animals for multiple consecutive secondary sessions (days) within the primary sampling period (year). Across the secondary sessions, we assumed that the population was closed to immigration, emigration, births, and deaths. This closure allowed population size, corrected for capture probability, to be estimated. Between primary sessions, we estimated survival and transitions among life stages. From these annual surveys, we had ten-year capture histories for 4362 individuals.

Bayesian methods for parameter estimation

Bayesian methods have become popular for analyzing mark-recapture data. Since the early studies (see Clark et al., 2005 for a literature review), methods have been widely applied and customized for specific
applications (e.g. Royle and Dorazio, 2008). The trends for methodological development have included using hierarchical structure to model heterogeneity, dealing efficiently with missing data, and using covariates to model parameters of interest. We use Bayesian methods for estimating parameters for multistate mark-recapture data (Calvert et al., 2009, Clark et al., 2005). Other authors have used Bayesian methods to estimate movement in open population models (Dupuis and Schwarz, 2007), examine age-dependence in survival (Zheng et al., 2007), model metapopulation dynamics (Royle and Kery, 2007), model individual effects in survival models (Royle, 2008), and estimate population size using data augmentation for both closed and open populations (Royle and Dorazio, 2010).

We developed a software program in MATLAB that implements a Bayesian technique for estimating population vital rates from mark-recapture data with multiple life stages. The Bayesian framework for mark-recapture data allows the probabilistic estimation of vital rate parameters (survival, transition and recapture probabilities). Instead of point estimates, the Bayesian approach gives the solution as a distribution of different possible vital rates that explain the observed data. These distributions are often analytically intractable, but there are numerical methods to approximate the distributions by producing random samples from them. We used a Gibbs sampling algorithm for estimating population vital rates, described in detail here.

We estimated separate survival, transition and recapture probabilities for each year and each life stage. This model is developed for structured populations, where the population can be stratified by life stage at each time step (Dupuis, 1995). We used the following notation:

- $x_{it}$: observed life stage for individual $i$ at year $t$
- $z_{it}$: true life stage for individual $i$ at year $t$
- $S_{it}$: survival probability from year $t$ to year $t+1$ for life stage $i$
- $R_{it}$: recapture probability in year $t$ for life stage $i$
- $P_{ijt}$: transition probability from life stage $j$ at year $t$ to life stage $i$ at year $t+1$

Raw mark-recapture data were entered directly into the model, where two matrices summarized the capture history for a given individual (Dupuis, 1995). Each entry in matrix $x$ was the observed life stage of an individual. If the individual was not observed, we set $x_{it} = 0$. Following Clark et al. (2005), we defined the matrix $z$, which contained the estimated life stage of an individual (which includes both the observed and unobserved life stages). The first matrix consists of raw data, and life stage values for the other matrix are unknown parameters, estimated as part of the Bayesian analysis.

We used the secondary sampling sessions within each summer to estimate the recapture probability for that year. Thus, the likelihood for observing $x$ for a given year $t$, with recapture probabilities $R$ was

$$P(x_{it}, R_{it}) \propto \prod_{i=1}^{4} R_{it}^{o_{it}} (1 - R_{it})^{u_{it}}$$

where the indicator variables $o_{it}$ and $u_{it}$ indicate the number of observed and unobserved individuals at year $t$ and life stage $i$, respectively (counted from all secondary capture sessions). Survival and transition probabilities were estimated between primary capture sessions. The likelihood for life stages $z$ for a given year $t$, with survival and transition probabilities $S$ and $P$ was

$$P(z_{it} | S_{it}) \propto \prod_{i=1}^{4} S_{it}^{s_{it}} (1 - S_{it})^{d_{it}}$$

$$P(z_{it} | P_{ijt}) \propto \prod_{i=1}^{4} \prod_{j=1}^{4} P_{ijt}^{h_{itj}}$$

where indicator variables $s_{it}$, $d_{it}$, and $h_{itj}$, indicate the number of survivals, deaths and transitions at year $t$, respectively (counted between primary sessions).
For estimating the true life stages, we followed Clark et al. (2005) and conditioned the life stage at time \( t \) on life stages at the previous and next time point (surviving previous time, stage transition, not capturing, surviving next time and stage transition):

\[
P(z_{it} = j | z_{i,t-1} = k, z_{i,t+1} = l, R, S, P) \propto S_k P_{jk} (1 - R_j) S_{jl} P_{lj}.
\]  

(4)

Our prior assumptions followed the robust design framework (Pollock, 1982): since secondary capture sessions were very close to each other (1 day), we assumed that survival probability was one between the secondary sessions and no transitions were possible between them. These priors were implicit in our computations, since we estimated survival and transition only between primary capture sessions. For all survival and transition estimates among years, we used uniform priors, spanning all possible probabilities (0 to 1 interval). Thus, from our ten-year capture histories, we estimated posterior probability distributions for annual survival, recapture, and transition probabilities.

The total likelihood for observing data \( x \) and having true life stages \( z \), given the vital rate parameters, was obtained by taking a product of equations (1-3) over years. The posterior distributions for the vital rates can be written down analytically (beta distribution for recapture and survival and Dirichlet distribution for transition). Thus, given our observed data \( x \) and an estimate for life stages \( z \), we sampled new values for \( R, S, \) and \( P \) from these known distributions. In contrast, for some given values of the vital rates, we calculated the probabilities for different \( z_{it} \) using equation (4) and sampled new values from the resulting multinomial distribution. Alternating these conditional samplings, we arrived at the Gibbs sampler (Gelman et al., 2004), which can be summarized by the following four steps:

1. Set the initial values for \( R, S, P \) and \( z \) and iteration counter \( i=1 \)
2. Sample \( R, S \) and \( P \) from their conditional posteriors, given \( x \) and \( z \), using equations (1-3)
3. Loop over individuals and time points where \( x_{it} = 0 \), sample a new life stage \( z_{it} \) using equation (4), given previously sampled \( R, S, \) and \( P \)
4. Set \( i=i+1 \) and go to step 2, until a desired number of iterations is obtained

In addition to the above survival and stage transitions, we estimated two additional terms, breeding probability and survival probability from egg to one year. For the breeding probability, we estimated of the number of females for each year. These were calculated by dividing the number of observed females (counted from the field data) by the recapture probability estimate. To get the estimate for breeding probability as a distribution, this calculation was performed by numerical integration over the distribution of recapture rates given by the Bayesian mark-recapture analysis, i.e., repeating calculations over the distribution of estimates generated by the Gibbs sampler. We then divided the number of female breeding each year.

Similarly, for survival probabilities from egg to one year, we needed estimates of the number of one-year-old frogs. These were calculated from a separate data set, which contains capture histories for one-year-old individuals (identified as frogs 34 mm and smaller). The above Gibbs sampling algorithm was used for estimating the recapture probability for this age class (in this estimation, we only had one life stage), and the number of observed individuals was then divided by different recapture probability estimates to get a distribution of the number of one-year-old frogs. To estimate survival from egg to one year, we then divided the number of one-year-old frogs by the number of eggs laid in the basin the previous spring.

Since the Bayesian sampling approach produces vital rate estimates as a set of samples, we were able to estimate the uncertainty of any function of the vital rates and extend the Bayesian analysis to all calculations made with the estimates. In this paper, we used this property in all of our computations of regressions with vital rates and snowpack and matrix model predictions.

Forecasting the effects of climate change

In all steps of the forecasting process, analyses were repeated over the distribution of estimates generated by the Gibbs sampler: we calculated predictions separately for the 4000 samples. We performed a
logistic regression of annual vital rate parameter distributions with peak SWE measured in that year to determine the relationship between SWE and each vital rate ($Y$) across the ten years of the study, i.e.,

$$\text{logit} (Y_t) = b_{0,Y} + b_{1,Y} \times \text{SWE}_t + \tau_{Y,t}$$  \hspace{1cm} (5)

with $b_{0,Y}$ and $b_{1,Y}$ indicating the regression intercept and slope, $\tau$ indicating residual error and $t$ indexing years.

We then used these relationships to predict vital rate values for various combinations of mean ($\mu$) and variance ($\sigma^2$) in peak SWE. We added process variance to the vital rates predicted using SWE using the residual differences between the model predictions and the observations ($\tau_Y$). The following equation was used to draw values for each vital rate $X$:

$$X = \text{logit}^{-1}(\beta_{0,Y} + \text{SWE} \times \beta_{1,Y} + \epsilon_Y)$$  \hspace{1cm} (6)

where: $\text{SWE} \sim N (\mu, \sigma^2)$ and $\epsilon_Y \sim N (0, \tau_Y)$

For consistency, we used the regressions for all vital rates, even if they did not show a strong relationship with SWE (except clutch size (Cs), where we added normally distributed noise to the mean value of 406 eggs/clutch). We parameterized a 4-stage, female-based projection matrix model with the vital rates predicted from the mean and variance in SWE for each year (Fig. 1). We used these models to estimate stochastic population growth rates ($\lambda_s$) and extinction probabilities under various scenarios where we changed mean and variance in SWE. General definitions and calculations of $\lambda_s$ and extinction probability follow Morris and Doak (2002).

We took two approaches to evaluate the effect of changing mean and variance in SWE on $R. \text{luteoventris}$. First, we examined how periods of historical variation in snowpack recorded at our site affected population viability. Specifically, we compared the mean and variance in SWE from our 10-year study (“current”) to a period from 1983-1995 where both mean and variance in SWE were lower than the current values (Fig. 2). This period was chosen to represent a period with conditions suitable for maintaining a viable population at this site; see Results. We examined four scenarios: (i) current mean and variance, (ii) 1983-1995 mean and variance, (iii) current mean and 1983-1995 variance, and (iv) 1983-1995 mean and current variance. For these scenarios, we forecasted population dynamics for 30 years, and this was repeated 1000 times for different randomly chosen values for snowpack drawn from the given mean and variance. We set ten female frogs as our quasi-extinction threshold. For our starting population size vector, we assumed 100 adult females and the stable stage distribution predicted by the mean matrix (Morris and Doak, 2002, Caswell, 2001).

Second, we examined effects of possible future changes in mean and variance of SWE on population viability. Climate models for the northwestern United States, which include our study site, predict a 10% to 40% decrease in mean peak SWE by 2100, depending on the model and the area within this region (Payne et al., 2004). Therefore we examined scenarios within this range. Although variance in precipitation is expected to change, there are no model predictions for the magnitude of this change, so we picked a range of hypothetical values to explore the effects of changes in this parameter. The mean SWE values encompassed everything from a slight increase in SWE to a 40% decrease in SWE, and the standard deviations cover a 50% reduction in variance to a 3-fold increase in variance. To quantify how viability would change across different combinations of future mean and variance in SWE, we estimated mean $\lambda_s$, extinction probabilities, and the proportion of samples with $\lambda_s \geq 1.0$ for each of 260 different combinations of mean and variance in SWE. Mean SWE ranged from 64 to 112 cm, and standard deviation in SWE ranged from 18 to 48 cm. For each scenario, we ran 4000 model predictions for 30 years, with different possible values for the vital rates given by our Bayesian vital rate estimation. This was repeated 100 times, changing the random value for SWE at each iteration. We used the same extinction threshold and starting population size vector as above.

Finally, we examined the sensitivity of our population viability metrics to changes in mean peak SWE versus changes in the variance in peak SWE. To do this, we calculated the slope of the relationship between population viability metrics and the mean versus variance of SWE over the range of possible values for future mean and variance in SWE.
Results

Vital rate estimates and climate variables

Peak SWE values at this site decreased slightly over the past 40 years, though confidence limits for the slope span zero (slope = -0.16 [-0.43 to 0.12 CI]; Fig. 2). Variability among years was high. Both mean (90 cm) and standard deviation (10 cm) in peak SWE were notably lower in the period from 1983-1995 than over the 1969-2009 period (mean = 106 cm, standard deviation = 27 cm) or over the years of our study (mean = 103 cm, standard deviation = 23 cm; Fig. 2).

Breeding probability and first year survival were extremely variable across the ten years of our study, and showed no relationship with peak SWE. Breeding probability ranged from 0.38 to 1.0, and there was considerable uncertainty in many of the estimates. First year survival ranged from 0.001 to 0.009. Notably, first year survival was the highest and most variable in one of the lowest snow years and in the highest snow year.

There was an inverse relationship between juvenile and adult survival and peak SWE (Fig. 3). There was no relationship between subadult survival and SWE. Transition probabilities from juvenile to subadult stage classes had a negative relationship with SWE, but there were no relationships between other transitions and SWE. These results are consistent with our previous, deterministic analysis of *R. lutetianus* vital rates in relation to climate variables (McCaffery and Maxell, 2010).

Climate scenarios

When we compared scenarios using the period from 1983-1995 to our current conditions (2000-2009), we found that the earlier period had a mean $\lambda$ of 1.01 (0.95-1.05 credible interval; hereafter "CI") with 68% probability of $\lambda$ being greater than one (probability of growth), while the current conditions had a mean $\lambda$ of 0.90 (0.79-0.99 CI) and a 0.95% probability of growth (Fig. 4). Thus, the earlier period generally predicted a growing population while the current period predicted a declining population, which was consistent with our observation of a declining population from 2000-2009 (McCaffery and Maxell 2010). Combining lower variance (1983-1995) with current mean SWE resulted in a slight increase in $\lambda$ to 0.92 (0.82-0.99 CI), but the population was still most likely declining (99% posterior density <1.0). When we combined high variance in SWE (as seen in 2000-2009) with the lower mean SWE (1983-1995), $\lambda$ increased to 0.99 (0.92-1.05 CI), with 34% probability of growth. Relative to 1983-1995 mean and variance, this scenario had lower mean $\lambda$ and a 34% lower probability of population growth, which illustrates how increased variance can influence viability. Still, overall, changing mean SWE had a greater effect than changing variance in SWE. The mean difference in $\lambda$ caused by changing mean SWE from earlier to current conditions was 0.09 (0.05-0.13 CI), while the mean difference in $\lambda$ caused by changing variance in SWE was 0.02 (0.0-0.04 CI).

Across all of our scenarios for future changes in the mean and variance of SWE, a decrease in mean SWE had a positive effect on population viability, and an increase in variability had a small negative effect (Fig. 5). Decreases in mean SWE resulted in an increase in mean $\lambda$, an increase in the proportion of samples with $\lambda > 1.0$, and a decrease in percent of runs going extinct. Increases in variance of SWE resulted in a decrease in mean $\lambda$, a decrease in the proportion of samples with $\lambda \geq 1.0$, and an increase in the percent of runs going extinct. Populations mostly changed from decreasing to increasing around a mean 84 cm SWE, which represents a 20% decrease relative to current conditions. Mean $\lambda$ had a higher sensitivity to changes in mean SWE than to changes in variance (Fig. 6). All 4000 slopes for the relationship between decreases in mean SWE and increases in $\lambda$ (-0.014, -0.02 to -0.008 CI) were greater than the slopes for the relationship between increases in the variance of SWE and decreases in $\lambda$ (-0.004, -0.007 to 0.0 CI). Similarly, the slopes for the relationship between decreases in mean SWE and decreases in the percent of runs going extinct (0.045, 0.002 to 0.064 CI) were generally greater than the slopes for the relationship between increases in the variance of SWE and the increase in percentage of runs going extinct (0.018, 0.001 to 0.025 CI).
Discussion

Interannual climatic variability is predicted to increase along with changes in mean climate conditions (Easterling et al., 2000), but the impacts of changing mean and variance in climate variables has rarely been examined in conjunction with animal population viability (but see van de Pol et al., 2010). For this population of *R. lutheiventriss* in a mountain ecosystem, we found that population viability metrics were more sensitive to changes in mean SWE than to changes in the variance of SWE. Decreases in SWE systematically increased mean λ, increased the probability of population growth, and decreased extinction probabilities; increases in variability had less of an effect. This conclusion contrasts with a previous meta-analysis of population time series data, in which environmental stochasticity was the most important determinant of extinction risk (Fagan et al., 2001). Indeed, population ecological theory has generally stated that environmental stochasticity decreases population viability (Lande et al., 2003). However, recent research examining the effects of mean versus variability in climatic change on wild populations has corroborated our results. Colchero et al. (2009) found that an increase in mean drought severity had a larger effect on the viability of a population of desert bighorn sheep (*Ovis canadensis*) than an increase in the variability in drought. Similarly, Jonzen et al. (2010) found that changes in average rainfall had a greater impact on population growth rates of the red kangaroo (*Macropus rufus*) than increased variability in rainfall. Finally, van de Pol et al. (2010) concluded that changes in mean winter temperature were more important to Eurasian Oystercatcher (*Haematopus ostralegus*) population viability that changes in variance in temperature. These studies, along with our own, add empirical evidence to a growing body of literature suggesting that increased environmental stochasticity may not always negatively impact population viability, and may actually improve viability in some cases (Drake, 2005, Higgins et al., 2000, Boyce et al., 2006).

Longevity may affect the resiliency of this population of *R. lutheiventriss* to increased environmental stochasticity. Morris et al. (2008) demonstrated that in general, long-lived species are more resilient to increases in climatic variability than short-lived species (insects and annual plants). Columbia spotted frogs may live up to ten or more years old in our system, and do not become sexually mature until 3 to 5 years of age at high elevation (Werner et al., 2004). Thus, it may be that for this long-lived frog species, populations can experience substantial environmental variation without adverse effect.

Heterogeneity in the habitat used by *R. lutheiventriss* in this system may also contribute to the relative unimportance of environmental stochasticity to population viability. This population uses a network of 12 lakes, permanent ponds, and ephemeral ponds for breeding, foraging, and overwintering. This heterogeneous habitat provides consistent breeding and foraging habitat on both wet and dry years (McCaffrey, unpubl. data), as has been demonstrated for other amphibian species inhabiting heterogeneous landscapes (Karraker and Gibbs, 2009). Pond-breeding temperate amphibian species are typically characterized as having extremely stochastic population dynamics (Green, 2003), which are thought to be largely driven by stochastic environmental conditions (e.g. Semlitsch et al., 1996, Church et al., 2007). However, these studies have typically examined single pond systems, where animals much more constrained in years where the pond habitat is unsuitable. Amphibians inhabiting simple or isolated pond habitats may be more affected by environmental variability.

The shape of the relationship between SWE and population viability metrics may also affect the extent to which variability is important for population viability. In our study, decreases in mean SWE resulted in an approximately linear increase in mean λ (Figure 5a). However, for non-linear relationships, changes in environmental variability may have greater consequences for population viability (Drake, 2005, Ruel and Ayres, 1999). If the relationship between SWE and vital rates were convex (positive 2nd derivative), variance in vital rates would be greater than variance in SWE. If the relationship between SWE and vital rates were concave (negative 2nd derivative), then variance in vital rates would be less than variance in SWE (Jensen, 1906, see discussion by Ruel & Ayres, 1999). This relationship suggests that, in principle, the shape of the relationship between environmental variation and population parameters could be an important indicator of the relative importance of changes in average vs. variance of environmental variables. In fact, four studies that have investigated these relationships have shown all three functional forms: concave (Jonzen et al., 2010), convex (Colchero et al., 2009), both concave (for survival) and convex (for fecundity; van de Pol et al. 2010), and linear (this study), but all conclude that changes in the mean affect population viability more than changes in variance. Therefore, the general conclusion that changes in mean conditions are more important than changes in variance
do not seem to depend critically on the shape of climate-vital rate relationships in practice, at least for long-lived species.

The influence of environmental stochasticity cannot wholly be discounted, however. We found that increases in variability generally decreased stochastic lambda, and could change populations from increasing to decreasing (Fig. 5). Similarly, in a meta-analysis of 15 plant and 21 animal taxa, Morris et al. (2008) emphasized that increased vital rate variability decreased long-term population growth rates. Although the relative importance of changes in variance may be less than changes in the mean, increases in variability can have consequences for population viability. The patterns of variability in climate over time may also influence the importance of including variability in population models, as highlighted by our comparison of population viability using SWE values from 1983-1995 and 2000-2009. The SWE values from 1983-1995 predicted a growing population, while the SWE values for the current time frame predicted a declining population. Mean and variance values from the 40-year record of SWE at this site would also predict a declining population, yet the population is persisting at this site. Thus, short periods of favorable conditions may promote viability even within longer-term predicted declines.

Our Bayesian analysis also demonstrates that decreases in mean snowpack, as projected by climate models for the region (Payne et al., 2004), promoted population viability of *R. luteiventris* in this system, and that these results are robust to uncertainty in our vital rate estimates, as well as increasing variation in snowpack. Population growth rates typically change from decreasing to increasing with a 20% decline in snowpack, which is well within the predicted range of decline for this area (Payne et al., 2004). This result challenges the overarching dogma that climate change will generally be negative for amphibians, or exacerbate declines (Corn, 2005, McMenamin et al., 2008, Wake, 2007). Rather, this case study offers the more nuanced perspective that in some circumstances, climate warming could benefit some populations and species, provided that suitable habitats remain intact. Frogs in this system persist at the upper elevational limit of their range, and endure long cold winters on many years. Populations of *R. luteiventris* inhabiting these elevations have broad thermal tolerances, and fitness may be enhanced by a warming climate if they are currently spending much of their time below their physiological optimum (Deutsch et al., 2008). Our results indicate that *R. luteiventris* and similar species inhabiting harsh alpine environments may be unlikely beneficiaries of climate change, provided that habitat suitability and prey availability remain unaffected. Similarly, van de Pol et al. (2010) found that for a long-lived shorebird, increases in mean winter temperatures were predicted to increase population viability via an increase in adult survival, even though these temperature changes were predicted to negatively affect fecundity.

Peak snowpack only explains some of variability in our vital rates, and only relates to some vital rates (notably, juvenile and adult survival). Undoubtedly other intrinsic and extrinsic factors are influencing these rates. For example, timing of spring snowmelt is related to breeding probability in this population (McCaffery & Maxell, 2010). Density dependence at either the larval stage or the adult stage could influence vital rates over time (Harper and Semlitsch, 2007, Semlitsch, 1987, Wilbur, 1976, Berven, 1990), although densities of larvae and frogs at this site are lower than what has been measured in other species. Vital rate variability might also be influenced by current predation rates and resource availability. Finally, there could be potential effects of toe-clipping in survival probabilities over time, as has been noted in other species (McCarthy and Parris, 2004). Furthermore, our models predict the effects of future snowpack decline on *R. luteiventris* population viability in isolation from other climate-related ecosystem changes that could be occurring alongside declines in snowpack. These may include habitat loss and changes in the abundance and distribution of food resources. Decreases in the total availability of breeding habitat due to drought could have negative impacts on the population (e.g., McMenamin et al., 2008), and may be accompanied by reductions in the quality of remaining habitat. Reductions in snowpack may be accompanied by changes in the timing, quantity or quality of food resources, which may reduce body conditions and survival rates in juvenile and adult frogs. Thus, models that simultaneously evaluate different ways a given population or species may be impacted by climate and which include other potential drivers of variation could help disentangle the relative influences of different results of changes in climate on population dynamics and viability. However, even with these potentially confounding factors, changes in snowpack do appear to contribute to population viability.

Our results are based on analyses of one population, and we might expect other populations, even at high elevations, to have different relationships with snowpack. In the alpine ibex (*Capra ibex*), spatial covariation in climate variables did not synchronize population dynamics over large spatial scales, suggesting that the effects of
climate variables on populations are heterogeneous over a broader spatial landscape (Grotan et al., 2008). In amphibians, configuration of breeding habitat (Karraker and Gibbs, 2009), presence of non-native species (Pope, 2008, Knapp and Matthews, 2000), and presence of disease (Scherer et al., 2005) may all influence the way climate changes affect population viability. For example, populations dependent on a single, ephemeral breeding pond may be more highly impacted by recruitment failures due to drought than changes in juvenile and adult survival, if the relative magnitude of recruitment failures is greater than increases in survival. Amphibian populations whose breeding is limited to a few fishless ponds may experience greater habitat constrictions with climate change than populations where fish have not been introduced. Finally, if climate change makes conditions more favorable for disease, then increases in survival due to milder winters may be offset by decreases in survival due to disease. Conducting similar analyses and forecasts on other populations and species will help determine the extent to which the patterns we observed are generalizable across a broader landscape.

In this study, we demonstrated a case where mean climate conditions were more important to population viability than variance in those conditions. These projected changes in climate predicted overall increases in population viability for a dynamic frog population. These results add to a small, but growing, body of literature examining the relative effect of changing mean versus variance of climatic variables on the viability of animal populations, and demonstrate the power of demographic modeling for synthesizing the effect of climate variables on multiple vital rates. The use of Bayesian methods in this analysis allowed for an explicit characterization of uncertainty in our parameter estimates and our forecasts of future population viability. The relative unimportance of increased environmental stochasticity on population viability was surprising, since it has been considered the most important determinant of population viability (Fagan et al., 2001). However, increased climatic variation can change populations from increasing to decreasing, and should not be discounted from analyses of the effects of climate change on wild populations.

Acknowledgements

We thank Bryce Maxell, who initiated the field component of this study and collected half of the demographic data used in this paper. We also thank the many field volunteers who have assisted with data collection over the years. We thank M. Laine for help with coding the Gibbs sampler, and J. Brodie, L. Eby, M. Ellis, C. Hartway, and W. Lowe for constructive comments on drafts of the manuscript. A.S. received funding from the Fulbright Program to conduct research at the University of Montana. Fieldwork was partially supported by funding from the U.S. Geological Survey’s Amphibian Research and Monitoring Initiative and U.S. Forest Service Region 1 Inventory and Monitoring grants.
References


Figure legends

Fig. 1. Female-based, post birth pulse matrix projection model for *R. luteiventris*. Vital rates include survival rates (S) for each life stage (first year = 0, juvenile = J, subadult = S, adult = F), transition probabilities (P) among life stages, probability of breeding (P_b), and clutch size (Cs).

Fig. 2. Peak SWE recorded at the Twin Lakes SNOTEL site from 1970-2009. Underlined periods are used to contrast a period with low mean and variance in SWE with the conditions observed over the course of our study (‘current period’).

Fig. 3. Annual posterior estimates (with 95% credible intervals) for juvenile, subadult, and adult female survival in relation to SWE. The line shows predicted values for survival for different values of SWE, with 50% (dark gray) and 95% (light gray) confidence envelopes.

Fig. 4. Posterior probability density estimate for stochastic lambda under 4 scenarios, comparing effects of changes in both mean and variance of peak SWE in two historical periods: (i) 2000-2009 mean and variance of peak SWE (current mean & var); (ii) current peak SWE and 1983-1995 variance in peak SWE (current mean, 83-95 var); (iii) 1983-1995 mean in peak SWE and current variance (83-95 mean, current var); and (iv) 1983-1995 mean and variance in peak SWE (83-95 mean & var).

Fig. 5. Contour plots showing the effect of changes in mean SWE and standard deviation of SWE on (a) mean stochastic λ, (b) proportion of samples with λ > 1.0, and (c) percent of runs going extinct.

Fig. 6. Distribution of regression coefficients demonstrating the effect of changing (i) mean SWE and (ii) variance in SWE on mean stochastic λ.
\[
\begin{bmatrix}
0 & 0 & P_{b}^*S_{s}^*C_{s} & P_{b}^*S_{f}^*C_{s} \\
S_{i} & P_{j}^*S_{j} & 0 & 0 \\
0 & P_{j}^*S_{j} & P_{s}^*S_{s} & 0 \\
0 & P_{j}^*S_{j} & P_{f}^*S_{f} & S_{f}
\end{bmatrix}
\]

Fig. 1

---

![Graph showing Peak SWE (cm) from 1965 to 2010 with two periods marked: 1983-1995 and Current period.](image)

Fig. 2
Fig. 4
Fig. 5
Fig. 6
Estimation of ECHAM5 climate model closure parameters with adaptive MCMC


© Authors (CC Attribution 3.0 Licence). Reprinted, with permission.
Estimation of ECHAM5 climate model closure parameters with adaptive MCMC

H. Järvinen\textsuperscript{1}, P. Räisänen\textsuperscript{1}, M. Laine\textsuperscript{1}, J. Tamminen\textsuperscript{1}, A. Ilin\textsuperscript{2}, E. Oja\textsuperscript{2}, A. Solonen\textsuperscript{3}, and H. Haario\textsuperscript{3}

\textsuperscript{1}Finnish Meteorological Institute, Helsinki, Finland
\textsuperscript{2}Aalto University School of Science and Technology, Espoo, Finland
\textsuperscript{3}Lappeenranta University of Technology, Lappeenranta, Finland

Received: 26 March 2010 – Published in Atmos. Chem. Phys. Discuss.: 6 May 2010
Revised: 8 October 2010 – Accepted: 11 October 2010 – Published: 25 October 2010

Abstract. Climate models contain closure parameters to which the model climate is sensitive. These parameters appear in physical parameterization schemes where some unresolved variables are expressed by predefined parameters rather than being explicitly modeled. Currently, best expert knowledge is used to define the optimal closure parameter values, based on observations, process studies, large eddy simulations, etc. Here, parameter estimation, based on the adaptive Markov chain Monte Carlo (MCMC) method, is applied for estimation of joint posterior probability density of a small number ($n = 4$) of closure parameters appearing in the ECHAM5 climate model. The parameters considered are related to clouds and precipitation and they are sampled by an adaptive random walk process of the MCMC. The parameter probability densities are estimated simultaneously for all parameters, subject to an objective function. Five alternative formulations of the objective function are tested, all related to the net radiative flux at the top of the atmosphere. Conclusions of the closure parameter estimation tests with a low-resolution ECHAM5 climate model indicate that (i) adaptive MCMC is a viable option for parameter estimation in large-scale computational models, and (ii) choice of the objective function is crucial for the identifiability of the parameter distributions.

1 Introduction

Atmospheric general circulation models (GCMs) consist of dynamical laws of atmospheric motions and physical parameterizations of sub-grid scale processes, such as cloud formation and boundary layer turbulence. Specified parameters appear in physical parameterization schemes where some unresolved variables are expressed by predefined parameters rather than being explicitly modeled. These are called closure parameters. A simple example of such a parameter is provided by turbulent transfer in the atmosphere. In a first order closure, the transfer of a quantity $q$ is assumed to be proportional to the gradient of $q$ multiplied by a fixed diffusion coefficient – note that a whole hierarchy of closures of different orders exists, each with different closure parameters (Mellor and Yamada, 1974). Another example is cloud shortwave optical properties which depend on cloud optical thickness. This can be related to resolved cloud liquid water amount via the mean effective radius of cloud water droplets. If the cloud micro-physics is not resolved, the mean effective radius has to be prescribed (Martin et al., 1994). The modelled shortwave radiation flux is sensitive to the specified value of this parameter, and it can act as an effective "tuning handle" of the simulated climate.

An underlying principle in climate model development is to aim at few rather than many closure parameters. In the model development process, best expert knowledge is used to define the optimal parameter values. They can be constrained to some degree based on observations, process studies, large eddy simulations, etc, but they do not necessarily represent any directly observable quantity. Additionally, parameter values can depend on the discretization details, such as grid interval or choices made regarding modeling of other physical processes. This is a dilemma since observations do not provide guidance towards resolution or modeling environment dependent parameter values. In summary, the closure parameters are determined such that (i) they are consistent with prior knowledge, and (ii) simulations prove to be realistic in posterior validation. In fact, both can be used in an iterative manner to optimize model performance.

The closure parameters of atmospheric general circulation models are, by definition, constant during the model run. Therefore they should perform well independent of particular weather situations, both locally and in a global sense.
Various approaches are available for solving the closure parameter estimation problem. First, the review paper of Navon (1993) concentrates on adjoint techniques (e.g., Rinne and Järvinen, 1993) and stresses the questions of parameter identifiability and stability. This implies that both the estimation method and the parameters to be estimated need to be selected carefully. Sequential state estimation in numerical weather prediction aims at fitting the initial condition and model parameters to prior information and to observations (e.g., Dee, 2005). Only the maximum-likelihood fit and a Gaussian error covariance are obtained from solving the tangent-linear analysis equation. If closure parameters are estimated in this framework, their values partly reflect the latest observations – this is in fact in slight contradiction to the notion that the closure parameter distributions should be stationary.

Annan and Hargreaves (2007) provide a review of the available parameter estimation methods in climate modelling. They also discuss the Markov chain Monte Carlo (MCMC) method and consider it too computationally expensive for estimating climate model closure parameters. Their treatment of MCMC is, however, somewhat restricted to the Metropolis algorithm (Metropolis et al., 1953), and recent advances in adaptive methods are not fully covered.

Jackson et al. (2008) used a stochastic optimization method, Multiple Very Fast Simulated Annealing (MVFSA), for the search of optimal closure parameters of the CAM3 climate model, with respect to a cost function based on multi-source observations. The search paths of the stochastic optimizer were used to infer about the parametric uncertainty. Villagran et al. (2008) further evaluated the performance of MVFSA and compared it against different MCMC methods with a surrogate climate model. These methods included Adaptive Metropolis (AM), the Single Component Adaptive Metropolis (SCAM) and the Delayed Rejection Adaptive Metropolis (DRAM), see (Haario et al., 2001, 2004, 2005, 2006; Andrieu and Moulines, 2006). Villagran et al. (2008) note that MVFSA may be efficient for optimization, but statistical inference about the parameter distribution tends to be biased. The results were in favor of DRAM and SCAM, especially in test cases with short sampling chains.

In this article, we demonstrate the use of MCMC in the context of the atmospheric general circulation model ECHAM5. A central outcome is that it is, indeed, viable to use MCMC for parameter estimation for a climate model, at least in the context of a coarse-resolution atmospheric GCM. Research methods are presented in Sect. 2, experimental setup and results in Sects. 3 and 4, and discussion and conclusions in Sects. 5 and 6.
other hand, it should be constructed such that the parameter accurate climate simulations from inaccurate ones. On the one hand, the objective function should construct an objective function which would replace this human element. On one hand, the objective function should be physically justified, i.e., being capable of distinguishing accurate climate simulations from inaccurate ones. On the other hand, it should be constructed such that the parameter distributions are identifiable with respect to the chosen objective function. If this is the case, the parameter posterior probability distribution should be compact and limited. If not, either the objective function does not provide the desired guidance for the parameters, or they are simply not relevant in tuning the model with respect to the objective function.

Five alternative formulations of the objective function are tested, all of which are related to the net radiative flux at the TOA in the ECHAM5 model \( \langle F \rangle \) and in CERES EBAF data \( \langle F^o \rangle \). Annual and monthly mean fluxes are denoted by \( \langle F \rangle \) and \( \langle F^o \rangle \), and global and zonal means by \( \langle F \rangle \) and \( \langle F^o \rangle \), respectively. Subscripts \( x \) and \( y \) refer to geographical location in zonal and meridional direction, and \( t \) refers to time (in months). The first of the five alternative formulations of the objective function is denoted by \( J^G(\theta) \), and it uses only the global–annual mean value of \( F \):

\[
J^G(\theta) = \left( \frac{\langle F \rangle - \langle F^o \rangle}{\sigma_o^2} \right)^2,
\]

where \( \theta \) is the vector of four closure parameters. It penalizes climate simulations which deviate from the global annual-mean net radiative flux in CERES EBAF data (0.9 Wm\(^{-2}\)). The squared net flux difference is normalized by the standard deviation \( \sigma_o^2 \) of the inter-annual variability of the global annual mean net flux, which is estimated from ERA-40 data (0.53 Wm\(^{-2}\)).

The second formulation is denoted by \( J^{XY}(\theta) \)

\[
J^{XY}(\theta) = \frac{1}{12} \sum_{t=1}^{12} \sum_{x} \sum_{y} w_{x,y} \left( \frac{\langle F_{x,y,t} \rangle - \langle F^o_{x,y,t} \rangle}{\sigma_{o_x,y,t}^2} \right)^2
\]

It accounts for local differences in monthly mean net fluxes. The weights \( w_{x,y} \) represent grid point area fractions. The squared net flux difference is normalized by the standard deviation of the inter-annual variability of the local monthly mean net fluxes, based on ERA-40 data. The third formulation, denoted by \( J^{ZONAL}(\theta) \), uses zonal mean values of monthly mean net fluxes:

\[
J^{ZONAL}(\theta) = \frac{1}{12} \sum_{t=1}^{12} \sum_{y} w_{y} \left( \frac{\langle F_{y} \rangle - \langle F^o_{y} \rangle}{\sigma_{o_y}^2} \right)^2
\]

Here, the weights \( w_{y} \) represent area fractions for the zonal bands, and the normalizing factor is the standard deviation of the inter-annual variability in monthly and zonal mean net fluxes.

The last two formulations

\[
J^{G+XY}(\theta) = J^G(\theta) + J^{XY}(\theta)
\]

\[
J^{G+ZONAL}(\theta) = J^G(\theta) + J^{ZONAL}(\theta)
\]

are combinations of the objective function Eq. (1) with Eqs. (2) and (3), respectively. Equations (4) and (5) attempt to emphasize the weight of the global annual mean net flux in addition to the regional details in net radiative fluxes.

**Table 1.** The considered sub-set of ECHAM5 closure parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAULOC</td>
<td>A parameter influencing the accretion of cloud droplets by precipitation (rain formation in stratiform clouds)</td>
</tr>
<tr>
<td>CMFCTOP</td>
<td>Relative cloud mass flux at the level above non-buoyancy (in cumulus mass flux scheme)</td>
</tr>
<tr>
<td>CPRCON</td>
<td>A coefficient for determining conversion from cloud water to rain (in convective clouds)</td>
</tr>
<tr>
<td>ENTRSCV</td>
<td>Entrainment rate for shallow convection</td>
</tr>
</tbody>
</table>

A semi-implicit time integration scheme is used for model dynamics with a time step of 40 min. Model physical parameterizations (see Roeckner et al., 2006) are invoked every time step with the exception of radiation, which is computed once in two hours.

Four ECHAM5 closure parameters were considered (Table 1). These parameters are related to physical parameterizations of clouds and precipitation. The choice of these parameters is motivated by their substantial influence on model cloud fields and therefore the radiative fluxes at the top of the atmosphere (TOA). It is thus plausible that they can be constrained by a suitable formulation of the objective function.

### 2.3 Observational data sets

In this initial study, the definition of the objective function is based solely on the net (longwave + shortwave) radiative flux at the TOA. The observational estimates are taken from the Clouds and the Earth’s Radiant Energy System (CERES) Energy Balanced and Filled (EBAF) dataset (Loeb et al., 2009). The CERES EBAF dataset is, however, only used for the mean values. Since this dataset contains data for five years once in two hours, we chose to derive the interannual standard deviations from the the European Centre for Medium-Range Weather Forecasts (ECMWF) reanalysis data (ERA-40; Uppala et al., 2005), which provides a much longer (44-year) timeseries.
Table 2. Parameter values applied in the MCMC tests. The first column gives the default values for resolution T21L19, the second column the initial estimate of one-sigma uncertainty used to initialize the MCMC chain, the third column minimum and maximum parameter values allowed, and the fourth column the range of parameter values applied in standard ECHAM5. The upper limit of parameter CAULOC was 100 for the first four experiments, but it was reduced to 30 for the $J^{GZONAL}$ experiment, when it was realized that all values of CAULOC higher than about 30 produce identical results.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default value</th>
<th>Initial std.dev.</th>
<th>Range in MCMC tests</th>
<th>Range in ECHAM5 (other model resolutions etc.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAULOC</td>
<td>1</td>
<td>1</td>
<td>0–30/100</td>
<td>1–5</td>
</tr>
<tr>
<td>CMFCTOP</td>
<td>0.10</td>
<td>0.08</td>
<td>0–1</td>
<td>0.10–0.35</td>
</tr>
<tr>
<td>CPRCON</td>
<td>$8 \times 10^{-4}$</td>
<td>$4 \times 10^{-4}$</td>
<td>0–1.5 $\times 10^{-2}$</td>
<td>$1 \times 10^{-4}$–$10^{-3}$</td>
</tr>
<tr>
<td>ENTRSCV</td>
<td>$3 \times 10^{-4}$</td>
<td>$3 \times 10^{-4}$</td>
<td>0–5 $\times 10^{-3}$</td>
<td>$3 \times 10^{-4}$–$10^{-3}$</td>
</tr>
</tbody>
</table>

3 Experimental setup

Five separate experiments were performed, one for each of the five objective functions listed above (Eqs. 1–5). An MCMC chain consisting of 1000 model runs was applied, with one exception: a longer chain of 4500 runs was carried out for the experiment using the $J^{GZONAL}$ objective function (Eq. 5). Each model evaluation represents a one-year climate simulation with the low-resolution ECHAM5 model. Prescribed distributions of sea surface temperature and sea ice for year 1990 were used (AMIP Project Office, 1996), and the model initial condition was 1 January 1990. One simulation step took about 17 min using 30 CPUs on a Cray XT5m computer.

Default parameter values and prior distributions (or ranges) applied in the experiments are provided in Table 2. The MCMC algorithm was broadly as follows:

**Step 0:** Initialize the four closure parameterstheir default values; Initialize proposal distribution to reflect the a priori knowledge about parameter uncertainty; Run the model for one year; Post-process the model data and evaluate the objective function.

**Step 1:** Draw new parameters from the proposal distribution centered at the current parameter values; Run the model with new parameter values and evaluate the objective function.

**Step 2:** Accept or reject new parameter values based on the difference of objective functions at current vs. previous step; Update the proposal distribution according to the adaptive MCMC algorithm.

**Step 3:** Return to Step 1 if the chain has not yet been completed.

Note that the difficulty in providing a correct initial proposal covariance in Step 0 makes the adaptation method applied in Step 2 crucial for the sampling to be efficient.

4 Results

The MCMC tests with the low-resolution ECHAM5 climate model are discussed in the next four subsections, with emphasis on general aspects of the results.

4.1 Parameter chains

The random walk process is started in each experiment from the default parameter values (Table 2). The parameter values for the subsequent runs depend on the definition of the objective function. We illustrate this by showing the MCMC chains for the five different objective functions and for two parameters with contrasting behaviour: CMFCTOP (Fig. 1) and CPRCON (Fig. 2). In Figs. 1 and 2, the dots represent parameter values included in the MCMC chain. If a parameter combination is rejected by the MCMC algorithm, the previous accepted parameter values are repeated. The horizontal grey line represents the default parameter value, 0.1 for CMFCTOP and $8 \times 10^{-4}$ for CPRCON, respectively. Note that in Fig. 1, the scale is different in different panels.

For CMFCTOP (Fig. 1), the parameter values are generally well-bounded from above. Only for $J^{XY}$ the constraint on CMFCTOP is somewhat weak, the largest accepted parameter values approaching the upper limit of physically meaningful values (CMFCTOP=1). For $J^{G}$, the accepted parameter values vary on both sides of the default value, while for the three remaining cost functions $J^{ZONAL}$, $J^{G+XY}$ and $J^{GZONAL}$, there is a clear tendency for parameter values smaller than the default. Overall, CMFCTOP is an example of a parameter which behaves quite in an expected way.

The MCMC chains for the parameter CPRCON (Fig. 2) behave rather differently from those for CMFCTOP (Fig. 1). Generally, the values of CPRCON are weakly bounded from above for all formulations of the objective function – sooner ($J^{X}$) or later ($J^{G}$) the upper limit of the prior range of...
Fig. 1. The MCMC chain for parameter CMFCTOP. The horizontal grey line at parameter value 0.1 is the default value. Note the different scaling in different panels. Also note that the length of the MCMC chains is slightly larger than the number of ECHAM5 runs performed (1000 for the first four chains, 4500 for the \( JG + \) ZONAL chain).

parameter values is met. There seems to be a tendency towards parameter values larger than the default. Figure 2 is an example of a parameter which is weakly constrained by any of the objective functions, and the overall behaviour is not very desirable. A possible explanation is that for changes in CPRCON, the corresponding changes in longwave and shortwave fluxes at the TOA tend to cancel each other, leading to smaller changes in the TOA net flux.

In general, Figs. 1 and 2 point to the importance of the choice of the objective function. This is seen both in the fact that some objective functions constrain CMFCTOP much more tightly than others (e.g., compare \( JG + \) ZONAL with \( JX + \) ZONAL) and in the failure of all objective functions to constrain CPRCON properly. The latter point calls for an improved definition of the objective function in future work. Specifically for CPRCON, compensation between longwave and shortwave fluxes suggests that an objective function that utilizes these fluxes separately, rather than only the net flux, might better constrain this parameter. Other model fields sensitive to CPRCON include precipitation rate, and middle and high cloud fractions.

Another general point evident in particular in Fig. 2 concerns the convergence of the MCMC chains. For example, when using the objective function \( JG \), the values of CPRCON remain relatively close to the default value nearly until the end of the chain, but then rapidly drift to a much higher level. Had we stopped the chain after (e.g.) 800 runs, we could have falsely concluded that \( JG \) constrains CPRCON rather well. This suggests that a MCMC chain length of 1000 runs is not long enough for analyzing the posterior parameter distributions properly. The issue of chain convergence is addressed more comprehensively in the following section, based on the \( JG + \) ZONAL experiment.

4.2 Analysis of the \( JG + \) ZONAL experiment

To visualize how the statistical characteristics of the MCMC parameter chains evolve in the \( JG + \) ZONAL experiment, we
evaluated the posterior distributions after each run, always using the last 500 runs only. The blue, black and red dotted lines in Fig. 3 show the 10th, 50th and 90th percentage points computed from these distributions for each of the four parameters. The solid horizontal lines show the respective percentage points computed from the last 3500 runs (i.e., runs 1001–4500). While the parameters start from their default values, all of them experience a drift in the early part of the chain, so that the distributions of CAULOC, CPRCON and ENTRSCV drift towards higher values and that of CMFCTOP towards lower values. By visual inspection of Fig. 3, the drift lasts for roughly 500–1000 runs, CPRCON and ENTRSCV stabilizing slightly earlier than CAULOC and CMFCTOP. For all parameters, percentage points computed from runs 1001–1500 are close to those computed from runs 1001–4500. This suggests that for the analysis of the parameter posterior distributions, it is sufficient to omit the first 1000 runs.

Figure 4 displays posterior distributions of the four parameters computed from runs 1001 through 4500. As shown before (Fig. 1), for CMFCTOP there is a strong preference for values smaller than default (0.1). Also, as suggested by Fig. 2, CPRCON is poorly constrained from above. In broad terms, all values are deemed equally likely by the MCMC algorithm, except that the very lowest values are ruled out. CAULOC behaves quite similarly to CPRCON. Finally, the posterior distribution for ENTRSCV features a broad maximum centred around ENTRSCV ≈ 1.5 × 10^{-3}. Both the lowest values and very high values of ENTRSCV are deemed unlikely. The 2-dimensional marginal posteriors (Fig. 5) do not display significantly large correlations between the parameters and further reveal the rather poor identifiability of CAULOC and CPRCON. In summary, when using the objective function $J_{G+ZONAL}$, the MCMC algorithm is able to provide some constraints on all four parameters considered, although for CAULOC and CPRCON, the constraints are rather weak.

4.3 Objective function versus radiative fluxes

Trivially, parameter retuning by the MCMC process can improve (i.e., decrease) the value of the objective function compared to its value for default parameter settings. A crucial question is, however, whether the MCMC process helps to reduce errors in those quantities not explicitly included in the objective function. A simple test illustrated in Fig. 6 indicates that this is, again, dependent on the choice of the objective function.

Figure 6 displays the five different objective functions versus global annual mean net, longwave (LW) and shortwave (SW) radiative fluxes at the TOA – recall that only the net flux, rather than LW and SW fluxes separately, is used in the objective functions (1)–(5). The vertical grey line represents the observed global annual mean fluxes from CERES EBAF data, and the grey dot corresponds to the default parameter values. For $J^G$ (Fig. 6, panels a–c), the cloud of points of the MCMC chain is exactly parabolic for net radiation, as $J^G$ penalizes of squared differences in global annual mean net radiation. The default parameter values correspond quite closely to the objective function minimum. Obviously, this has been used as a criterion in the ECHAM5 model tuning.
For $J_G$, the default parameter values correspond to LW and SW biases of 7–8 W m$^{-2}$. It is possible to select parameter values for an unbiased model in net radiation which correspond to LW and SW biases in the interval of about 3 to 20 W m$^{-2}$, but not smaller. In particular, an overestimate of the (down-up) LW radiation at the TOA compared to CERES EBAF data seems to be an inherent bias of ECHAM5 at T21 resolution.

For $J^{X_Y}$ (Fig. 6d–f), the cloud of points of the MCMC chain is diffuse and weakly parabolic for net radiation, and $J^{X_Y}$ varies rather little from one MCMC step to another. There is a strong tendency for a positive net flux bias. Thus, minimization of errors in the geographical distribution of the monthly net flux is not a sufficiently strong constraint for obtaining correct global annual net flux. Note, however, that $J^{X_Y}$ tends to decrease when the LW and SW biases decrease, which is a very desirable property of $J^{X_Y}$.

For $J^{ZONAL}$ (Fig. 6g–i), the main cloud of points has a weak tendency for a positive bias in the global annual net flux, implying that $J^{ZONAL}$ constrains somewhat better the global annual mean flux than $J^{X_Y}$. There is a very clear tendency for $J^{ZONAL}$ to decrease when the LW and SW biases decrease. The default model is somewhat outlying in the LW/SW fluxes compared to the main cloud of points.

Next, the formulations $J^{G+X_Y}$ and $J^{G+ZONAL}$, which utilize both the global annual net flux and the geographical distribution on monthly basis, are examined. The behaviour of $J^{G+X_Y}$ versus net radiation is largely dominated by the global annual mean term (Fig. 6, j–l). This is mainly because the normalizing factor $\sigma$ is much smaller in Eq. (1) than in Eq. (2) (i.e., the global annual mean flux varies much less than local monthly mean values, and therefore provides a stricter constraint on the parameters). However, $J^{G+X_Y}$ constrains the LW and SW parts somewhat better than $J^G$ alone (Fig. 6a–c). Finally, the behaviour of $J^{G+ZONAL}$ versus net radiation is to some extent dominated by the global annual mean term (Fig. 6m–o), but the zonal net flux distribution makes a significant contribution. The LW and SW parts are nicely constrained such that their biases decrease as $J^{G+ZONAL}$ decreases. Overall, the behaviour of $J^{G+ZONAL}$ is probably the most attractive of the five tested objective functions. In conclusion, addition of the global annual net flux term in $J^{G+X_Y}$ and $J^{G+ZONAL}$ (Fig. 6, last two rows) has the desired effect that the results are unbiased with
Fig. 7. Time-latitude cross section of TOA net flux difference between the default ECHAM5 and CERES observations (panel a), the corresponding difference for the ECHAM5 run with the smallest value of $J_{G+ZONAL}$ (panel b), and the difference between these two ECHAM5 runs (panel c; note the different scale for shading). (d)–(f): Same as (a)–(c) but for the (down-up) shortwave flux at the TOA. (g)–(i): Same as (a)–(c) but for the (down-up) longwave flux at the TOA. (j)–(l): Same as (a)–(c) but for total cloud fraction ($C_{tot}$) compared with ISCCP satellite observations. The parameter values corresponding to default ECHAM5 are CAULOC = 1, CMFCTOP = 0.1, CPRCON = 8 × 10^{-4}, and ENTRSCV = 3 × 10^{-4}; while those for the best run are CAULOC = 29.83, CMFCTOP = 0.0333, CPRCON = 1.05 × 10^{-2}, and ENTRSCV = 1.35 × 10^{-2}. The corresponding values of $J_{G+ZONAL}$ are 28.7 and 16.7.

respect to the net flux and the geographical distributions are respected to some extent.

Based on Fig. 6 (and also Figs. 1 and 2), what can we conclude regarding the choice of the objective function? First, given that a reasonable simulation of the global annual-mean net flux at the TOA is necessary to avoid climate drift in coupled atmosphere-ocean GCMs, it seems prudent to include this term explicitly in the objective function (cf. Jackson et al., 2008). Second, use of the global-mean TOA radiation balance alone does not work well. It is possible to get a single number “right” with very different model climates, as demonstrated by the wide range of global mean LW and SW radiation corresponding to low values of $J^G$ in Fig. 6b and c. Thus, terms addressing modeled spatio-temporal structures should also be included in the objective function. In this respect, use of zonal and monthly-mean values ($J_{ZONAL}$ and $J_{G+ZONAL}$) appears a better choice than the use of local monthly-mean values ($J^G$ and $J^{G+XY}$). The reason for this is that zonal values exhibit smaller interannual “random” variations than the local values (i.e., values of $\sigma_0^F$ in Eq. (3) are substantially smaller than those of $\sigma_0^F$ in Eq. 2). In order for MCMC to be able to efficiently distinguish “good” from “bad” parameter combinations, the systematic impact of parameter changes needs to be relatively large compared to the random variations.

Lastly, it is by no means our purpose to imply that $J_{G+ZONAL}$ is the ultimate solution to the problem of defining the objective function. Other model fields beyond the net flux could/should be included. Also, it should be stressed that while the use of zonal means is simple, it is hardly the optimal choice for the description of spatial structures. For example, the use of empirical orthogonal functions, as in Jackson et al. (2008), is certainly an option worth considering.

4.4 Illustration of the simulation errors

Figure 7 illustrates the impact that a parameter retuning through the MCMC process has on the climate simulated by ECHAM5. Two model runs are considered: the “default run” using the default parameter setting, and the “best run” corresponding to the smallest value of the objective function $J_{G+ZONAL}$. The corresponding values of $J_{G+ZONAL}$ are 28.7 and 16.7, respectively.

For the default run, the largest net flux errors appear at high latitudes ($\sim 55^\circ$ S and $\sim 60^\circ$ N) during local summer, with differences of about $\sim 40\ Wm^{-2}$ from CERES EBAF data (Fig. 7a). At lower latitudes, smaller and predominantly positive biases prevail. For the optimized closure parameters (Fig. 7b), the maximum monthly mean errors are reduced by about $10\ Wm^{-2}$. The pattern of differences between the two runs (Fig. 7c) is, for the most part, opposite to that of the original biases.

The SW and LW fluxes at the TOA are considered in Figs. 7d–f and g–i, respectively. It can be seen that the improved simulation of the net flux in the “best run” mainly results from reduced biases in SW radiation, especially at high latitudes. The impact of parameter tuning on zonal-mean longwave fluxes is, overall, neutral: the positive bias at mid- to high-latitudes is reduced, but at the same time, the negative bias in the tropics is increased. For precipitation (not shown), the impact of tuning also appeared neutral: the “default run” and the “best run” both featured essentially similar biases when compared to Climate Prediction Center’s (CPC) Merged Analysis of Precipitation (CMAP) data (Xie and Arkin, 1997).
Finally, total cloud fraction is considered in Figs. 7j–l. Compared to the International Satellite Cloud Climatology Project (ISCCP) D2 data (Rossw et al., 1996; Rossow and Duedel, 2004), the default run features too much cloudiness at high latitudes, and too little cloudiness at lower latitudes, with largest negative biases around 30°S (Fig. 7j). In the “best run”, cloudiness is, overall, reduced (Fig. 7l). While this alleviates the positive bias compared to ISCCP data at high latitudes, the negative bias prevailing over most of the globe is increased (Fig. 7k). In summary, some of the quantities not included in the cost function are improved, while others are deteriorated.

5 Discussion

The MCMC approach requires long chains of model runs and is therefore best applicable to models that can be run relatively fast. In the present work, we have demonstrated (as far as we know, for the first time) that it is viable to apply MCMC to parameter estimation in an atmospheric general circulation model (GCM) used for climate simulations. This is based on three facts: the low spatial resolution of the model, application of the adaptive MCMC algorithm (DRAM), and the relatively fast response of atmospheric processes to “external forcing” (in our case, changes in parameter values). As to the limits of the approach, we note that in, e.g., ocean GCMs, the response time scales are much longer and MCMC would be computationally more demanding. Also, additional care is needed in selecting the cost function if we model systems which include important reservoirs associated with long time scales, such as carbon pools. This is the case with comprehensive Earth system models with sub-models for terrestrial biosphere and ocean biogeochemistry. One can of course estimate parameters off-line for terrestrial biosphere models (Tuomi et al., 2009), for instance, but interactions and feedbacks with the rest of the modelling systems are omitted in this procedure.

Traditional model parameter sensitivity analysis applies perturbations on model parameters, and draws conclusions about the sensitivity of model simulations on parameter values. This is typically done separately for different model parameters. This study illustrates that the range of parameter values that can produce good simulations in terms of an objective function can be much wider when more than one parameter is considered simultaneously. This is because the combined effect of two or more parameters can keep the model simulation in an acceptable region. Traditional sensitivity analysis thus makes the parameter space to appear more limited than it really is. Also, it is extremely hard to find these combined effects with traditional methods.

One issue of concern with the MCMC approach is related to error compensation. The optimal values of the closure parameters may depend on processes that these parameters do not directly influence. For example, all-sky net radiation at the TOA is a sum of clear-sky net radiation and cloud radiative forcing. Any bias in clear-sky radiative transfer calculations could influence the posterior distribution of closure parameters that affect cloudiness. The problem of error compensation is, however, not inherent to MCMC but applies to model returning in general. Presumably the best way to mitigate this problem in the framework of MCMC is to carefully select an objective function that accounts for multiple aspects of climate. Various other fields beyond the TOA net radiation could be used, as in Jackson et al. (2008).

In this article, the definition of the objective function has been based on very simple statistics such as global and zonal mean values. More sophisticated formulations would account for observed climate phenomena, especially those associated with three-dimensional distributions and possibly including also their temporal evolution. The spatial characteristics can be captured using standard statistical techniques, such as empirical orthogonal functions. Their extensions (e.g., Ilin et al., 2006) can account for more distinctive features of the observed climate variability. Formulation of such an advanced cost function is one of the future directions of our research. Other questions that have to be addressed in the cost function formulation are, e.g., how to combine several similarity criteria in one objective function, and what is the length of climate simulation required to alleviate the effects of purely random variations in the objective function.

6 Conclusions

All general circulation models of the atmosphere or ocean – including climate models – contain closure parameters to which the model simulations are sensitive. These parameters appear in physical parameterization schemes where some unresolved variables are expressed by predefined parameters. In climate modeling, typically, best available expertise is used to define the optimal closure parameter values, based on observations, process studies, large eddy simulations, etc. This procedure has the drawback that little is learned about the parameter posterior distributions: is the optimum local or global, are parameters correlated, etc. Here, parameter estimation, based on the adaptive Markov chain Monte Carlo (MCMC) method, was applied for estimation of joint posterior probability distribution of closure parameters in the ECHAM5 climate model run at a coarse horizontal resolution T21. The four selected parameters related to clouds and precipitation were sampled by an adaptive random walk process, subject to an objective function. Five alternative formulations of the objective function were tested, all of which were related to the net radiative flux at the top of the atmosphere.

We have demonstrated the viability of MCMC methods, especially adaptive MCMC, for the objective estimation of the uncertainties related to closure parameters in climate models. For the four closure parameters, we found an
MCMC chain consisting of 4500 one-year ECHAM5 runs sufficient for a full exploration of the posterior distribution. Chains of 1000 runs were somewhat too short due to an initial drift of the parameter distributions. Even with the most promising cost function, only two parameters (CMFCTOP and ENTRSCV) were found to identify and produce reasonable tight posterior uncertainties within the predefined parameter limits (Fig. 4). However, it is a strength of the MCMC methodology that it allows one to study the identifiabilities and the correlation structures of the parameters (Fig. 5) even when the parameters do not identify. Further work is ongoing in the construction of more refined objective functions for the estimation of the closure parameters and their uncertainties.

Acknowledgements. Pleasant discussions with Erich Roeckner, Marco Giorgetta, and Tuomo Kauranne are gratefully acknowledged. This research has been supported by the Academy of Finland (project numbers 127210, 132808, 133142, and 134999).

References


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

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 in case 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 non-linear expressions of the geo-physical and bio-geochemical 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.

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 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 a 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 times 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 most parts 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 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 \( \pi(\theta | y) \) is the random walk Metropolis-Hastings (MH) algorithm [Metropolis et al., 1953, Hastings, 1970]. At each MH step, one randomly selects a candidate value \( \theta_{i+1} \) from a proposal distribution \( q(\cdot | \theta_i) \) that may depend on the current point \( \theta_i \). Assuming a symmetric proposal distribution, the proposed point is accepted with probability \( \alpha = \min(1, \pi(\theta_{i+1} | y) / \pi(\theta_i | y)) \). 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 \( \pi(\theta | y) \), 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 \( \theta_{i+1} \sim N(\theta_i, \Sigma_i) \), where \( \Sigma_i = \text{Cov}(\theta_1, ..., \theta_i) + \epsilon I \). 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 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

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

\[ \begin{align*}
\text{PROPOSE CANDIDATE} & : \theta^t \sim N(\theta^t, \Sigma) \\
\text{EVALUATE POSTERIOR} & : \pi(\theta^t) \propto p(y|\theta^t) \pi(\theta^t) \\
\text{ACCEPT / REJECT} & : \alpha = \min(1, \pi(\theta_{i+1}^t)/\pi(\theta_i^t)) \\
& \quad \text{if accepted, set } \theta_{i+1}^t = \theta_i^t \\
& \quad \text{otherwise set } \theta_{i+1}^t = \theta_i^t \\
\text{UPDATE PROPOSAL COVARIANCE} & : \\
\Sigma & := \frac{1}{i}(\Sigma + \frac{1}{i}(\theta_{i+1}^t - \bar{\theta})(\theta_{i+1}^t - \bar{\theta})^T) \\
\end{align*} \]

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 \( i \)th sample of the \( j \)th parallel chain is denoted by \( \theta_i^t \) and the proposed candidate for chain \( j \) by \( \bar{\theta}_j \). In the proposal covariance update, the chain index is \( j \in 1, ..., k \). The notation := means substitution.

### 3.1 Example: Gaussian Target Distribution

Let us first consider a 4-dimensional Gaussian target with zero mean and identity covariance \( I \). 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 \( \Sigma_0 = \alpha I \). 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 \( \chi^2 \)-distribution\(^2\).

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

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

### 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 as a cost function, defined as follows:

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

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 $\theta$. The same quantities calculated from observation data sets are denoted in the formulas by superscript $o$ and variances are denoted by $\sigma^2$. The cost function $J(\theta)$ is calculated by running the climate model with parameters $\theta$ 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. Uniform distribution with rather wide upper and lower bounds for the parameters is used as
the prior. The variances $\sigma^2$, 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.](image)

## 4 Early Rejection by Retrospective Simulation

Application of MCMC to parameter estimation problems typically requires a large number of model evaluations. In case 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 example, 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 $\theta_i$. Recall that MH proceeds by proposing a candidate value $\theta_{i+1}$ 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 $\theta_{i+1}$ 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

Figure 4: MCMC chains from a single chain (red) and 7 parallel chain (black) experiments.
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 \( 0 \leq x \leq x_{\text{max}} \). 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 \( \epsilon \) 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 \( x_{\text{max}} \) that lead to different types of posteriors. First, we concentrate the observations to small values of \( x \) and set \( x_{\text{max}} = 4 \). This produces a difficult case, where the parameters are rather poorly identified and a strong nonlinear correlation between the parameters is present. Setting \( x_{\text{max}} = 10 \) 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 \( x_{\text{max}} \) further leads to even better parameter identification with an approximately Gaussian posterior, and the computational savings drop below 10%. On the other hand, decreasing \( x_{\text{max}} \) 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):

\[
\begin{align*}
\frac{dx}{dt} &= \alpha(y - x), \\
\frac{dy}{dt} &= x(\rho - z) - y, \\
\frac{dz}{dt} &= x y - \beta z.
\end{align*}
\]

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 \( 0 \leq x \leq x_{\text{max}} \). 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 \( \epsilon \) 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 \( x_{\text{max}} \) that lead to different types of posteriors. First, we concentrate the observations to small values of \( x \) and set \( x_{\text{max}} = 4 \). This produces a difficult case, where the parameters are rather poorly identified and a strong nonlinear correlation between the parameters is present. Setting \( x_{\text{max}} = 10 \) 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 \( x_{\text{max}} \) further leads to even better parameter identification with an approximately Gaussian posterior, and the computational savings drop below 10%. On the other hand, decreasing \( x_{\text{max}} \) 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.
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).
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.

<table>
<thead>
<tr>
<th>Parameter estimation</th>
<th>Initial value estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>time (sec)</td>
<td>time (sec)</td>
</tr>
<tr>
<td>ER</td>
<td>262.8</td>
</tr>
<tr>
<td>MH</td>
<td>376.4</td>
</tr>
<tr>
<td>ratio</td>
<td>0.698</td>
</tr>
<tr>
<td>ODE count</td>
<td>1134</td>
</tr>
<tr>
<td></td>
<td>1615</td>
</tr>
<tr>
<td></td>
<td>0.702</td>
</tr>
</tbody>
</table>

\[
\frac{dz}{dt} = xy - \beta z.
\]

We use \( \alpha = 10, \rho = 28 \) and \( \beta = \frac{8}{3} \), \( x(0) = 26.61 \), \( y(0) = -2.74 \) 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 \( \theta_{\text{PAR}} = (\alpha, \rho, \beta) \) and estimating the initial values \( \theta_{\text{INIT}} = (x(0), y(0), z(0)) \), using the measurements \((t_i, x_i, y_i, z_i)\) where \( i = (1, \ldots, 15) \). 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 \( \theta_{\text{PAR}} \) works in a routine way, roughly 45% of the proposals are accepted. In contrast, the sampling of the initial values \( \theta_{\text{INIT}} \) - 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 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 from \( \pi(\theta_{\text{PAR}}|y) \) and \( \pi(\theta_{\text{INIT}}|y) \) with and without ER. The experiments were performed in MATLAB, using the numerical ODE solver \texttt{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 by 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:

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

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
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.
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-Hasting 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 Cournet et al. [2011], 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 analyses 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
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.
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 heavy 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 approximate 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 is 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 distributions 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.

Acknowledgements

We thank Dr. Harri Auvinen for contributing in the development of the Fortran90 MPI implementation of the parallel MCMC algorithm. This research has been supported by the Academy of Finland projects 132808, 134937 and 134999.
References


Simulation-Based Optimal Design Using a Response Variance Criterion


Simulation-Based Optimal Design Using a Response Variance Criterion

Antti SOLONEN, Heikki HAARIO, and Marko LAINE

Classical optimal design criteria suffer from two major flaws when applied to nonlinear problems. First, they are based on linearizing the model around a point estimate of the unknown parameter and therefore depend on the uncertain value of that parameter. Second, classical design methods are unavailable in ill-posed estimation situations, where previous data lack the information needed to properly construct the design criteria. Bayesian optimal design can, in principle, solve these problems. However, Bayesian design methods are not widely applied, mainly due to the fact that standard implementations for efficient and robust routine use are not available.

In this article, we point out a concrete recipe for implementing Bayesian optimal design, based on the concept of simulation-based design introduced by (Muller, Sanso, and De Iorio 2004). We develop further a predictive variance criterion and introduce an importance weighting mechanism for efficient computation of the variances. The simulation-based approach allows one to start the model-based optimization of experiments at an early stage of the parameter estimation process, in situations where the classical design criteria are not available. We demonstrate that the approach can significantly reduce the number of experiments needed to obtain a desired level of accuracy in the parameter estimates. A computer code package that implements the approach in a simple case is provided as supplemental material (available online).

Key Words: Adaptive MCMC; Bayesian optimal design; D-optimal design; Importance sampling; Simulated annealing.

1. INTRODUCTION

Let us consider nonlinear models of form $y = f(x, \theta) + \epsilon$, where $y$ is the observed quantity, $\theta$ is the vector of unknown parameters, and $x$ are the experimental design variables. Measurement error is denoted by $\epsilon$. In parameter estimation, the problem is to estimate parameters $\theta$ based on measured $y$. In Bayesian terms, this amounts to finding the posterior...
distribution \( p(\theta|y) \). Applying the Bayes formula gives
\[
p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta) d\theta},
\]
(1.1)
where \( p(y|\theta) \) is the likelihood and \( p(\theta) \) is the prior. The likelihood is constructed using the knowledge of the measurement error. For example, assuming independent and identically distributed Gaussian errors for \( N \) observations \( y = (y_1, \ldots, y_N) \), the likelihood can be written as
\[
p(y|\theta) = \prod_{i=1}^{N} p(y_i|\theta) \propto \exp\left(-\frac{1}{2\sigma^2} SS(\theta)\right),
\]
(1.2)
where the sum of squares \( SS(\theta) = \sum_{i=1}^{N} (y_i - f(x_i, \theta))^2 \) is a measure of the distance from the model to the observations. In traditional nonlinear regression analysis (refer to, e.g., Bard 1974) the sum of squares is minimized with some optimization method to obtain a point estimate \( \hat{\theta} \) and approximative confidence limits are calculated using linearization around \( \hat{\theta} \). In the Bayesian approach, Markov chain Monte Carlo (MCMC) methods can be used to produce a set of samples from the posterior distribution \( p(\theta|y) \) to reveal “all” possible solutions to the parameter estimation problem; see Gelman et al. (2004) for an introduction.

1.1 CLASSICAL OPTIMAL DESIGN

The task in optimal design is, after measuring \( y \) and estimating \( \theta \), to find a new experimental setting \( x \) so that after measuring at \( x \), the uncertainty of \( \theta \) is minimized. In the case of linear models \( y = D\theta \) with Gaussian errors, the distribution of the least squares estimator is analytically known. More specifically, if \( \epsilon \sim N(0, \sigma^2 I) \), the least squares solution is \( \hat{\theta} = (D^T D)^{-1} D^T y \) and we have
\[
\text{Cov}(\hat{\theta}) = \sigma^2 (D^T D)^{-1}.
\]
(1.3)
Designing new measurements means adding new rows \( X \) to the design matrix \( D \), which gives the covariance
\[
\text{C}(X) \propto (D^T D + X^T X)^{-1}.
\]
(1.4)
The classical “alphabetical” design criteria minimize \( \text{C}(X) \) (or maximize its inverse) in different ways, based on the eigenvalues \( (\lambda_1, \ldots, \lambda_n) \) of the covariance matrix. For example, the D-optimal design given by
\[
x_D^* = \arg \min_X \left( \prod_{i=1}^{n} \lambda_i \right) = \arg \min_X (\det(\text{C}(X)))
\]
(1.5)
minimizes the volume of the confidence ellipsoid at \( \hat{\theta} \). An introduction to classical optimal design was given by Atkinson, Donev, and Tobias (2007).

With nonlinear models \( y = f(x, \theta) \) one can obtain a covariance estimate by linearizing the model around \( \hat{\theta} \): the matrices \( D \) and \( X \) are replaced by Jacobians \( J = \partial f(x, \theta)/\partial \theta \) evaluated at \( \theta = \hat{\theta} \) (Box and Lucas 1959). The linear approximation and the dependency on \( \hat{\theta} \) are major problems in classical design criteria. Moreover, if the parameter estimation
problem is ill-posed in the sense that the covariance becomes singular, the classical criteria cannot be used. Thus, one can design measurements with the classical criteria only after the parameters can already be reasonably well estimated from data, and one has to assume that a good approximation of the true value of $\theta$ is known. Thus, classical methods require an initial set of measurements in order to design new measurements. It is clear that the parameter estimation results are better, if one can optimize already these initial measurements.

Bayesian simulation-based optimal design (Muller 1999; Muller, Sanso, and De Iorio 2004) can, in principle, solve these problems. However, the Bayesian simulation-based approach suffers from the lack of practical implementations that would allow routine application in a broad class of design problems. The aim of this article is to propose such an approach.

1.2 Bayesian Optimal Design

An introduction to a general Bayesian experimental design framework was given, for example, in the article by (Chaloner and Verdinelli 1995). Bayesian optimal design is based on defining a utility function $u(x, y, \theta)$ that is a measure of the utility of obtaining measurement $y$ at point $x$ with parameter value $\theta$. Let us assume that we have already some information about model parameters, coded into a density $p(\theta)$. If we have previous measurements $y_{\text{old}}$, $p(\theta)$ represents the current parameter posterior: $p(\theta) = p(\theta|y_{\text{old}})$. Let $p(y|\theta, x)$ denote the likelihood for observing a new measurement $y$ at $x$, given that the parameter value is $\theta$. Bayesian optimal design solves the locality problem by averaging over parameters $\theta$ and possible measurements $y$. The expected utility $U(x)$ that one wishes to maximize wrt $x$ is defined as

$$U(x) = \int \int p(y, \theta|x) u(x, y, \theta) \, dy \, d\theta = \int \int p(\theta) p(y|\theta, x) u(x, y, \theta) \, dy \, d\theta.$$  

(1.6)

All classical “alphabetical” design criteria for linear models can be represented in this general Bayesian framework as well; see the work of Chaloner and Verdinelli (1995).

The integral in Equation (1.6) is often analytically intractable. Direct numerical integration also quickly breaks down when the dimensions of the parameter and measurement spaces grow. In simulation-based optimal design, introduced by Muller (1999), Muller, Sanso, and De Iorio (2004), one uses Monte Carlo sampling methods to integrate over $\theta$ and possible measurements $y$. The expected utility $U(x)$ that one wishes to maximize wrt $x$ is defined as

$$\hat{U}(x) = \frac{1}{n} \sum_{i=1}^{n} u(x, \theta_i, y_i).$$

(1.7)

Sampling from $p(\theta, y|x) = p(\theta)p(y|\theta, x)$ can be carried out by first sampling $\theta$ from $p(\theta)$ and then simulating measurements with the model to get samples from $p(y|\theta, x)$. However, this straightforward Monte Carlo integration, with respect to $(\theta, y)$ for each fixed $x$, can be ineffective. In the article by Muller, Sanso, and De Iorio (2004), the integration was done using MCMC by sampling from target

$$\pi(x, \theta, y) \propto p(y, \theta|x) u(x, y, \theta) = p(\theta)p(y|\theta, x) u(x, y, \theta)$$

(1.8)
using the Metropolis algorithm. The above expression clearly admits \( U(x) \) as its marginal (it is the integrand in Equation (1.6)). Thus, we can sample triples \((x, \theta, y)\) and read \( x \) that follow \( U(x) \) from the columns of the resulting chain. In practice, we often have \( p(\theta) \) available as an MCMC chain and \( y \) can be produced using the model.

Optimal design is traditionally viewed as an optimization problem. However, additional information about the sensitivity of the design problem can be obtained by running MCMC across designs and by looking at the whole design surface \( U(x) \). If one wants to do optimization, in the work of Muller, Sanso, and De Iorio (2004) it was noted that the simulation-based MCMC sampling approach can be turned into an optimization algorithm by sampling from an augmented target

\[
\pi_J(x, \theta_1:J, y_1:J) = \prod_{i=1}^{J} p(\theta_i)p(y_i|\theta_i, x)u(x, y_i, \theta_i)
\]

that admits \( U(x)^J \) as its marginal. Thus, increasing \( J \) has an annealing effect on the target: the samples will be more concentrated around the modes of the target. If direct Monte Carlo integration (see Equation (1.7)) is used, simulated annealing can be implemented in the usual way by directly maximizing \( \hat{U}(x)^J \).

Since the work done by Muller, Sanso, and De Iorio (2004), alternative simulation-based algorithms to maximize the expected utility have been proposed. In the article by Amzal et al. (2006), the Metropolis algorithm is replaced with an interactive particle system, that combines importance sampling and MCMC. It was shown that the method can perform better than the one of Muller, Sanso, and De Iorio (2004). In this article, however, we found the basic adaptive MCMC methods (see, e.g., Haario, Saksman, and Tamminen 2001; Haario et al. 2006; Laine 2008) effective enough for sampling the design space and we do not pursue this algorithmic question further. Instead, we concentrate on a few general-purpose utility functions that can be easily implemented using basic sampling methods. We employ utilities that are based on finding designs that maximize the variance in the model response (measured quantity). Especially, we introduce an efficient way to evaluate the variance utility for designing multiple measurements simultaneously. We illustrate that the approach works also with uninformative priors and in situations where the parameter estimation problem is ill-posed, and linearization-based classical design criteria fail. This allows us to start the design optimization at an early stage of the parameter estimation process. We demonstrate by numerical examples that our approach can significantly reduce the number of experiments needed to obtain accurate parameter estimates.

Our approach is detailed in Section 2. Section 3 includes some remarks and discussion related to the approach. In Section 4, nonlinear examples are given. Section 5 concludes the article.

## 2. OPTIMAL DESIGN USING RESPONSE VARIANCES

How should one define a utility function \( u(x, y, \theta) \), using an existing MCMC chain from \( p(\theta) \)? A straightforward extension to classical local criteria is to perform linearization at
SIMULATION-BASED DESIGN USING RESPONSE VARIANCES

5
different points given in the chain. For example, using the D-optimal criterion would give

\[ u(x, \theta_i, y) = u(x, \theta_i) = \det(J(x, \theta_i)^T J(x, \theta_i)), \]

(2.1)

where \( J(x, \theta_i) \) is the Jacobian with design points \( x \) added, evaluated at a chain point \( \theta = \theta_i \).

By varying \( \theta_i \) one may, to some extent, solve the problems related to locality and nonlinearity, supposing that the problem is well-posed (nonsingular Jacobians). However, the criterion cannot be used in ill-posed settings.

An intuitive idea is that a measurement helps the most in experimental settings where the model response contains the largest uncertainty, given the current knowledge about \( \theta \).

In practice this amounts to searching for regions of high response variance. Thus, we define the utility for a single measurement simply as

\[ U(x) = \sigma^{-2} \text{Var}_{\theta}(f(x, \cdot)) \approx \sigma^{-2} \frac{1}{N_\theta} \sum_{i=1}^{N_\theta} (f(x, \theta_i) - \bar{f}(x, \cdot))^2, \]

(2.2)

where \( \theta_i, i = 1, \ldots, N \), is a sample from the chain and \( \bar{f}(x, \cdot) \) is the sample mean of the calculated model predictions. If the measurement error does not depend on \( x \) (iid measurement error), we simply maximize the response variance. If the error depends on \( x \), we can just plug in the variance \( \sigma^{-2} = \sigma^{-2}(x) \) in the above formula: if two points give equal response variance, we prefer the one with smaller measurement error.

We note that the response variance criterion is close to the model discrimination or distinguishability methods (see, e.g., Bard 1974; Isaacson 1986): if all \( f(x, \theta_i) \) are considered as different possible models, we want to make a new experiment that discriminates between different possible responses. A similar approach was used in simulation-based experimental design context in the work of Haario and Turunen (2003) and applied in that of Haario and Kalachev (2003), where a design was chosen so it that maximizes the distance between two possible model predictions:

\[ \mathbf{x}^* = \arg \max \| f(x, \theta_1) - f(x, \theta_2) \|. \]

The points \( \theta_1 \) and \( \theta_2 \) were deterministically chosen based on the principal components calculated from the parameter MCMC chain.

In the book by Seber and Wild (1989) it was noted that the response variance utility has a connection to classical D-optimal design in the linear case. Let us consider a linear model and add a single row \( x \) to the existing design matrix \( D \) (designing multiple measurements at once is considered later). Applying the matrix determinant lemma (see, e.g., Harville 1997), we get

\[ \det(D^T D + x^T x) = (1 + x(D^T D)^{-1} x^T) \det(D^T D) \propto 1 + \sigma^{-2} x \sigma^2 (D^T D)^{-1} x^T. \]

(2.3)

Thus, D-optimality here is equivalent to maximizing \( x \sigma^2 (D^T D)^{-1} x^T \), which is the response variance (assuming iid Gaussian errors): \( \text{Var}(x_\theta) = x \text{Cov}(\theta) x^T = x \sigma^2 (D^T D)^{-1} x^T \).

This means that we can calculate the D-optimal criterion by considering only the distribution of the model response. This interpretation of D-optimality is directly generalized to the nonlinear case by Equation (2.2): we approximate the response variance by calculating the sample variance from the possible model responses given by samples from \( p(\theta) \).

Other utilities (design criteria) that use response variances exist. A well-known approach is to look for a design that minimizes the average predictive variance over a given
design region. In classical design literature, this is known as V-optimality; see, for example, the book by Myers and Montgomery (2002). A similar utility in the nonlinear setting was developed in the work of Cohn (1996), where variances are estimated using linearizations in the parameter space. The approach used here differs from these in the sense that we directly maximize response variance. This criterion is known as such; see again the book of Seber and Wild (1989). However, note that unlike the classical formulation, our approach does not involve derivatives with respect to the parameters (Jacobians) and the question about the “true” parameter value is avoided in the simulation-based approach.

The simple utility presented above works for the design of one new measurement at a time. If the task is to design multiple measurements at once or if one measurement leads to vector-valued data, one has to take into account that each individual measurement changes the parameter posterior and therefore also the utility of the next measurements. The design criterion has to take this dependency into account. How can the response variance criterion be extended to this setting? A simple sum of the utilities (2.2) at different design points is not enough; for instance, the parameters may be better discriminated by an experiment with a large variance at one design point rather than moderately large variances at many points. The utility of a single measurement conditionally depends on the other measurements designed at the same time.

Let us first discuss the design criterion in the linear case, when multiple measurements are optimized. Designing multiple measurements means adding multiple rows to the design matrix. We denote the design matrix with \( n \) rows \( x_1, \ldots, x_n \) added as \( \tilde{D}_n \). Applying the matrix determinant lemma recursively leads to

\[
\det(\tilde{D}_n^T \tilde{D}_n) = \det(D^T D) \prod_{i=1}^{n} (1 + x_i (\tilde{D}_{i-1}^T \tilde{D}_{i-1})^{-1} x_i^T )
\]

(2.4)

\[
\propto \prod_{i=1}^{n} (1 + \sigma^{-2} x_i \sigma^2 (\tilde{D}_{i-1}^T \tilde{D}_{i-1})^{-1} x_i^T )
\]

(2.5)

where \( \tilde{D}_0 = D \). The terms \( x_i \sigma^2 (\tilde{D}_{i-1}^T \tilde{D}_{i-1})^{-1} x_i^T \) in the product correspond to response variance given the rows added so far. This motivates our utility function

\[
u(x, y, \theta) = \prod_{i=1}^{n} (1 + \sigma^{-2} \text{Var}_P(\theta| y_{1:i-1})(f(x_i, \theta)))
\]

(2.6)

where we use the notation \( y_{1:k} = (y_1, \ldots, y_k) \) for data from the first \( k \) experiments (each component is a scalar here), performed at design points \( x_{1:k} = (x_1, \ldots, x_k) \). The expression \( \text{Var}_P(\theta| y_{1:i-1})(f(x_i, \theta)) \) gives the response variance at \( x_i \), given measurements \( y_{1:i-1} \) at points \( x_{1:i-1} \). The utility has an intuitive interpretation: each term in the product corresponds to new information brought by a single measurement. When the variance gets smaller, the terms approach 1 and the “information gain” decreases. The effect of the measurement error is also visible in the formula: the smaller the measurement variance is, the more a large response variance matters.

Now, instead of optimizing for a single \( x \), we can use the above criterion for evaluating the utility for the entire vector \( x \). The vector of future observations \( y \) can be simulated
using the model with parameter value $\theta$. By changing the parameter value in the MCMC sampler in the design space, we average over possible parameters and possible future measurements (see Equation (1.8)). Measurements are simulated by adding noise to the model responses according to our error model. If we assume additive iid Gaussian errors, we simulate measurements as $y_i = f(x_i, \theta) + \epsilon$.

The crucial step left is the calculation of the conditional variances in Equation (2.6). This seems laborious at first. In principle, since the parameter posterior changes after each single simulated measurement, approximating the value of $\text{Var}_p(\theta | y_1: (i-1))(f (x_i, \theta))$ would require a loop with MCMC resampling of the model parameters with simulated measurements $y_1: (i-1)$. Since the model evaluation is often numerically heavy (e.g., a numerical solution to a differential equation system), the computational cost of full MCMC resampling would be excessive for all but trivial cases.

Instead, we propose to achieve the “posterior shrinkage” effect of the new simulated measurements by an approximation based on importance weighting. We evaluate the utility at $x_i$ by calculating a weighted variance, where each simulated model response is weighted based on the likelihood of previous simulated measurements, $p(y_1: (i-1)|\theta, x_1: (i-1))$:

\[ \text{Var}_p(\theta | y_1: (i-1))(f (x_i, \cdot)) \propto \sum_{j=1}^{N_i} w_j (f (x_i, \theta_j) - f (x_i, \cdot))^2, \quad \text{with} \quad (2.7) \]

\[ w_j = p(y_1: (i-1)|\theta_j, x_1: (i-1)) \propto \prod_{k=1}^{i-1} p(y_k|\theta_j, x_k). \quad (2.8) \]

Weights are normalized to sum up to 1 and the mean $\bar{f}(x_i, \cdot)$ is calculated as a weighted mean. Thus, instead of selecting parameters from a subdomain of the previous distribution created by MCMC sampling, we re-weight the selected parameters with the new simulated measurements. For a graphical illustration of the weighting approach, see Figures 7 and 8 in Section 4.

We note that this approach is analogous to resampling the model responses with replacement after a simulated measurement using importance resampling, but since we only need the response variance, resampling is not necessary, since the needed statistics can be calculated directly using the weighted samples. Numerical comparisons have shown that the weighted variance produces good estimates of the true conditional variance. The approach allows us to estimate the conditional variances using a relatively small amount of model simulations, without resampling parameters after added simulated measurements.

To clarify the implementation, we formulate below the Metropolis algorithm for performing MCMC in the design space using utility function (2.6). In our examples, we use an adaptive MCMC sampler that constantly updates the proposal distribution $q$ based on the obtained samples, but for simplicity we present our utility in connection with the standard Metropolis sampler used by Muller, Sanso, and De Iorio (2004).

The sampling algorithm can be turned into an optimization algorithm using simulated annealing and increasing $J$ along the iterations; see Equation (1.9) and the examples below.
Algorithm Designing Multiple Measurements

Assume that the MCMC sampler is currently at design $\mathbf{x}$ that has utility $u$ and samples from $p(\theta)$ are available. One Metropolis step for sampling a new design vector is given as

1. propose a new design $\tilde{\mathbf{x}} \sim q(\cdot | \mathbf{x})$
2. choose samples $(\theta_1, \ldots, \theta_{N_\theta})$ and a “true” value $\theta$ randomly from $p(\theta)$
3. simulate responses $f(\mathbf{x}, \theta_i)$ and measurements $y_i = f(\mathbf{x}, \theta) + \epsilon$
4. calculate the weighted variances $\text{Var}_p(\theta | y_1^{i-1})(f(x_i, \theta))$ for each added design point $x_i$ as in Equation (2.7)
5. evaluate the expected utility $\tilde{u}$ as in Equation (2.6)
6. accept $\tilde{\mathbf{x}}$ with probability $\alpha = \min(1, \tilde{u}/u)$.

3. REMARKS

- An attractive feature of the proposed utility is that it is computable even if the Jacobians are not available due to lack of information about $\theta$. In classical design of experiments, one often performs an initial set of measurements, for example using a factorial design, to make the problem suitable for subsequent model-based optimal design criteria (e.g., D-optimal design). Simulation-based design removes the need for initial trial-and-error designs. In fact, the utility can be computed even with only a flat prior, such as upper and lower bounds for parameters. This can be useful, since often the model formulation itself favors some experimental designs: while the first measurements are not able to identify parameters, they do exclude certain parameter values. In the same way, our design approach allows us to avoid uninformative initial experiments. In this spirit, it is possible to “optimize” the very first measurements, without an initial set of measurements or initial parameter guess, that would be needed if using classical design criteria. This is demonstrated in Section 4 in Examples 1 and 3.

- The experimental design problem is traditionally viewed as an optimization problem, where one looks for a single design point $\mathbf{x}^*$ that maximizes the expected utility. However, using MCMC sampling also to explore the design space allows one to look at the distribution of promising regions in the design space and study the sensitivity of the design problem. Sometimes the design problem does not have a unique optimum and a design variable might not even have a significant effect on the utility. The samples produced by the algorithm can be used to draw an “experimental map” that contains the design regions of high expected utility. This can be seen as an improvement to classical design methods that give the solution as a single optimal design point. Naturally, the MCMC sampling approach can be turned into an optimization method by annealing the target.
• When MCMC is used to explore the design surface instead of an optimizer, the result is a distribution instead of a single optimal point, earlier referred to as an “experimental map.” However, one has to have a way of picking a point \( x^* \) from the map. In many of our examples, a mode of the distribution can be chosen “by hand” by looking at the marginal densities of the distribution. In many cases this is easy, since the optimal points are often found at the corners of the design region. When more contrast in the experimental map is needed, annealing can be performed by increasing \( J \), as demonstrated in Example 1 in Section 4.

• In the evaluation of the response variance utility in Equation (2.2), we choose the samples \((\theta_1, \ldots, \theta_{N_{\theta}})\) randomly from the existing MCMC chain to evaluate the utility. To save CPU time, one would like to minimize the number \( N_{\theta} \) of (often computationally costly) model evaluations. But choosing the samples randomly leads to a stochastic target function: the utility evaluated twice at the same design point \( x \) produces slightly different utility values. To regularize this variability, a sufficient number \( N_{\theta} \) of samples is needed: increasing \( N_{\theta} \) reduces the variation in the utility, making the MCMC sampling more robust. Numerical tests show, however, that this stochasticity is handled well by the MCMC samplers: we get the same results using a relatively small \( N_{\theta} \) as when using, for instance, all the points in the \( p(\theta) \) chain. The number of model responses \( N_{\theta} \) needed to evaluate (2.2) with small enough variability is problem dependent. As a guide to select \( N_{\theta} \), one may first evaluate the variance of (2.2) due to sampling with respect to \( x \), and then find a minimal \( N_{\theta} \) that produces a statistically significantly smaller variability. In our examples we used values ranging from 10 to 200.

• The examples are presented in this article for the case where we only have one observed component. However, the approach generalizes to the multi-response case. One way to handle the multivariate case is to use the utility for multiple measurements given in the previous section. One can also do the derivation in Equation (2.3) for multiple measurements simultaneously; see, for example, the book by Seber and Wild (1989). For example, assuming a linear model with design matrix \( D \) and iid Gaussian error with variance \( \sigma^2 \), the D-optimal criterion becomes

\[
\det(D^T D + X^T X) \propto \det(I + X(D^T D)^{-1}X^T) \\
= \det(I + \sigma^{-2} X \sigma^{-2} (D^T D)^{-1} X^T) \\
= \det(I + \sigma^{-2} \text{Cov}_{p(\theta|y)}(X\theta)) \\
= \det(I + \sigma^{-2} \text{Cov}_{p(\theta|y)}(f(x, \theta))).
\]

Thus, instead of response variances, we calculate response covariances in the criterion. In the book by Seber and Wild (1989), the above criterion was used for nonlinear models using linearizations. In our simulation-based context, an empirical covariance matrix can be calculated from the simulated samples. Naturally, the response covariance extension can be combined with the weighting approach used in the previous section.
The examples in this article contain only continuous design variables, which are often encountered in engineering applications. However, in many cases, such as experimental design in clinical trials, design variables can be discrete. The approach presented in this work generalizes directly to the case where there are discrete variables. For discrete cases, one needs to implement a discrete proposal mechanism in the MCMC sampler (in step 1 of the algorithm).

4. SIMULATED EXAMPLES

In this section, we give three simulated examples of the proposed design approach. The first one deals with estimating the rate of a chemical reaction and uses the variance utility to design both single measurements iteratively and multiple measurements at once. The example is from the work of Berger et al. (2000), a benchmark project for estimation and design in chemical reaction kinetics. The second one is a simple heat transfer example, used to demonstrate the utility for multiple measurements designed at once. The third example, adopted from the work of Muske and Georgakis (2003), is a more complicated ODE system that describes an exothermic chemical reaction mechanism.

We have included a supplemental material package to the article, which contains MATLAB codes that demonstrate our optimal design approach in the heat transfer example; see the Supplementary Materials section.

4.1 E U R O K I N BENCHMARK

We first demonstrate how the parameter re-weighting procedure works for the case of multiple design points (see Equation (2.7)). We made experiments with different cases, where we compared the “one by one” measurement strategy to strategies where multiple measurements are designed in batches of two and three. After each design, we perform the respective “measurements” by evaluating the model with the true parameters, synthetically adding noise, and creating the MCMC chain to get the sampled approximation of the posterior distribution.

The rate \( r \) of the reaction

\[
\text{CO} + 2\text{H}_2 \leftrightarrow \text{MeOH}
\]  

is modeled as a function of temperature \( T \) and partial pressures of the reactants, \( p_C, p_H, \) and \( p_M \):

\[
r = \frac{k_{\text{ref}} \exp(-E/(RT^*)) (p_Cp_H^2 - p_M/K_{\text{eq}})}{(1 + K_1p_C + K_3 \exp(-\Delta H_3/(RT^*)) + K_4p_H/p_C)pcp_H}.
\]  

(4.2)

In this example, the unknown parameters are \( \theta = [k_{\text{ref}}, E, K_1, K_3, K_4, \Delta H_3] \) and the experimental conditions \( \mathbf{x} = [T, p_C, p_H, p_M] \). In the original example, 27 measurements (as given by a central composite type of design) were given and the task was to design the 28th measurement. Here we design experiments iteratively, starting the optimization from the first measurements. Only a flat prior is assumed by giving lower and upper bounds for the parameters.
We searched for the optimal designs using simulated annealing by increasing $J$ by one in the annealed target (Equation (1.9)) at every 2000th MCMC iteration. An example of the development of the MCMC iterations is given in Figure 1. When the first measurements are designed, such a clear unique optimal design may not exist, and the simulated annealing algorithm simply freezes to some informative value. Later on in the estimation process, a unique optimum can usually be found.

In Figure 2, we compare the sizes of the parameter posteriors with varying number of experiments using the different measurement strategies. The posterior size is estimated as the determinant of the sample covariance calculated from the posterior MCMC chain. The results obtained with the “two and three at a time” strategies are roughly as good as
the results with the “one by one” strategy, which indicates that our re-weighting approach works well. The parameter posteriors for the three strategies with six measurements are compared in more detail in Figure 3. While the “one by one” strategy yields the smallest posteriors, all of the strategies result in measurements that lead to unimodal posteriors around the true value.

As an example of designing the “initial” measurements, the effect of the fifth measurement is illustrated in Figure 4, when using the “one by one” strategy. It is clearly visible that the Jacobians are singular when the first experiments are designed, and classical criteria are unavailable. In such an ill-posed case, the solutions to the parameter estimation problem lie in some nonlinear manifold within the whole parameter space. Adaptive MCMC methods can efficiently discover these manifolds, since the sampler automatically learns the correlations between the parameters.

To further compare with the traditional design approaches, we performed a fractional factorial $2^{4-1}$ design, with variable values in the same range as that used in the original plan of 27 measurements. In Figure 5, we compare this to the parameter posteriors given by eight optimized measurements (with the one by one strategy) and the original 27 measurements. The parameters remain clearly unidentified after such an initial design plan, and eight measurements optimized with our approach give better results than even all of the given 27 measurements.

The example shows that starting the design optimization early can result in dramatic reduction in the number of experiments needed to estimate the parameters. Traditionally, a fixed design, a fractional factorial design for example, is initially employed in order to get the parameters well enough identified. Only then model-based optimization of the design
becomes possible. Our approach allows to start the design optimization from the very first experiment. In this example, the classical design criteria start to work (i.e., the Jacobian is non-singular) after six measurements optimized with our approach.
4.2 Liquid Cooling

We demonstrate here the algorithm for designing multiple measurements using a simple (yet realistic) example. A glass of liquid at an initial temperature \( T(0) \) is cooled or heated by water at temperature \( T_w \). The surrounding air is at temperature \( T_a \). The purpose is to study the heat transfer coefficients through the glass \( (k_w) \) and through the air-liquid interphase \( (k_a) \). A model for the liquid temperature may be written as

\[
\frac{dT}{dt} = -\frac{k_w A_w}{M c} (T - T_w) - \frac{k_a A_a}{M c} (T - T_a),
\]

(4.3)

where \( A_w \) and \( A_a \) are the areas of the water-liquid and air-liquid interfaces, \( c \) is the specific heat capacity, and \( M \) is the mass of the liquid. The goal is to estimate \( \theta = (k_w, k_a) \) and the design parameters to be optimized are \( x = (T(0), T_w) \).

Measurements of \( T \) are taken every 2 minutes, stopping at 20 minutes. We consider the situation where one set of measurements already has been performed at \( x = (31, 5) \), and the MCMC chain for parameters has been run. The task is to design optimal temperatures \( x \) for the next set of 10 measurements. For simplicity, we keep the measurement instants fixed. The measurement error was assumed to be Gaussian with variance \( \sigma^2 = 0.33 \).

In such a low-dimensional design problem, it is possible to explore the whole “experimental map” visually. The design surface (a two-dimensional histogram of the sampled \( x \) values) and the effect of the optimized measurements are plotted in Figure 6. There are several local optima in the corners of the design region, especially in those corners further away from the point \( x = (31, 5) \), where measurements have already been done. The global optimal design point is at \( x^* = (4, 60) \) (cold liquid, hot water). Note that the worst measurement, to put both liquid and water to room temperature (no heat transfer would happen), is clearly exhibited as well. The weighting of the parameters, the “posterior shrinkage,” is illustrated in Figure 7 in the parameter space for the optimal design point. In Figure 8, the effect is shown for the model responses at another “good” design point \( x = (60, 60) \) (for clarity, only the five first time points are plotted).

In this example, the Jacobian can be calculated and the classical design criteria can be used. For comparison, we plot also the classical D-optimal design surface in Figure 6, calculated in a grid of values for \( x \). The results are similar, as was expected, since D-optimality agrees theoretically with the response variance criterion in the linear case. In
many cases, however, the Jacobians are not available but the response variance criterion can be computed. This example shows that in a “well-posed” case, the response variance criterion and D-optimal criterion produce comparable results.

4.3 CSTR Reactor Model

In this example, modified from the article by Muske and Georgakis (2003), compound A is fed into a continuously stirred tank reactor (CSTR) with rate $F$. The heat produced by the reaction is removed by a cooling jacket, through which a coolant flows with rate $F_c$. The ODE system for the concentration $c_A$, reactor temperature $T$, and coolant temperature $T_c$ is

$$
\frac{dc_A}{dt} = -k c_A,$$

$$
\frac{dT}{dt} = \frac{Q}{m} - \frac{U A}{m c_p} (T - T_c),$$

$$
\frac{dT_c}{dt} = \frac{U A}{m c_p} (T - T_c).$$

Figure 7. Development of the weights in the parameter space after adding simulated measurements in the optimal design point $x^* = (4, 60)$. First measurements bring more information about $\theta_1$ and later measurements about $\theta_2$.

Figure 8. Illustration of weighted model responses for the liquid cooling example (see Section 4) at $T(0) = T_w = 60$. Measurements (dots with confidence intervals) are simulated around a response $f(x, \theta)$ (solid line). Gray colors correspond to weights given to different responses by the simulated measurements, used to calculate the variances at $t_1, \ldots, t_5$. The online version of this figure is in color.
Table 1. CSTR model parameters (top), design variables (middle), and fixed control parameters (bottom).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$</td>
<td>$2.5 \text{ h}^{-1}$</td>
</tr>
<tr>
<td>$E/R$</td>
<td>$255 \text{ K}$</td>
</tr>
<tr>
<td>$\Delta H$</td>
<td>$160 \text{ kJ/mol}$</td>
</tr>
<tr>
<td>$h$</td>
<td>$1000 \text{ W/(m}^2\text{C)}$</td>
</tr>
<tr>
<td>$c_A(0)$</td>
<td>$1000 \text{ mol/m}^3$</td>
</tr>
<tr>
<td>$T_f$</td>
<td>$20 \text{ C}$</td>
</tr>
<tr>
<td>$c_A^f$</td>
<td>$2500 \text{ mol/m}^3$</td>
</tr>
<tr>
<td>$T_c^f$</td>
<td>$10 \text{ C}$</td>
</tr>
<tr>
<td>$F$</td>
<td>$0.1 \text{ m}^3/\text{h}$</td>
</tr>
<tr>
<td>$F_c$</td>
<td>$0.15 \text{ m}^3/\text{h}$</td>
</tr>
<tr>
<td>$C_p$</td>
<td>$1600 \text{ kJ/(m}^2\text{C)}$</td>
</tr>
<tr>
<td>$C_{p_c}$</td>
<td>$1600 \text{ kJ/(m}^2\text{C)}$</td>
</tr>
<tr>
<td>$V$</td>
<td>$0.2 \text{ m}^3$</td>
</tr>
<tr>
<td>$V_c$</td>
<td>$0.055 \text{ m}^3$</td>
</tr>
<tr>
<td>$A$</td>
<td>$4.5 \text{ m}^2$</td>
</tr>
</tbody>
</table>

$T_c$ is written as

$$\dot{c_A} = \frac{F}{V}(c_A^f - c_A) - r_A, \quad (4.4)$$

$$\dot{T} = \frac{F}{V}(T_f - T) + \frac{\Delta H}{C_p} r_A - \frac{hA}{C_p V}(T - T_c), \quad (4.5)$$

$$\dot{T_c} = \frac{F_c}{V_c}(T_c^f - T_c) + \frac{hA}{C_{p_c} V_c}, \quad (4.6)$$

where $r_A = k \exp(-E/R) c_A$. Here the estimated parameters are $\theta = (k, E/R, \Delta H, h)$ and the controllable design variables are $x = (c_A(0), T_f, c_A^f, T_c^f, F, F_c)$. Let us assume that we measure every 0.2 hr. The time interval for the experiments is $t \in [0, 10]$. We begin with one set of measurements already done, and the respective MCMC chain created. The assumed true values for the parameters and the experimental setting for the existing measurements are given in Table 1.

The algorithm for multiple simultaneous measurements was employed. The response distribution in the optimal design point, the new simulated measurements, and the effect of the new optimized measurement are given for three iterations in Figure 9. These four sets of measurements (initial and three optimized) are enough to identify model parameters. The classical design criteria are available only after the last set of measurements. But note that after this point the new measurements do not anymore significantly add the accuracy of parameter estimates, because the response variances become smaller than the measurement error variance.

5. CONCLUSION

We have proposed a simple way to implement Bayesian optimal design, using a criterion based on the variance of model predictions. The approach works well in nonlinear
and ill-posed problems, where classical local design methods run into problems. With the proposed approach it is possible to start the design optimization with fewer data than with classical methods, resulting in fewer measurements needed to obtain a desired level of accuracy in the estimates.

The proposed approach utilizes existing parameter samples, typically produced by MCMC sampling, a technique more and more routinely used in modeling. In addition, the sampling in the design space can also be performed, with little tuning effort, using existing samplers. We introduce a novel importance weighting procedure to efficiently compute the
A. SOLONEN, H. HAARIO, AND M. LAINE

proposed design criterion. The approach was tested with three nonlinear design problems and the approach was shown to be computationally feasible with realistic models.

SUPPLEMENTARY MATERIALS

MATLAB-package for Optimal Design: A code package written in MATLAB that implements the optimal design algorithm in the liquid cooling example of Section 4.2. In order to run the code, the MCMC toolbox is needed, which can be downloaded from http://www.helsinki.fi/~mjlaine/mcmc/. For a detailed description, see the README.txt file in the package. (OptDes.zip)

ACKNOWLEDGMENTS

This work was funded by the Finnish Center of Excellence in Inverse Problems Research consortium (Academy of Finland) and by the MASI project of the National Technology Agency of Finland (TEKES).

[Received April 2010. Revised February 2011.]

REFERENCES

Model-Based Process Optimization In the Presence of Parameter Uncertainty

Solonen A. and Haario H., 2011. Accepted for Publication in *Engineering Optimization*.

© 2011 Taylor & Francis. Printed with permission.
Model-Based Process Optimization in the Presence of Parameter Uncertainty

Antti Solonen\textsuperscript{\dag}, Heikki Haario\textsuperscript{*}

\textsuperscript{*}Department of Mathematics and Physics, Lappeenranta University of Technology
\textsuperscript{\dag}Finnish Meteorological Institute, Helsinki, Finland

Abstract

In model-based process optimization one uses a mathematical model to optimize a certain criterion, for example the product yield of a chemical process. Models often contain parameters that have to be estimated from data. Typically, a point estimate (e.g. the least squares estimate) is used to fix the model for the optimization stage. However, parameter estimates are uncertain due to incomplete and noisy data. In this paper, we show how parameter uncertainty can be taken into account in process optimization. To quantify the uncertainty, we use Markov Chain Monte Carlo (MCMC) sampling, an emerging standard approach in Bayesian estimation. In the Bayesian approach, the solution to the parameter estimation problem is given as a distribution, and the optimization criteria are functions of that distribution. We study how to formulate and implement the optimization and show by numerical examples that parameter uncertainty can have a large effect in optimization results.

Keywords: Process Optimization; Bayesian Estimation; MCMC; Stochastic Optimization

1 Introduction

In mathematical modeling, one often has unknown parameters in the model that need to be estimated from measured data. After fitting the model to the measurements, the model can be used to study the phenomenon of interest. One common use for the fitted model is process optimization, where the goal is to find operating conditions that optimize a certain criterion, e.g. maximize the yield of a desired product in a chemical reaction system. The problem in the optimization is that parameter estimates are often uncertain, since they are estimated from incomplete and noisy measurements. Traditionally, one chooses a specific point estimate for the parameter, e.g., the least squares estimate, and ignores the uncertainty in the parameters. In this paper, we study how parameter uncertainty can be incorporated into process optimization. To quantify the parameter uncertainty, we use Bayesian parameter estimation and MCMC sampling (see section 2 for a short introduction and the appendix for more details).

Let us define more clearly what we mean by model-based process optimization. We consider nonlinear models of form \( y = f(x, \theta) + \epsilon \), where \( x \) are the controllable variables and \( \theta \) the unknown parameters. In parameter estimation, the task is to estimate \( \theta \) from measurements \( y \). Measurement error is denoted by \( \epsilon \). In process optimization, one wishes to optimize some criterion \( c(x, \theta) \) with respect to the control variables \( x \). For example, in chemical reaction engineering, \( x \) could be the temperature of a reaction, \( \theta \) might represent the reaction rate constants and \( c(x, \theta) \) could be the yield of a product. A common way to optimize processes is to use response surface methods, where a sequence of experiments is used to create empirical regression models, that are used to find the optimal process, see e.g. (Myers et al. 2009). In many cases, however, a mechanistic model \( f(x, \theta) \) is available, that enables more comprehensive design...
optimization, often with much smaller amount of experimental work. This approach is called model-based process optimization.

In process optimization literature, a few approaches for handling parameter uncertainty have been proposed. In these approaches – see e.g. (Ma and Braatz 2003; Rooney and Biegler 2003) and the review paper of Lee and Chen (2009) – a fixed parametric form for the parameter uncertainty (e.g. Gaussian) is used. This is in accordance with the standard nonlinear parameter estimation procedures in engineering: usually a Gaussian approximation of parameter uncertainty is used to calculate confidence intervals for parameter estimates, based on linearization around a point estimate. Our approach provides a fully nonlinear extension to the existing methods, that utilizes the output from MCMC parameter estimation.

Recently, Bayesian model fitting methods have opened a way for proper statistical analysis of parameter estimation for nonlinear models. For examples of Bayesian parameter estimation and MCMC for model fitting in engineering applications (chemical reaction engineering) see e.g. (Vahteristo et al. 2008; Kuosa et al. 2009). In the Bayesian approach, one can avoid linear approximations in parameter uncertainty analysis. In this paper, we show how these results can further be used as input for process optimization. The approach for solving the resulting stochastic optimization problem is based on the same random sampling methods as those used in parameter estimation.

The paper is organized as follows. We start Section 2 by shortly recalling how Bayesian parameter estimation with MCMC works. Then, we present our process optimization approach together with some general purpose optimization criteria that can be used when the parameters are given as a distribution instead of a point estimate. Section 3 contains remarks and discussion and in Section 4 we give two synthetic examples of our approach. Section 5 concludes the paper. Since MCMC is not a standard tool in engineering, we give a brief introduction and a simple parameter estimation example in the appendix.

2 Parameter Uncertainty in Process Optimization

When unknown parameters are estimated from data, it is important to obtain information about the uncertainty of the estimates. Uncertainties can be quantified using statistical methods. Classical statistical analysis, that gives the optimal parameter values and error estimates for them, is approximate (based on linearization of the model) and may sometimes be quite misleading. Moreover, the question of the reliability of the model predictions is left open, i.e., how is the uncertainty in model parameters reflected to the model response. Both of these problems may be properly treated by Markov chain Monte Carlo (MCMC) methods. In MCMC, the estimation of model parameters and predictions are performed according to the Bayesian paradigm. All uncertainties in the data as well as the modeling results are treated as random variables that have statistical distributions. Instead of a single fit to the data, ‘all’ parameterizations of the model that, statistically, fit the data ‘equally well’ are determined. A distribution of the unknown parameters is generated using available prior information (e.g. previous studies) and statistical knowledge of the observation noise. Computationally, the distribution can generated using the Markov chain Monte Carlo (MCMC) sampling approach.

MCMC gives the solution to the parameter estimation problem as a set of samples from the distribution of the parameters, instead of a point estimate. See the appendix for a brief introduction to MCMC and for a demonstration of the differences between MCMC and classical parameter estimation. In the rest of the paper, it is assumed that MCMC parameter estimation can be done, and samples from the parameter distribution are available. In this Section, we show how the uncertainty presented in the form of samples can be incorporated into model-based optimization problems.

In more detail, Bayesian estimation considers \( \theta \) as a random variable that has probability density \( p(\theta | y) \), from which we are able to produce samples using e.g. MCMC (see the appendix for more details). Following this interpretation, also the optimization criterion \( c(x, \theta) \) is a distribution that can obtain a range of possible values at any point \( x \). Thus, we are dealing with an optimization problem with a stochastic target function. There are numerous approaches for these kind of optimization problems, see
e.g. (Shapiro et al. 2009) for a methodological introduction. In this paper, we present a novel approach, based on the same MCMC methods that are used for the parameter estimation.

Instead of optimizing \(c(x, \theta)\) for a specific fixed \(\theta\), the Bayesian approach allows one to optimize a function of the distribution of \(c(x, \theta)\). However, it is not obvious how this statistical knowledge of the modelling uncertainty should be taken into account. Should we, for example, maximize the expected value of the cost function? Or should we employ a more conservative ‘worst-case’ approach where we maximize the smallest predicted product yield, taking into account the parameter distribution? Moreover, we have to consider the algorithmic details of the optimization, to avoid excessive CPU times that often plague Monte Carlo type calculations.

Below, we introduce some ‘distribution-based’ optimization criteria and present two ways to evaluate them. First, we discuss the obvious (but computationally costly) Monte Carlo approach, where the distribution of \(c(x, \theta)\) is directly simulated with different parameter samples given by MCMC parameter estimation. Then, we present how MCMC integration can be used to efficiently compute similar criteria.

### 2.1 Direct Monte Carlo Sampling

An obvious candidate criterion for optimization is the expectation: one averages \(c(x, \theta)\) over \(\theta\) and optimizes the mean. This reduces the risk of obtaining \(x\) that produces a good value for \(c(x, \theta)\) only locally for a specific value for \(\theta\). The mean criterion is defined as

\[
C(x) = \mathbb{E}_{p(\theta | y)}[c(x, \theta)] = \int c(x, \theta)p(\theta | y) d\theta.
\]

This integral can be approximated using the existing MCMC samples from \(p(\theta | y)\). The most obvious way is to pick a (large) number of samples \((\theta_1, ..., \theta_N)\) from the MCMC output and use a direct Monte Carlo approximation:

\[
C(x) \approx \frac{1}{N} \sum_{i=1}^{N} c(x, \theta_i).
\]

However, using expectation alone as an optimization criterion does not take into account the variability in the criterion. For instance, it might be useful to look for solutions that both give good criterion values on average and have small variances. We can add a requirement of small variance to the optimization criterion by e.g. penalizing large standard deviations:

\[
C(x) = \mathbb{E}_{p(\theta | y)}[c(x, \theta)] - \alpha \text{Std}_{p(\theta | y)}[c(x, \theta)].
\]

Tuning parameter \(\alpha\) defines the weight given for the variability in the criterion: decreasing \(\alpha\) gives more weight to the expectation. We note that this criterion is close to the mean-risk models discussed in e.g. (Shapiro et al. 2009). This robust mean criterion in equation 3 can also be computed easily using the direct Monte Carlo approximation using empirical mean and standard deviation formulas.

As the third option, we mention the ‘worst case criterion’, where the worst possible value of \(c(x, \theta)\) is maximized:

\[
C(x) = \min_{\theta} c(x, \theta).
\]

In this case, direct Monte Carlo approximation means calculating \(c(x, \theta)\) with different possible values for \(\theta\) and finding the minimum from the calculated samples.

The direct Monte Carlo approximation of the above criteria is simple to implement and include in different optimization routines. However, the approach is computationally challenging, since we might need a large number of points \(\theta_i\) to evaluate \(C(x)\) just once. This can be a problem, if the optimization criterion (simulation model) is computationally expensive to evaluate. In the following, we present another way to approximate \(C(x)\) using MCMC sampling and to optimize \(C(x)\) using a simulated annealing approach, following the optimal experimental design method of Müller et al. (2004).
2.2 MCMC Sampling and Simulated Annealing

The idea, originally presented in (Müller et al. 2004) in a design of experiments context, is to use MCMC sampling for approximating \( C(x) \). When the expectation is maximized (see equation 1), this can be done by running MCMC with target density

\[
\pi(x, \theta) = c(x, \theta)p(\theta|y),
\]

(5)

assuming that \( c(x, \theta) \) is non-negative and bounded for all \( x \) and \( \theta \). The above target density admits \( C(x) \) as its marginal. Thus, we can use MCMC to sample from \( \pi(x, \theta) \) and read \( C(x) \) from the resulting chain. Sampling from \( \pi(x, \theta) \) can be done using e.g. the Metropolis algorithm by proposing a new \( x \), picking a sample from \( p(\theta|y) \) from an existing MCMC chain, evaluating \( c(x, \theta) \) and accepting or rejecting the proposed \( x \) according to the Metropolis acceptance rule (see the appendix). The MCMC sampler is run in the joint space of \( x \) and \( \theta \), where parameters \( \theta \) are proposed directly from the parameter posterior distribution and controls \( x \) from a separate proposal distribution as in the Metropolis algorithm.

Thus, in addition to using MCMC to produce samples from \( p(\theta|y) \), we use MCMC sampling to explore the \('x\)-space' and approximate \( C(x) \) as well. This might seem contradictory, since we are dealing with an optimization problem instead of a sampling problem. However, using MCMC can add value to optimization: \( C(x) \) might not have a unique optimum and some variables in \( x \) might not even have a significant effect on \( C(x) \). This kind of sensitivity information can be obtained by MCMC.

The mean criterion surface \( C(x) \) might be flat so that it is hard to distinguish the optimal \( x \) from the obtained samples. As noted in (Müller et al. 2004) in an optimal design context, a more peaked surface can be obtained by sampling from an augmented target

\[
\pi(x, \theta_1, ..., \theta_J) = \prod_{i=1}^{J} c(x, \theta_i)p(\theta_i|y)
\]

(6)

that has the marginal distribution \( C(x)^J \). In this case, one has to pick \( (\theta_1, ..., \theta_J) \) at each MCMC iteration randomly from the existing parameter chain. Increasing \( J \) concentrates the samples more tightly around the regions of high \( C(x) \). Increasing \( J \) along the sampling process is analogous to simulated annealing, which is a popular stochastic optimization method. Thus, MCMC sampling can be turned into optimization by increasing \( J \).

The robust mean criterion in equation (3) cannot be expressed as a convenient integral like the expectation, and the MCMC sampling approach used with expectation in equation (5) cannot be directly applied. However, note that the expression \( \text{Std}_{\theta|y}[c(x, \theta)] \) essentially gives weight to deviations of the cost function, \( |c(x, \theta) - c(x, \theta^*)| \). Thus, a similar effect of penalizing large deviations in \( c(x, \theta) \) can be obtained by defining the criterion as

\[
C(x) = \text{Exp}_{\theta|y}[c(x, \theta)] - \alpha \text{Exp}_{p(\theta, \theta^*|y)}[|c(x, \theta) - c(x, \theta^*)|]
\]

(7)

where \( p(\theta, \theta^*|y) \propto p(\theta|y)p(\theta^*|y) \) is the joint distribution of two independent and identically distributed parameter vectors. It is straightforward to write this as an integral, which can be approximated by MCMC sampling from target

\[
\pi(x, \theta, \theta^*) = (c(x, \theta) - \alpha |c(x, \theta) - c(x, \theta^*)|) p(\theta|y)p(\theta^*|y).
\]

(8)

Thus, at each MCMC iteration, one picks two independent random samples \( \theta \) and \( \theta^* \) from the parameter distribution and evaluates the expression \( c(x, \theta) - \alpha |c(x, \theta) - c(x, \theta^*)| \). Naturally, the annealed target (see equation 6 for expectation) can be used with the robust mean criterion as well. This can be done simply by picking \( (\theta_1, ..., \theta_J) \) and \( (\theta_1', ..., \theta_J') \) from the parameter distribution and using MCMC to sample from the augmented target

\[
\pi(x, \theta_1, ..., \theta_J, \theta_1', ..., \theta_J') = \prod_{i=1}^{J} \pi(x, \theta_i, \theta_i').
\]

(9)
The presented MCMC sampling and simulated annealing approaches cannot be implemented for all target function formulations. The requirement is that the criterion can be written as an integral (expectation) over the parameter distribution. For example, the worst-case criterion in equation (4) cannot be written in such a form, and one has to use the direct Monte Carlo approximation.

3 Remarks and Discussion

- Lee and Chen (2009) conduct a comparative study of different methods to quantify uncertainties in different performance measures (model outputs) caused by uncertain model inputs (parameters in our case). They consider direct numerical integration methods, based on quadrature formulas, and more sophisticated methods like the polynomial chaos expansion (PCE) for approximating the distribution of the performance measures. In the comparisons, they use certain fixed forms for the parameter distributions. The starting point of our study is different: we estimate the true, analytically intractable parameter distribution by MCMC from measured data, and study how the output of the (increasingly popular) MCMC estimation can be used as input in optimization tasks. For process optimization after MCMC parameter estimation, Monte Carlo type of approaches are natural, since the parameter distribution is only available as samples, not in closed form. The posterior distribution often does not follow any known parametric form, and approximating the posterior with a certain distribution might give biased parameter estimation and optimization results.

- Another source of uncertainty in process optimization is the variation in ‘external’ operating conditions \( \tilde{x} \) that we cannot control (related to e.g. weather). We would like to control the system so that it works well for a range of possible values for \( \tilde{x} \). While this paper concentrates on handling parameter uncertainty, we note that it might be useful consider also \( \tilde{x} \) as a random variable, with some given density \( p(\tilde{x}) \). The uncertainty in \( \tilde{x} \) can be taken into account by integrating over \( \tilde{x} \). For example, for the mean criterion we can write

\[
C(\tilde{x}) = \int \int c(\tilde{x}, \theta, \tilde{x}) p(\theta | y) p(\tilde{x}) d\theta d\tilde{x}.
\]

The integration over \( \tilde{x} \) can be done using direct Monte Carlo integration or MCMC integration, in a similar way as with model parameters \( \theta \). In MCMC integration, the uncertainty in \( \tilde{x} \) can be incorporated simply by picking \( \tilde{x} \) at each MCMC step from the known distribution \( p(\tilde{x}) \). The distribution \( p(\tilde{x}) \) can be of some parametric form or it can be the empirical distribution (density estimate) obtained from existing measurements.

- When using direct Monte Carlo approximation in evaluating the criteria, one has to decide which samples \( \theta_i \) are used for evaluating \( C(\tilde{x}) \). In practice, one could always pick the same (large) subset of the MCMC chain for the evaluation of \( C(\tilde{x}) \), which would result in a deterministic optimization problem that could be solved with many nonlinear optimization methods. However, the results calculated in this way would be dependent on the chosen parameter subset. Another option is to pick the parameter points randomly from the MCMC chain whenever \( C(\tilde{x}) \) is evaluated. This leads to a stochastic optimization problem: \( C(\tilde{x}) \) evaluated twice with the same \( \tilde{x} \) gives two different values. In this case, one has to use an optimization method that can handle noise in the target function. The amount of noise can be controlled by the number of samples chosen for the Monte Carlo approximation.

- In this paper, the parameter \( \alpha \) that defines the weight given for the variance in \( c(\tilde{x}, \theta) \) in the optimization is chosen by hand. A useful approach might be to consider the problem as a multi-objective
optimization problem with two objectives: large expectation and small deviations. The 'optimal compromises' could be read from the resulting Pareto-optimal front. The challenge in applying multi-objective optimization techniques is the stochastic nature of the optimization problem.

- The downside of the presented approach is, as in many Monte Carlo methods, the large number of model evaluations required. In the simulated annealing approach, the amount of computation per step increases as $J$ is increased. However, parallel computing is directly applicable here: one can evaluate each term in the product in equation (6) independently.

- If parameter estimation is done in a Bayesian way, using for example MCMC, it is natural to ask how the parameter estimation output can be used in other common statistical analyses, such as simulation, experimental design and model-based optimization. Simulation is straightforward: one can simply solve the model with different parameter values given by MCMC estimation and study the uncertainty in model simulations. Recently, also optimal experimental design methods that can utilize MCMC output have been proposed, see e.g. (Müller et al. 2004; Solonen et al. 2011). Thus, these common statistical analyses for mathematical models can now be performed in a unified Bayesian way.

4 Examples

In this section, process optimization using MCMC output is demonstrated with two synthetic examples. Data is simulated with assumed true parameter values, based on which MCMC parameter estimation is performed. Based on the resulting MCMC chain, a process criterion is optimized, both by fixing $\theta$ to its MAP-estimate (least squares estimate) and by taking the possible parameter values given by MCMC into account in the proposed way. In the examples, we use the robust mean criterion.

4.1 Chemical Reaction

Let us consider a simple chemical reaction $A \rightarrow B \rightarrow C$. Compound $A$ transforms into $B$ with rate $k_1(T)$ and $B$ into $C$ with rate $k_2(T)$. The reaction rates depend on temperature $T$. The model is written as an ODE system as

$$
\begin{align*}
\frac{dA}{dt} &= -k_1(T)A \\
\frac{dB}{dt} &= k_1(T)A - k_2(T)B \\
\frac{dC}{dt} &= k_2(T)B.
\end{align*}
$$

The temperature dependency of the reaction rates is defined using the Arrhenius equation

$$
k_i(T) = a_i \exp \left( -\frac{E_i}{R} \left( \frac{1}{T} - \frac{1}{T_0} \right) \right),
$$

where $a_i$ are the reaction rates at the reference temperature $T_0$, $E_i$ are the activation energies of the reactions, and $R$ is the gas constant. In parameter estimation, we want to estimate $\theta = (a_1, a_2, E_1, E_2)$ by measuring $A$ and $B$.

In process optimization, we want to calculate collection time and temperature that maximize the concentration of the intermediate product $B$. Thus, our control variables are $x = (t, T)$ and the criterion function is defined as

$$
c(x, \theta) = B(t, T, \theta)
$$
where $B(t, T, \theta)$ denotes the component $B$ of the solution of the above ODE at time $t$ with temperature $T$ and parameter values $\theta$.

4.1.1 Parameter Estimation

We generate synthetic data, assuming that the true parameter values are $\theta_{\text{true}} = (0.005, 0.001, 4E4, 3.8E4)$. The values are chosen by hand, roughly corresponding to ‘typical’ Arrhenius parameter values in chemical kinetics. Gaussian noise is added to the true model response. Instead of using the model directly, we add noise using $\sqrt{y} = \sqrt{f} + \epsilon_i$. This transformation produces positive measurements, where measurement error increases as the model response increases, see e.g. (Malve et al. 2006). As a reference temperature, we use $T_\text{ref} = 333.15K$. Data is simulated at two temperatures, $T = 313.15K$ and $T = 343.15K$.

The same error structure that was used in data generation is used as the likelihood in MCMC sampling. Thus, we formulate the likelihood as

$$p(y|\theta) \propto \exp\left(-\frac{1}{2\sigma^2}||\sqrt{y} - \sqrt{f(x, \theta)}||^2\right).$$

A priori, we set positivity constraints for the parameters. In data generation, we use $\sigma = 0.5$. For MCMC sampling, we use an effective adaptive MCMC method called Delayed Rejection Adaptive Metropolis (DRAM) of (Haario et al. 2006), see the appendix for some discussion of adaptive MCMC. The DRAM algorithm is run for 10000 iterations.

Simulated data, model fit and parameter posterior $p(\theta|y)$ obtained by MCMC are plotted in figure 1. From the fit one can see that all parameters can be estimated from data with rather good accuracy. However, even this small uncertainty has an effect in the process optimization, as seen in the next section.

![Figure 1: Left: simulated data and predictive distributions calculated from MCMC. Right: pairwise marginals from the parameter posterior distribution with 1d and 2d density estimates.](image)

4.1.2 Optimal Process

Since we have only two control variables in this case and a simple model (the ODE system can be solved analytically), we can demonstrate different optimization criteria by calculating direct Monte Carlo
approximations on a 2D-grid of different values for the two control variables. Here, we compare the optimal process given by a point estimate (least squares estimate) and by the robust mean criterion in equation 8. Let us write $x^*_{\alpha}$ for the optimal process condition given by the robust mean criterion with weight $\alpha$ given for the deviations, and $x^*_{\text{map}}$ for the optimal condition given by using only the MAP estimate for $\theta$.

For evaluating the robust mean criterion (equation 3), we pick 1000 samples from the parameter distribution. We calculate the different optimization criteria in a 100x100 grid that has evenly based points in intervals $t \in [5,100]$ and $T \in [60,160]$. In figure 2, the robust mean criterion with different values for $\alpha$ and the criterion value with the MAP estimate are compared. The results differ significantly for varying values for $\alpha$. The optimum with the MAP estimate is at $x^*_{\text{map}} = (10.8, 143.8)$. With larger values for $\alpha$, the optimum does not change much from $x^*_1$.

Using the MAP estimate only in the optimization suggests that in order to get maximal product yield, the temperature should be set high and the product should be collected soon after the process starts. Optimization with the robust mean criterion tells another story: it is better to put the temperature to a lower value and to wait for a bit longer before collecting the product. This difference is caused by the fact that, based on the parameter estimation, we do not know accurately how the system behaves, especially at high temperatures where we have no measurements. To illustrate this, the distributions of model responses with $x^*_{\text{map}}$ and $x^*_1$ are compared in figure 4. The distribution of the product yields with different values for $\alpha$ are given in figure 3.

Note that taking the parameters into account as a distribution can yield different solutions for the process optimization problem, even if the parameter uncertainty is relatively small (as in this toy example). In real situations, one often cannot estimate parameters as well as in this example. The more uncertainty there is in parameter estimates, the more important it is to consider the optimized quantity as a distribution instead of a point estimate.

### 4.1.3 Comparison to the Classical Approach

From the MCMC output, one can study the sensitivity of the process optimum found with different methods, see e.g. figure 3, but the analysis does not tell how one method compares to another in real life. In synthetic cases, however, the 'true' behavior of the model is known, and one can compare how the MCMC approach performs compared to the classical approach, where a fixed point estimate is used and the parameter uncertainty is neglected.

Here, we perform such an experiment as follows. First, synthetic data is generated, as in the previous sections, by adding noise to the model simulated with 'true' parameter values. Then, parameters are estimated using both least squares and MCMC, and the optimization is performed using the obtained estimates. Thus, in the least squares approach the obtained point estimate is fixed at the optimization stage, and in the MCMC approach the parameter uncertainty is taken into account as in the previous section. Finally, the obtained optimization results are plugged in to the true model (model with assumed true parameters) to see how large product yields were actually obtained with the two methods. The above procedure is repeated a number of times (here 400) to get statistics for the performance comparison.

The results of the comparison are illustrated in figure 5. For the 400 test cases, the proposed approach that takes parameter uncertainty into account gives robust results, whereas using the point estimate only can lead to poor product yield. For instance, the product yield was below 60 in 33/400 cases when the point estimate was used, but only in 1/400 cases when the MCMC approach was applied. Note that it is possible that the least squares approach gives better results for individual cases, e.g. if the obtained point estimate happens to be very close to the 'true' value. In the long run, however, it pays off to take the uncertainty into account.

8
Figure 2: Criterion surfaces with the MAP estimate (top left) and with the robust mean criterion with different values for $\alpha$. Temperatures are in Celsius.

Figure 3: Distributions of product yield at MAP-optimal conditions (blue) and with the robust mean criterion (red and green).
Figure 4: Distribution of model response at the MAP-optimal conditions (left) and at the optimal temperature calculated with the robust mean criterion (right). Two confidence envelopes, 50% (dark gray) and 99% (light gray) are plotted.

Figure 5: The optimal product yield (red) and the actual product yield obtained using the proposed approach (green) and the classical way (blue).
4.2 Williams-Otto Process

In this example, we consider the well-known Williams-Otto plant, introduced in (Williams and Otto 1963), which is a commonly used benchmark problem in process optimization. The process consists of an ideal stirred tank reactor, in which three elementary chemical reactions take place:

\[
\begin{align*}
A + B & \xrightarrow{r_1} C \\
C + B & \xrightarrow{r_2} P + E \\
P + C & \xrightarrow{r_3} G.
\end{align*}
\]

The raw materials \(A\) and \(B\) are fed into the reactor with rates \(F_A\) and \(F_B\). The reaction mixture is directed into a cooler with rate \(F_R\), after which the residue \(G\) is totally separated in a decanter. The remaining mixture is fed into a distillation column, where the desired product \(P\) is separated. The recovery of \(P\) is not complete and some \(P\) is left in the bottom of the distillation column. A fraction of the bottom product is recycled back to the reactor (flow \(F_T\)) and the rest is removed from the process. The process is given as a flowchart in figure 6.

![Figure 6: The Williams-Otto plant as a flowchart. Figure provided by prof. Ugur Akman.](image)

The reaction rates are given by

\[
\begin{align*}
r_1 &= k_1 F_A F_B V / F_R^2 \\
r_2 &= k_2 F_B F_C V / F_R^2 \\
r_3 &= k_3 F_C F_P V / F_R^2
\end{align*}
\]

where \(F_A\), \(F_B\), \(F_C\) and \(F_P\) are the flow rates in the reactor. Reactor volume and density of the reaction mixture are denoted by \(V\) and \(\rho\) respectively. The reaction rate coefficients follow the Arrhenius equation:

\[
k_i(T) = q_i \exp \left(-b_i \left(\frac{1}{T} - \frac{1}{T_0}\right)\right)
\]

where \(T\) is the reactor temperature and \(T_0\) the reference temperature. The kinetic parameters \(\theta = (q, b)\) are estimated from measurements (details below).
Finally, the model can be written as a steady state model using the flow rates. Here we write the model separately for different parts of the plant. First, for the reactor we write

\[
\begin{align*}
F_A + F_{TA} - F_{RA} - r_1 &= 0 \\
F_B + F_{TB} - F_{RB} - r_1 - r_2 &= 0 \\
F_{TC} - F_{RC} + 2r_1 - 2r_2 - r_3 &= 0 \\
F_{TE} - F_{RE} + 2r_2 &= 0 \\
F_{TG} - F_{RG} + 1.5r_3 &= 0 \\
F_{TP} - F_{RP} + r_2 - 0.5r_3 &= 0.
\end{align*}
\]

The equations in the decanter are trivial. In the distillation column, some product is left in the bottom of the column, since \( P \) forms an azetrope with \( E \). In the Williams-Otto plant, the amount of \( P \) retained in the column is taken to be 10% of the flow rate of \( E \). Thus, we write

\[
\begin{align*}
F_P &= F_{SP} - 0.1F_{SE} \\
F_{YP} &= 0.1F_{SE}.
\end{align*}
\]

In the splitter, a fraction \( \alpha \) of the total flow is removed from the process and the rest is recycled back to the reactor. We write

\[
\begin{align*}
F_{Ti} &= (1 - \alpha)F_{Yi} \\
F_{Di} &= \alpha F_{Yi}
\end{align*}
\]

where \( i = (A, B, C, P, E) \). For the flow rates, the model is a nonlinear system of algebraic equations and can be solved with numerical methods.

### 4.2.1 Parameter Estimation

To estimate the kinetic parameters \( \theta = (q, b) \), the chemical reaction is studied separately by running an experiment in a separate batch reactor with no flows. The kinetic system is now written as an ODE system:

\[
\begin{align*}
\frac{dA}{dt} &= -k_1AB \\
\frac{dB}{dt} &= -k_1AB - k_3CB \\
\frac{dC}{dt} &= 2k_1AB - 2k_2CB - k_3CP \\
\frac{dP}{dt} &= k_2CB - 0.5k_3CP \\
\frac{dE}{dt} &= 2k_2CB \\
\frac{dG}{dt} &= 1.5k_3PC.
\end{align*}
\]

Let us assume that we can measure the concentration of components \( A \) and \( B \) at different temperatures. We create synthetic measurements by adding normally distributed random noise to model responses with assumed ‘true’ parameter values given in table 1. To get the model response, we solve the ODE system numerically, using initial values \( A(0) = B(0) = 1 \) and \( C(0) = P(0) = E(0) = G(0) = 0 \). We simulate three batches of data in temperatures 320K, 340K and 360K and use 350K as our reference temperature \( T_0 \). The reactor is run for 0.1 hours in each case and the concentrations of the compounds are collected.
in 20 evenly spaced times. An MCMC chain of length 10000 is created from the parameter distribution using the DRAM algorithm, see (Haario et al. 2006) and the appendix for DRAM details. Figures 7 and 8 show the distribution of the parameters and the model fit. Note that again we have a very idealized situation: in real problems, the data and parameter identifiability are often worse. As shown in the next section, even this seemingly harmless parameter uncertainty has a clear impact on process optimization results.

Table 1: True parameter values for the Williams-Otto plant, obtained from (Kaijaluoto 1984).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(q_1)</td>
<td>31.9223</td>
</tr>
<tr>
<td>(b_1)</td>
<td>6666.67</td>
</tr>
<tr>
<td>(q_2)</td>
<td>118.5762</td>
</tr>
<tr>
<td>(b_2)</td>
<td>8.333e+03</td>
</tr>
<tr>
<td>(q_3)</td>
<td>157.1899</td>
</tr>
<tr>
<td>(b_3)</td>
<td>1.111e+04</td>
</tr>
</tbody>
</table>

Figure 7: Posterior distribution of the kinetic parameters.

4.2.2 Process Optimization

Given a specific parameter value, we can use the steady-state model and solve the system in terms of flow rates. Our control variables are the input flow rates of the raw materials, the reactor temperature and the purge fraction: \(\mathbf{x} = (F_A, F_B, T, \alpha)\).
As a cost function, we use "return of investment", a similar cost function that was used in (Kaijaluoto 1984). In practice, raw materials and disposal of unwanted products cost money and profit is obtained from the desired product \( P \). The cost function is written as

\[
c(x, \theta) = \frac{(84A_{\text{in}} - 201.96\dot{n}_D - 336\dot{n}_G + 1995.52\dot{n}_P - 2.22\dot{n}_R)}{(6V\rho)}.
\]

We use the robust mean criterion and MCMC integration for process optimization (equation 8). The estimation is run with \( \alpha = 0.5, \alpha = 1 \) and \( \alpha = 3 \). We use the annealed target with \( J = 40 \) to obtain enough contrast in the surface \( C(x) \). The robust mean criterion is given in figure 9 as an example for \( \alpha = 1 \).

When using the MAP estimate \( \hat{\theta} \) only (traditional approach), we obtain optimal conditions \( x_{\text{map}}^* = [6489, 15589, 379.3, 0.1002] \) that gives objective function value \( c(x_{\text{map}}^*, \hat{\theta}) = 144.3 \). The results agree well with the ones obtained in (Kaijaluoto 1984). The optimal conditions \( x_{\alpha}^* \) given by the robust mean criterion with different values for \( \alpha \) are \( x_{0.5}^* = [6415, 12390, 370.5, 0.081] \), \( x_{1}^* = [6016, 13760, 367.6, 0.081] \) and \( x_{3}^* = [4395, 8034, 358.6, 0.0556] \). The distributions of the cost functions with the robust mean criterion and the MAP-optimal point are illustrated in figure 10.

It turns out that the distribution of the objective function value, given the uncertainty in the reaction kinetics, is rather wide, indicating that the objective function is sensitive to the values of the kinetic parameters (see figure 10). With the results given by the robust mean approach, the risk of obtaining poor (even negative) return of investment is minimized. However, putting more weight to small deviations comes at a cost: the expected value decreases slightly as the deviations are emphasized.

5 Conclusions

We have presented a way to incorporate parameter uncertainty into process optimization. The approach utilizes the output from MCMC methods that are more and more routinely used in statistical model fitting problems. The same MCMC methods can also be used in calculating the optimization criteria effectively, instead of direct Monte Carlo approximations. We have tested the approach in two simple chemistry examples. Even in these simple cases, with small parameter uncertainties, the difference to the
Figure 9: Robust mean criterion surface with $\alpha = 1$ and the MAP-optimal conditions (blue dot).

Figure 10: Distributions of product yield at MAP-optimal conditions (blue) and optimal conditions calculated with the robust mean criterion (red, green, yellow).
traditional approach is clearly visible. With the presented approach it is possible to reduce the risk of obtaining poor process optimization results caused by incomplete knowledge of the process.

Acknowledgements

We wish to thank Prof. Ugur Akman for providing the flowchart figure of the Williams-Otto process. Mr. Kalle Riihimäki is acknowledged for suggesting the Williams-Otto experiment.

This work was funded by the Finnish Center of Excellence in Inverse Problems Research consortium (Academy of Finland).

Appendix

In this appendix, we briefly summarize the basic concepts and algorithms of Bayesian parameter estimation and MCMC. We also give a simple example that demonstrates the difference of the MCMC approach to classical nonlinear parameter estimation procedures.

In the Bayesian approach the unknown parameter vector is interpreted as a random variable. The aim of the analysis is to find its distribution. Before experimental data is available the parameter \( \theta \) has a prior distribution \( p(\theta) \). The observations \( y \) update the distribution \( p(\theta) \) to the posterior distribution by the Bayes formula

\[
\pi(\theta) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta) \, d\theta},
\]

(15)

Here \( p(y|\theta) \) is the likelihood function that gives the likelihood of data \( y \) for given parameter value \( \theta \). The posterior distribution \( \pi(\theta) = p(\theta|y) \) gives the probability distribution of parameter values, given the measured data \( y \). The integral \( \int p(y|\theta)p(\theta) \, d\theta \) is needed as the normalizing constant, to ensure that \( \pi \) indeed is a probability measure, with total measure equal to one, \( \int \pi(\theta) \, d\theta = 1 \).

In the usual settings the parameter vector \( \theta \) and data \( y \) are connected by a model \( y = f(x, \theta) + \epsilon \), where the experimental error \( \epsilon \sim N(0, \sigma^2I) \), i.e., in all experiments the measurement noise is Normally distributed, independent and with standard deviation of size \( \sigma \). It is not difficult to see that then

\[
p(y|\theta) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\sum_{i=1}^{n}(y_i - f(x_i, \theta))^2/2\sigma^2\right).
\]

So, we arrive at the familiar least squares function: maximizing the likelihood function turns out to be equivalent to minimizing the residual sum of squares.

In principle, the Bayes formula solves the estimation problem in a fully probabilistic sense: we find the peak, the maximum a posteriori (MAP) point, of the parameter distribution. Then we determine a required portion of the probability mass (typically some 95% or 99% of the mass) around it. However, we face the problem of how to define the a priori distribution, and how to calculate the integral of the normalizing constant. Often, only a 'flat' prior is used, that is, a uniform distribution that only defines physically possible lower and upper bounds for each parameter. However, the integration of the normalizing constant often is a formidable task, even for only moderately high number of parameters in a nonlinear model. So, a direct application of the Bayes formula is intractable for all but trivial nonlinear cases. The MCMC methods provide a tool to handle this problem. They generate a sequence of parameter values \( \theta_1, \theta_2, ..., \theta_N \), whose empirical distribution approximates the true posterior distribution for large enough sample size \( N \).

The trick here is that we do not know the distribution from which to sample, but we still can generate samples from it. Instead of sampling from the true distribution, we only may sample from an artificial proposal distribution. Combining the sampling with a simple accept/reject procedure, the posterior
can be correctly approximated. The simplest MCMC method is the Metropolis algorithm introduced in (Metropolis et al. 1953):

1. Initialize by choosing a starting point \( \theta_1 \).

2. Choose a new candidate \( \hat{\theta} \) from a suitable proposal distribution \( q(.|\theta_n) \) that may depend on the previous point of the chain.

3. **Accept** the candidate with probability

   \[
   \alpha(\theta_n, \hat{\theta}) = \min\left(1, \frac{\pi(\hat{\theta})}{\pi(\theta_n)}\right).
   \]

   If rejected, repeat the previous point in the chain. Go back to item 2.

So, points with \( \pi(\hat{\theta}) > \pi(\theta_n) \), i.e., steps 'uphill', are always accepted. But also points with \( \pi(\hat{\theta}) < \pi(\theta_n) \), i.e., steps 'downhill', may be accepted, with probability that is given by the ratio of the \( \pi \) values. In practice, this is done by generating a uniformly distributed random number \( u \in [0,1] \) and accepting \( \hat{\theta} \) if \( u \leq \pi(\hat{\theta})/\pi(\theta_i) \). Note that only the ratios of \( \pi \) at consecutive points are needed, so the main difficulty is omitted: the calculation of the normalizing constant is not needed since the constant cancels out!

However, the choice of the proposal distribution may still pose a problem. It should be chosen so that the 'sizes' of the proposal \( q \) and target distributions suitably match. This often may be difficult to achieve. An unsuitable proposal leads to inefficient sampling, typically due to

- the proposal being too large. Then the new candidates mostly miss the essential region of \( \pi \); they are chosen at points where \( \pi \approx 0 \) and only rarely accepted.

- the proposal being too small. The new candidates mostly are accepted, but from a small neighborhood of the previous point. So, the chain moves only slowly, and may not cover the target \( \pi \) in finite number of steps.

For simple cases, the proposal might be relatively easy to find by some hand-tuning. However, the 'size' of the proposal distribution is not a sufficient specification. In higher dimensions, especially, the shape and orientation of the proposal are crucial. The most typical proposal is a multi-dimensional Gaussian (Normal) distribution. In the random walk version, the center point of the Gaussian proposal is chosen to be the current point of the chain. The task then is to find a covariance matrix that produces efficient sampling.

Several efficient adaptive methods have been recently proposed, see, e.g., the Adaptive Metropolis (AM) and the Delayed Rejection Adaptive Metropolis (DRAM) algorithms (Haario et al. 2001, 2006). In adaptive MCMC, one uses the sample history to automatically tune the proposal distribution 'on-line' as the sampling proceeds. In AM, the empirical covariance from the samples obtained so far is used that as the covariance of a Gaussian proposal. The simplicity of AM adaptation allows for its use in more advanced adaptation schemes. The DRAM algorithm combines the delayed rejection (DR) method by Mira (2001) with AM. This DRAM method 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 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 this paper, the DRAM method was used for all sampling tasks. The DRAM calculations were performed by using the Matlab adaptive MCMC Toolbox, that may be downloaded at http://www.helsinki.fi/~mjlaine/mcmc/.
Here, the difference between classical nonlinear regression analysis and Bayesian model fitting is illustrated using a simple toy example. The model \( y = \theta_1(1 - \exp(-\theta_2 x)) \), used to model e.g. the biological oxygen demand, is fitted to data \( x = (1, 3, 5, 7, 9) \), \( y = (0.076, 0.258, 0.369, 0.492, 0.559) \). First, the parameters are estimated using the classical least squares approach, and the covariance of the parameters is approximated using a linearization around the least squares estimate \( \hat{\theta} \). In practice the covariance is estimated as \( \text{Cov}(\hat{\theta}) = \sigma^2(J^T J)^{-1} \), where \( J \) is the Jacobian of the parameters evaluated at \( \hat{\theta} \) and \( \sigma^2 \) is the measurement error variance (assumed here to be i.i.d. Gaussian). The measurement error variance was estimated from the residuals, giving \( \sigma = 0.014 \).

Then, the DRAM sampler is run for 10000 steps to get samples from the posterior distribution. In figure 11, the parameter distributions obtained with the classical linearization-based way and with the MCMC approach are presented. One can see how the classical approach can go wrong, if the parameter distribution is not Gaussian. In the figure, the predictive distribution calculated from the MCMC output is also given, illustrating that extending the uncertainty analysis to predictions (and other functions of the parameters) can be easily achieved simply by simulating the model with different parameter values given by MCMC.

![Figure 11](image-url)

Figure 11: Left: parameter posterior with MCMC (blue) and linearization around the LSQ estimate (red). Right: predictive distributions with MCMC and the single prediction produced by the LSQ estimate. Gray colors correspond to 50%, 80%, 95% and 99% confidence envelopes.
References


Variational Ensemble Kalman Filtering Using Limited Memory BFGS

Abstract. The extended Kalman filter (EKF) is one of the most used nonlinear state estimation methods. However, in large-scale problems, the CPU and memory requirements of EKF are prohibitively large. In [2, 3], a promising approximation to EKF, called the Variational Kalman Filter (VKF), was proposed. The implementation of VKF requires the tangent linear and adjoint codes for propagating error covariances in time. However, the trouble of building the codes can be circumvented by using Ensemble filtering techniques, where an ensemble of states is propagated in time using the full nonlinear model, and the statistical information needed in EKF formulas is estimated from the ensemble. In this paper, we show how the VKF ideas can be used in the ensemble filtering context. Following VKF, we obtain the state estimate and its covariance by solving a minimization problem using the limited memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) method (see e.g. [29]), which provides low-storage approximations to the state covariances. The resulting hybrid method, the Variational Ensemble Kalman Filter (VEnKF), has several attractive features compared to existing ensemble methods. The model error and observation error covariances can be inserted directly into the minimization problem, instead of randomly perturbing model states and observations as in the standard Ensemble Kalman filter. New ensembles can be directly generated from the LBFGS covariance approximation, without the need of a square root (Cholesky) matrix decomposition. The frequent resampling from the full state space circumvents the problem of ensemble in-breeding frequently associated with Ensemble filters. Numerical examples are used to show that the proposed approach performs better than the standard Ensemble Kalman filter, especially when the ensemble size is small.

Key words. Data assimilation, state estimation, Kalman filtering, ensemble filtering, LBFGS.

1. Introduction. Since the introduction of the Kalman filter (KF) in [22] and its nonlinear extension, the extended Kalman filter (EKF), many approaches to overcome the problems with computational complexity present in KF and EKF have been proposed, see e.g. [10, 7, 14]. In these approaches, the state is often projected onto a fixed, low-dimensional subspace. It is known that the fixed projection operator might not correctly capture the dynamics of a nonlinear system, see [19]. In particular such reduced rank Kalman filters tend to suffer from covariance leakage, see [21].

In [2, 3], a low-storage variational approach to approximate KF and EKF was proposed, called the Variational Kalman Filter (VKF). In VKF, the large matrices in KF formulas were replaced with low-storage approximations provided by the quasi-Newton optimization method called limited memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS). The applicability of VKF is hindered by the requirement of tangent linear and adjoint codes for the evolution model, which require a considerable development effort separately for every model.

In ensemble filtering methods, the problems related to standard EKF (large matrices, need for tangent linear and adjoint codes) are circumvented by representing uncertainty in the model state as a number samples instead of covariance matrices. Instead of moving the covariance in time using the linearized model, uncertainty is propagated simply by moving the ensemble members in time with the full nonlinear evolution model. The simplest version of this idea is the Ensemble Kalman Filter (EnKF), first proposed in [12], where the covariance matrices in the KF formulas are essentially replaced with sample statistics calculated from the ensemble. However, EnKF suffers from some problems, e.g. sampling errors due to random perturbation of model state and observations, and from ensemble in-breeding that results in similar covariance leakage as that associated with reduced rank Kalman filters, see [30], [27], [24].

In this paper, we show how VKF ideas can be used in the ensemble filtering context to overcome some problems related to existing ensemble methods. In our approach, the model error and observation error covariances are inserted directly into the minimization problem, instead of randomly perturbing model states and observations as in EnKF. New ensembles can be efficiently generated directly from the LBFGS covariance approximation, without explicitly constructing the large covariance matrix. We call our hybrid approach the Variational Ensemble Kalman Filter (VEnKF), and show by numerical examples that the hybrid method performs well compared to the standard EnKF, especially when the ensemble size is small.

The paper is organized as follows. In section 2, we recall the basics of Kalman filtering and ensemble methods. We introduce the VEnKF algorithm in section 3 and demonstrate it with numerical examples in section 4. In section 5, we discuss some specific topics related to our approach and the differences to existing ensemble filters. Section 6 concludes the paper.

2. Filtering Methods. In this section, we provide an overview of some existing Kalman filtering methods that are related to our approach. We start by recalling how the basic Kalman filter and some of its variants work, and continue with an introduction to ensemble filtering methods.
2.1. Kalman Filtering and Variants. The Kalman Filter can be used to estimate the state $x_k$ at discrete times $k$ from observations $y_k$, when the model and observation equations are linear:

\begin{align}
\mathbf{x}_k &= \mathbf{M}_k \mathbf{x}_{k-1} + \mathbf{e}_k^m \\
\mathbf{y}_k &= \mathbf{K}_k \mathbf{x}_k + \mathbf{e}_k^o.
\end{align}

In the above system, $\mathbf{M}_k$ is the $d \times d$ evolution model and $\mathbf{K}_k$ is the $m \times d$ observation operator. The $d \times 1$ vector $x_k$ represents the model state and observed data is denoted by the $m \times 1$ vector $y_k$. Model error $\mathbf{e}_k^m$ and observation error $\mathbf{e}_k^o$ are assumed to be normally distributed zero mean random variables with covariance matrices $\mathbf{C}_m$ and $\mathbf{C}_o$, respectively. The Kalman filter algorithm for estimating states and their error covariances can be written as follows.

The Kalman Filter algorithm
1. Move the state estimate and covariance in time:
   (a) Compute $\mathbf{x}_k^\text{est} = \mathbf{M}_k \mathbf{x}_{k-1}^\text{est}$.
   (b) Compute $\mathbf{C}_k = \mathbf{M}_k \mathbf{C}_{k-1} \mathbf{M}_k^T + \mathbf{C}_m$.
2. Combine the prior with observations:
   (a) Compute the Kalman gain $\mathbf{G}_k = \mathbf{C}_k(\mathbf{K}_k \mathbf{C}_o + \mathbf{C}_m)^{-1}$.
   (b) Compute the state estimate $\mathbf{x}_k^\text{est} = \mathbf{x}_k^\text{est} + \mathbf{G}_k(\mathbf{y}_k - \mathbf{K}_k \mathbf{x}_k^\text{est})$.
   (c) Compute the covariance estimate $\mathbf{C}_k^\text{est} = \mathbf{C}_k^\text{est} - \mathbf{G}_k \mathbf{K}_k \mathbf{C}_k^\text{est}$.
3. Set $k \to k + 1$ and go to step 1.

The extended Kalman filter directly uses the Kalman filter formulas in the nonlinear case by replacing the nonlinear model and observation operators with linearizations: $\mathbf{M}_k = \partial \mathbf{M}(\mathbf{x}_{k-1}^\text{est})/\partial \mathbf{x}$ and $\mathbf{K}_k = \partial \mathbf{K}(\mathbf{x}_k^\text{est})/\partial \mathbf{x}$.

In the variational formulation of the Kalman filter, the state estimation at step $k$ is viewed as an optimization problem, where a quadratic function

\begin{equation}
l(\mathbf{x}|\mathbf{y}_k) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_k^o)^T(\mathbf{C}_m^{-1})^{-1}(\mathbf{x} - \mathbf{x}_k^o) + \frac{1}{2}(\mathbf{y}_k - \mathbf{K}(\mathbf{x}))^T(\mathbf{C}_o)^{-1}(\mathbf{y}_k - \mathbf{K}(\mathbf{x}))
\end{equation}

is minimized with respect to $\mathbf{x}$. In the VKF method, introduced in [2], the minimization is done with the LBGFS optimization method, that produces both the state estimate and a low-storage approximation of the covariance (inverse hessian at the minimizer). In the VKF algorithm, the inverse of the prior covariance $\mathbf{C}_m^{-1}$ is also approximated using LBFGS by setting up an auxiliary optimization problem (with a trivial solution), given as

\begin{equation}
\arg\min_{\mathbf{B}_k^\theta} \frac{1}{2} \mathbf{u}^T \mathbf{C}_m^{-1} \mathbf{u}.
\end{equation}

Thus, the LBFGS optimization routine provides low-storage approximation for both $\mathbf{C}_m^{-1}$ and $\mathbf{C}_o$. All computations with the covariances can be carried out efficiently using the implicit low-storage representation, without forming the full matrices. The VKF method is given as an algorithm below.

The Variational Kalman Filter algorithm
1. Move the state estimate and covariance in time:
   (a) Compute $\mathbf{x}_k^\text{est} = \mathbf{M}_k \mathbf{x}_{k-1}^\text{est}$.
   (b) Define $\mathbf{C}_k^\text{est} = \mathbf{M}_k \mathbf{B}_k^\theta \mathbf{M}_k^T + \mathbf{C}_m$.
   (c) Apply LBFGS to (2.4) to get an approximation $\mathbf{B}_k^\theta$ of $(\mathbf{C}_m^{-1})^{-1}$.
2. Combine the prior with observations:
   (a) Minimize expression (2.3) using LBFGS to get the state estimate $\mathbf{x}_k^\text{est}$ and covariance estimate $\mathbf{B}_k^\theta$.
3. Set $k \to k + 1$ and go to step 1.

Note that while VKF can solve the storage problem related to EKF, it requires a way to evolve the covariance in time (step 1(b) in the algorithm above). Propagating the covariance using a direct linearization, as in EKF, is infeasible in high dimensions. In VKF, covariance propagation is done using tangent linear and adjoint codes, that implement differentiation in the ‘code level’. This is a standard technique in variational data assimilation, see e.g. [23] and [16]. These codes must be prepared separately for every model and their construction is laborious, although automatic code generators have been recently developed, see e.g. [9]. In ensemble filters, that we discuss next, tangent linear and adjoint codes are not needed.
2.2. Ensemble Filtering. In ensemble filtering, the uncertainty in the state estimate \( x_k \) is represented as \( N \) samples instead of a covariance matrix, here denoted as \( s_k = (s_{k,1}, s_{k,2}, \ldots, s_{k,N}) \). The first ensemble filtering method was the ensemble Kalman filter (EnKF), introduced in [12] and implemented in operational numerical weather prediction e.g. in [20]. The ensemble Kalman filter essentially replaces the state covariance matrices in EKF with the sample covariance calculated from the ensemble. The sample covariance can be written as \( \text{Cov}(s_k) = X_k X_k^T \), where

\[
X_k = ((s_{k,1} - \bar{x}), (s_{k,2} - \bar{x}), \ldots, (s_{k,N} - \bar{x})) / \sqrt{N - 1}.
\]

The sample mean is denoted by \( \bar{x} \). Using our notation, the EnKF algorithm can be formulated as follows.

The Ensemble Kalman Filter algorithm
1. Move the state estimate and covariance in time:
   (a) Move ensemble forward and perturb members with model error:
   \[
s_{k,i} = M(s_{k,i-1}) + e_{k,i}, \quad i = 1, \ldots, N.
   \]
   (b) Calculate sample mean \( \bar{x} \) and covariance \( C_p = X_k X_k^T \).
2. Combine the prior with observations:
   (a) Compute the Kalman gain \( G_k \).
   (b) Update ensemble members
   \[
s_{k,i} = s_{k,i} + G_k(y_k - X_k s_{k,i}^p + e_{k,i}),
   \]
   (c) Calculate state estimate as the sample mean \( \bar{x} \).

In the above algorithm, vectors \( e_{k,i} \) and \( e_{k,i}^p \) are realizations of the model error and observation error distributions (Gaussians with covariances \( C_{e,e} \) and \( C_{o,o} \), respectively).

The Ensemble Kalman Filter is very simple to implement and it does not require tangent linear and adjoint codes. However, EnKF has various problems and numerous variants have been developed to overcome these issues, see e.g. [20], [34], [1], [11]. In Section 5, we discuss these variants in light of the proposed VEnKF algorithm, which we introduce in the next Section.

3. Variational Ensemble Kalman Filtering. Here we follow the VKF ideas and show how they can be implemented in the ensemble filtering context, resulting in a filter that we call the Variational Ensemble Kalman Filter (VEnKF). As in VKF, the state estimation in VEnKF is based on minimizing the cost function in equation (2.3). The prior covariance needed in the cost function is defined here as

\[
C_p = \text{Cov} (M(s_{k-1}^\text{ext}) + e^p_k) = \text{Cov} (M(s_{k-1}^\text{ext})) + \text{Cov} (e^p_k) \approx X_k X_k^T + C_{e,e}.
\]

Note that the above formula contains the common assumption, that the model error and model response are uncorrelated. In VEnKF, we calculate the sample covariance using the state estimate evolved from the previous time as the expectation, instead of the sample mean used in EnKF. Thus, we define

\[
X_k = ((s_{k,1} - x_k^p), (s_{k,2} - x_k^p), \ldots, (s_{k,N} - x_k^p)) / \sqrt{N},
\]

where \( x_k^p = M(s_{k-1}^\text{ext}) \) and \( s_{k,i} = M(s_{k-1,i}^\text{ext}) \). Note that now the ensemble members do not contain random perturbations; the model error is added directly in equation (3.1).

The inverse of the prior covariance \( C_p = X_k X_k^T + C_{e,e} \), needed when evaluating the cost function (2.3), can be obtained in two ways. Following VKF derivation, we can approximate the inverse by applying LBFGS to the artificial optimization problem

\[
\arg \min_u u^T(X_k X_k^T + C_{e,e}) u.
\]

The sample covariance matrix in the above expression naturally does not have to be handled as a full matrix - in order to evaluate the cost function, we can just keep the covariance in the 'ensemble form' \( X_k X_k^T \) and evaluate the cost function in the form \( u^T X_k X_k^T u + u^T C_{e,e} u \). For the computation to remain efficient, we assume (as in VKF) that the model error covariance \( C_{e,e} \) can be efficiently multiplied with a vector, which is the case e.g. if the covariance is assumed to be diagonal. As a result of the above optimization, we obtain an LBFGS representation of the inverse of the prior covariance, \( (C_p)^{-1} \). We can use the LBFGS representation to evaluate the first term when optimizing the cost function (2.3). For computing the matrix-vector product when the matrix is in the LBFGS form, there exists an efficient recursive algorithm, see appendix A and [29] for details.
An attractive alternative way to proceed is to calculate the inverse of the prior covariance using the Sherman-Morrison-Woodbury (SMW) matrix inversion formula, see e.g. [19]. The inverse of the prior covariance can be written as

\[
(C_p^\epsilon)^{-1} = (X_k X_k^T + C_p^\epsilon)^{-1}
\]

\[
= C_p^{-1} - C_p^{-1} X_k (I + X_k^T C_p^{-1} X_k)^{-1} X_k^T C_p^{-1}.
\]

This formulation of the inverse of the prior covariance can be directly inserted into equation (2.3) when it it minimized. The computation of the quadratic expression \((x - x_0) (C_p^\epsilon)^{-1} (x - x_0)^T\) can be organized so that we do not have to store full matrices of size \(d \times d\). With this formulation, the first LBFGS approximation can be avoided and the prior can be included 'exactly' in the second optimization. The application of the formula requires that the inverses of the model error covariances \(C_i^\epsilon\) are available. If the model error is assumed to be constant (same for all \(i\)), this matrix inversion needs to be computed only once. In addition, the inversion of \(I + X_k^T C_p^{-1} X_k\) needs to be computed at every step. However, this matrix is only \(N \times N\) where \(N\) is the ensemble size, which is always very small compared to the dimension of the state space in large-scale applications. In the examples of this paper, we use the SMW formula for inverting the prior covariance.

When LBFGS optimization is applied to minimize function (2.3), we get a low-storage approximation for the covariance \(C_p^\epsilon\). After that, we sample a new ensemble of state vectors from \(N(x_0^m, C_p^\epsilon)\). Samples can be drawn efficiently, since the LBFGS representation for \(C_p^\epsilon\) can be written in the form

\[
C_p^\epsilon = B_0 b_0^T + \sum_{i=1}^{n} b_i b_i^T,
\]

where \(B_0\) is a \(d \times d\) matrix and \(b_i\) are \(d \times 1\) vectors. From this representation one can produce a zero mean random vector \(r \sim N(0, C_p^\epsilon)\) simply by calculating

\[
r = B_0 z + \sum_{i=1}^{n} \omega_i b_i,
\]

where \(z \sim N(0, I)\) and \(\omega_i \sim N(0, 1)\). The matrix \(B_0\) does not have to be stored explicitly, since the product \(B_0 z\) can be computed implicitly using the stored LBFGS vectors. See appendix A for details about constructing \(B_0\) and \(b_i\).

Finally, we are ready to present our Variational Ensemble Kalman Filter (VEnKF) as an algorithm. After setting the initial guesses for the state and its covariance to \(x_0^m\) and \(C_0^\epsilon\) respectively, and setting \(k = 1\), we write our algorithm as follows:

**The VEnKF algorithm**

1. Move the ensemble forward and build the prior:
   a. Compute prior center point \(x_0^k = M(x_0^{m,k})\).
   b. Compute prior ensemble \(x_{i,0}^k = M(x_{i,0}^{m,k})\), \(i = 1, ..., N\).
   c. Define \((C_p^k)^{-1}\) using SMW formula (3.5) (Alternatively: Apply LBFGS to (3.3) to get \((C_p^k)^{-1})\).

2. Calculate the posterior estimate and generate the new ensemble:
   a. Apply LBFGS to minimize (2.3) to get \(x_k^m\) and \(C_k^\epsilon\).
   b. Sample new ensemble \(x_{i,k}^m \sim N(x_k^m, C_k^\epsilon)\), see Appendix A for details.

3. Set \(k \rightarrow k + 1\) and go to step 1.

Since the optimization tasks are both quadratic, only a quadratic version of the LBFGS method is needed. The LBFGS algorithm is given in appendix A, for detailed analysis refer to e.g. [29].

The proposed VEnKF algorithm has several attractive features. First of all, it does not suffer from ensemble in-breeding that is often encountered with many ensemble filtering methods, since the new ensemble is re-generated at each step. The ensembles are generated from dynamically changing covariances that are not restricted to any fixed subspace as in reduced rank methods. The ensemble generation is performed directly by the low storage LBFGS construction, without the need of any further matrix (Cholesky) decomposition. Moreover, the model error covariance term (representing both model error and natural variability) can be added explicitly to the optimization problem in step 1(c) of the algorithm, whereas in EnKF it is handled by randomly perturbing the prior ensemble.
with the model error. In addition, VEnKF uses the state estimate given by the optimizer as the expectation in the sample covariance calculations, instead of the sample mean. Mainly due to these reasons, VEnKF has better performance than EnKF in our examples, when the ensemble size is small (see Section 4). Note that VEnKF especially applies to high dimensional problems, since all calculations are performed using the LBFGS covariance representations, without handling full covariance matrices.

It is worth noting that the covariances produced by LBFGS are full rank and VEnKF is therefore not a ‘reduced rank’ method. Thus, the new ensemble members generated in VEnKF perturb also the directions of small eigenvalues. This is not the case for many other covariance approximation/optimization methods that could be used instead of LBFGS. For example, Lanczos and conjugate gradient methods operate on a low-dimensional subspace.

4. Numerical Experiments. In this section, we demonstrate VEnKF with two synthetic examples, adopted from [2]. The first example is the well-known Lorenz 95 benchmark problem (a low-order nonlinear chaotic ODE system), that shares some characteristics with weather models. The second example is a linear system, where the dimension of the problem can be controlled by changing discretization.

For comparing methods, we use the root mean square error (rms), written as

\[
[rms]_k = \frac{1}{N} \sum_{i=1}^{N} \|x_{est}^i - x_{true}^i\|^2,
\]

where, at iteration \(k\), \(x_{est}^i\) is the filter estimate and \(x_{true}^i\) is the truth used in data generation.

4.1. Lorenz 95. In this example, we consider the well-known nonlinear and chaotic Lorenz’95 model, introduced in [25] and analyzed in [26]. The model shares many characteristics with realistic atmospheric models and it is often used as a low-order test case for weather forecasting schemes. We use a 40-dimensional version of the model, given as an ODE system as

\[
\frac{dx_i}{dt} = (x_{i+1} - x_{i-2})x_{i-1} - x_i + 8, \quad i = 1, 2, ..., 40.
\]

The state variables are periodic: \(x_{-1} = x_{39}, \ x_0 = x_{40} \) and \(x_{41} = x_1\). Out of the 40 model states, measurements are obtained from 24 states. We define the observation operator (following [2]) as \(K(x) = \mathbf{K}x\), where

\[
[K]_{rp} = \begin{cases} 
1, & (r, p) \in \{(3j + i, 5j + i + 2)\} \\
0, & \text{otherwise} 
\end{cases}
\]

where \(i = 1, 2, 3\) and \(j = 0, 1, ..., 7\). Thus, we observe the last three states in every set of five. To generate data, we add Gaussian noise to the model solution with zero mean and covariance \((0.15\sigma_{lim})^2\mathbf{I}\), where \(\sigma_{lim} = 3.641\) (standard deviation used in climatological simulations). In the filtering methods, we use \(C_{e} = (0.05\sigma_{lim})^2\mathbf{I}\) as the model error covariance and \(C_{rv} = (0.15\sigma_{lim})^2\mathbf{I}\) as the observation error covariance. As initial guesses in the filtering, we use \(x_{est}^0 = \mathbf{1}\) and \(C_{v0} = \mathbf{I}\). For more details about the example, refer to [2].

We run experiments with varying ensemble size \(N\) and varying number of LBFGS iterations. In figure 4.1, we compare the performance of EKF, EnKF and VEnKF with \(N = (10, 15, 20, 40)\) in terms of rms error. Since EnKF and VEnKF are stochastic methods, we show rms errors averaged over 10 repetitions. In VEnKF, the number of LBFGS iterations and the number of LBFGS vectors stored was the same as the ensemble size. From the results it is clear that VEnKF works better, when the ensemble size is small. When the ensemble size gets large, VEnKF and EnKF performances approach each other.

In figure 4.2, we compare the forecast skills given by different methods, using the same ensemble sizes as above. The forecast skill is here defined as the mean squared difference between the ‘truth’ and the forecast made with the model, scaled with \(\sigma_{true}\), see [2] for details. Again, VEnKF performs better, especially when \(N\) is small. For instance, VEnKF with \(N = 10\) performs equally well as EnKF with \(N = 20\). Even with larger \(N\), VEnKF is better on average.

To further demonstrate the behavior of VEnKF, we compare in figure 4.3 the rms errors (averaged over time) with varying ensemble sizes and varying number of LBFGS iterations used. For reference, we plot also the EnKF performance. One can see that 30 LBFGS iterations gives practically as good performance as 40 or 100 iterations, and EnKF starts to produce acceptable results when \(N \geq 30\).

4.2. Heat Equation. The purpose of this example, adopted from [2], is to demonstrate VEnKF behavior when the dimension is large. The example is linear, so we can directly compare to KF. However, as the dimension of the problem is increased, KF cannot be run anymore due to memory issues. Note that while the example does illustrate some computational aspects related to the methods, this system is well-behaved and we cannot conclude...
much about how the methods work in a high-dimensional chaotic case such as numerical weather prediction, for example.

The model describes heat propagation in a two-dimensional grid and it is written as a PDE as

$$\frac{\partial x}{\partial t} = -\frac{\partial^2 x}{\partial u^2} - \frac{\partial^2 x}{\partial v^2} + \alpha \exp \left( -\frac{(u - 2/9)^2 + (v - 2/9)^2}{\sigma^2} \right),$$

where $x$ is the temperature at coordinates $u$ and $v$ over the domain $\Omega = \{(u,v)|u,v \in [0,1]\}$. The last term in the equation is an external heat source, whose magnitude can be controlled with the parameter $\alpha \geq 0$.

We discretize the model using a uniform $S \times S$ grid. This leads to a linear forward model $\mathbf{x}_{k+1} = \mathbf{Mx}_k + \mathbf{f}$, where
VARIATIONAL ENSEMBLE KALMAN FILTER

M corresponds to the heat diffusion and f to the external forcing, see [2] for details. The dimension of the problem can be controlled by changing S. The observation operator K is defined as in [2]: the measured temperature is a weighted average of temperatures at neighboring points at $S^2/64$ evenly spaced locations.

Data is generated by adding normally distributed random noise to the model state and the corresponding response:

$$x_{k+1} = Mx_k + f + N(0, (0.5\sigma_{ev})^2 I)$$
$$y_{k+1} = Kx_{k+1} + N(0, (0.8\sigma_{obs})^2 I).$$

In data generation, we use $\alpha = 0.75$ and choose $\sigma_{ev}$ and $\sigma_{obs}$ so that the signal to noise ratios at the initial condition, defined by $||x_0||^2/S^2\sigma_{ev}^2$ and $||Kx_0||^2/m\sigma_{obs}^2$, are both 50. The initial condition for data generation is

$$x_{0_{ij}} = \exp\left(-\frac{(u_i - 1/2)^2}{\sigma_v^2} - \frac{(v_j - 1/2)^2}{\sigma_v^2}\right).$$

For the filtering we use a biased model, where the forcing term is dropped by setting $\alpha = 0$. The error covariances used for model and observations are $\sigma_{ev}^2 I$ and $\sigma_{obs}^2 I$, respectively. We start all filters from initial guess $x_0 = 0$. For ensemble filters, all members are initialized to the same value and for KF we set initial covariance estimate to $C_0 = 0$.

As our first test, we take $S = 2^j$ and choose $j = 5$, which is the largest integer so that KF can still be computed in a standard desktop computer. Thus, the dimension of the initial test was $d = S^2 = 1024$. In figure 4.4, we compare KF, VEnKF and EnKF, using ensemble sizes $N = (5, 10, 20, 50, 100)$ for VEnKF and EnKF. In VEnKF, we always take 20 LBFGS iterations and store 20 LBFGS vectors. The performance of VEnKF approaches KF as the ensemble size increases, but EnKF performs poorly: only with larger ensemble sizes we get acceptable results. We think that such a dramatic difference between VEnKF and EnKF is related to the handling of the measurement error in the method. In this case, we have simple linear dynamics and a rather good guess about the model error, and just a plain ‘3D-Var’ method, where the prior covariance is kept constant, performs rather well. This is the lower limit of the performance of VEnKF, upon which we can improve by adding ensemble members. In EnKF, model error can be added only by perturbing the (few) samples randomly, which can lead to large errors. In addition, in EnKF the state estimate is calculated as the sample mean, whereas VEnKF uses the MAP estimate: in this case this might produce the large errors in EnKF.

Next, we compared VEnKF to EnKF in a case, where the dimension is much higher: $(j = 7, d = S^2 = 16384)$, using the same ensemble sizes and same LBFGS settings. In this case, KF cannot be used anymore due to memory issues. As seen in Figure 4.5, the difference between EnKF and VEnKF is even more dramatic in this case: EnKF performance is poor, whereas VEnKF is able to improve the simple 3D-Var results.
5. Discussion. In the past decade, a wide literature about ensemble filtering has emerged. We start this section by reviewing some of it and discuss the existing approaches in light of our VEnKF method. Later, we discuss some specific topics related to VEnKF.

The standard EnKF is criticized in many papers about the additional sampling errors brought by randomly perturbing the observations. In so called square root ensemble filters (SRFs) this is not needed, see e.g. [13] and the review of SRF methods given in [31]. Similar approaches include the Ensemble Adjustment Kalman Filter (EAKF) and the Ensemble Transform Kalman Filter (ETKF), given in [1] and [5] respectively. In SRF methods, prior ensemble is deterministically transformed so that the posterior statistics match with the theory in the linear
case (Kalman filter equations). This is done basically by writing the Kalman formulas for matrix 'square roots' (symmetric decompositions) instead of covariance matrices in a manner that avoids forming full covariance matrices. This transformation is non-unique, leading to various SRF formulations, see [31] for a comparison. One difficulty in SRF methods, as pointed out in [31], is the handling of the model error. A simple way is to include it by perturbing the ensemble (as in EnKF), but this potentially leads to sampling errors, much in the same way as with perturbed observations in the standard EnKF. Dropping the model error altogether leads to underestimation of errors and possible divergence issues, and different 'covariance inflation' mechanisms need to be developed for this purpose.

In our VEnKF algorithm, both the model error and the observation error covariances are explicitly present in the minimized cost functions, and these problems do not exist. Moreover, the square root filters operate, as the standard EnKF, only in the subspace spanned by the ensemble vectors, whereas the VEnKF samples from the full state space.

Hybrid EnKF methods that incorporate features from both EnKF and variational methods (as VEnKF) have been found appealing in many studies. In [17], a hybrid approach is adopted, which combines elements of ensemble filtering and 3D-Var. However, the hybrid method uses perturbed observations and makes the perfect model assumption; model error covariance cannot be easily incorporated. The prior covariance is defined as a linear combination of the sample covariance and the static model error covariance used in 3D-Var, which is rather ad-hoc and introduces a tuning parameter (the weight given for the sample covariance). Similar approaches are introduced in [11] and [32]. In [43], the hybrid approach is extended so that it can be used with the 4D-Var method, but the same problems remain.

Probably the closest method to our VEnKF approach is the Maximum Likelihood Ensemble Filter (MLEF) proposed in [34]. In MLEF, an iterative optimization method is used to optimize a 3D-Var type of cost function, and the found optimum is used as the state estimate instead of the sample mean. However, the model error term is neglected and the method operates only in the ensemble subspace.

One criticism that is faced by all Kalman filter based methods are the linear (Gaussian) approximations: in all of the methods discussed so far in this paper, a Gaussian form for the prior is used. The ensemble methods are a bit 'more nonlinear' than EKF in the sense that the covariance information is propagated using the nonlinear model instead of the linearized model. Purely nonlinear filtering methods exist as well, see e.g. [8] for a recent introduction to Bayesian particle filters. Their benefit over the linear methods can be easily shown in small cases, but they become infeasible in large systems. Some nonlinearity in the prior formulation can be induced by representing the prior e.g. as a mixture of Gaussians or a kernel density estimate fitted to the ensemble, see e.g. [4, 1] for some discussion. We note that different, nonlinear prior and likelihood formulations can be rather easily incorporated into the cost functions used in the proposed VEnKF method. The only complication is that the cost function is no longer quadratic and one has use the full LBFGS algorithm instead of the quadratic version used in this paper and worry about e.g. the Wolfe conditions in the optimization. Extending VEnKF to this type of nonlinear filtering is a promising topic for further research.

In the VEnKF algorithm, we sample new ensemble members at each iteration. Traditionally, sampling from a multivariate Gaussian distribution is done by performing a symmetric decomposition on the covariance matrix, $C = LL^T$, for example using the Cholesky decomposition, and then producing random vectors as $Lz$, where $z \sim N(0, I)$. The work in [18] comes close to our approach as it employs the BFGS covariance approximation for the proposal distribution in the Metropolis MCMC algorithm, but using again the Cholesky decomposition. In high dimensions, performing these decompositions is infeasible, since we cannot even store the full covariance matrix. However, as shown in this paper, random sampling can be done directly using the LBFGS vectors. This is potentially useful in other contexts as well, where high dimensional random sampling is needed, for example in the field of inverse problems.

Although it was mentioned that EnKF has no issue with storing full covariance matrices, memory problems may still arise in computing the Kalman gain if there are very many observations. In VEnKF, there is no need for computing the Kalman gain.

Although we have shown that VEnKF performs well, the approach has its downsides. In particular, the LBFGS optimization is sensitive to certain 'parameters' and therefore VEnKF requires some case specific tuning. One tuning parameter is the preconditioner for the inverse Hessian used in the LBFGS optimization. In this paper, we use a heuristic given in [29], that eliminates this tuning parameter (see Appendix A for details). Other tuning parameters remain, such as the choice of the initial guess, the number of LBFGS vectors that we store and the number of LBFGS iterations that we take. At present, we have not found any general ways to define these parameters, other than 'trial and error'.

Naturally, the performance of VEnKF depends on the accuracy of the LBFGS covariance approximations. In [2], the quality of LBFGS covariances was found to be good in low-dimensional numerical examples. However, LBFGS
performance in approximating covariance in high dimensions and realistic data assimilation problems remains a topic of future research.

6. Conclusions. In this paper, we propose a hybrid method called Variational Ensemble Kalman Filter (VEnKF) for high-dimensional data assimilation that combines elements from ensemble filtering and variational methods. VEnKF is based on the Variational Kalman Filter (VKF) method of [2], where the memory issues related to EKF are solved by low-storage approximations of state covariances obtained using the LBFGS optimization method. The proposed approach can solve some problems often encountered with ensemble methods, such as sampling errors due to random perturbation of states and observations and ensemble in-breeding. All of the calculations in VEnKF, noting especially the generation of new ensembles, can be made using the LBFGS covariance representation, without handling full covariance matrices. Since VEnKF is an ensemble method, it does not need tangent linear and adjoint codes. We show with synthetic examples that the method can perform better than the standard ensemble Kalman filter. Testing the applicability of the method to real data assimilation problems remains a topic of future research.

Acknowledgements. The work was supported by the Centre of Excellence in Inverse Problems of the Academy of Finland.

Appendix A. In this appendix, we give details of some computational issues in VEnKF. First, we recall how the LBFGS optimization algorithm for quadratic minimization problems works. Then, we shown how random samples can be produced from the LBFGS covariance representation.

The LBFGS algorithm for minimizing a quadratic function \( q(u) = \frac{1}{2}u^TAu \), given an initial guess \( u_0 \), reads as

**LBFGS algorithm for quadratic problems**

1. Choose inverse Hessian approximation \( H_0 \).
2. Compute gradient \( g_k = \nabla q(u_k) = Au_k \).
3. Compute search direction \( p_k = H_k g_k \), where \( H_k \) is the LBFGS approximation of the inverse Hessian (details below).
4. Compute step size \( \alpha_k = (g_k^T p_k)/(p_k^T A p_k) \).
5. Set \( u_{k+1} = u_k - \alpha_k p_k \) and \( k \to k+1 \) and go to step 1.

The LBFGS algorithm uses the BFGS formula for approximating \( H_k \), which is recursively defined as

\[
H_{k+1} = V_k^T H_k V_k + \rho_k s_k s_k^T,
\]

where

\[
\begin{align*}
\rho_k &= 1/(y_k^T s_k) \\
V_k &= I - \rho_k y_k s_k^T \\
s_k &= u_{k+1} - u_k \\
y_k &= g_{k+1} - g_k
\end{align*}
\]

but in LBFGS only a certain number \( n \) of the vectors \( s_k \) and \( y_k \) are used in the above formula. Writing out the recursive formula and storing only \( n \) most recent vectors leads to the following formula for the LBFGS inverse Hessian approximation:

\[
H_k = (V_{k-1}^T \ldots V_{k-n}^T) H_k^0 (V_{k-n} \ldots V_{k-1}) + \rho_k s_{k-n} s_{k-n}^T + \rho_{k-n+1} (V_{k-n+1}^T \ldots V_{k-1}^T) s_{k-n+1} s_{k-n+1}^T + \ldots + \rho_{k-1} s_{k-1} s_{k-1}^T.
\]

Note that in the LBFGS formulation, the initial inverse Hessian approximation \( H_0 \) can vary from iteration to another. One can choose e.g. a fixed diagonal covariance \( H_0^0 = \gamma I \) but then the issue of tuning \( \gamma \) emerges. We use a heuristic from [29], in which \( H_0 = \gamma_0 I \) and \( \gamma_k = (s_{k-1}^T y_{k-1})/(y_{k-1}^T y_{k-1}) \), which attempts to estimate the size of the covariance along the last search direction, see [29].
In our applications, we never want to calculate and store the full inverse Hessian, but to keep it in the above 'vector form'. There exists an efficient iterative algorithm for computing matrix-vector products with the inverse Hessian, needed e.g. when calculating the search direction in the L-BFGS algorithm, see e.g. [29] for details.

Assuming that the initial inverse Hessian can be decomposed into $H^0_k = L_0 L_0^T$, the above L-BFGS inverse Hessian formula can be written in the form

$$H_k = B_0 B_0^T + \sum_{i=1}^{n} b_i b_i^T,$$

where

$$B_0 = (V^T_{k-1} \ldots V^T_{k-n}) L_0, \quad b_1 = \sqrt{\rho_{k-1}} s_{k-1}, \quad b_i = \sqrt{\rho_k (V^T_{k-i+1} \ldots V^T_{k-1}) s_{k-i}}, \quad i = 2, \ldots, n.$$  

Note that the square roots $\sqrt{\rho_i}$ can always be calculated, since in the L-BFGS algorithm we choose the step length so that $\rho_i \geq 0$ for all $i$, see [29] for details. Thus, we can sample zero mean random variables from the covariance $H_k$ by calculating

$$r = B_0 z + \sum_{i=1}^{n} \omega_i b_i,$$

where $z \sim \mathcal{N}(0, 1)$ and $\omega_i \sim \mathcal{N}(0, 1)$. It is simple to verify that $\text{Cov}(r) = H_k$. The needed products can be calculated efficiently without storing full matrices of size $d \times d$. For the first product $q = B_0 z$, we can use the iteration

- Set $q \leftarrow L_0 z$.
- For $i = 1 \ldots n$.
  1. $\rho_i = 1/(y^T_i s_i)$.
  2. Set $q \leftarrow q - (\rho_i y_i^T q) s_i$.
- Return $q$.

For the other products $\omega_i b_i$, we can first calculate the vectors $b_i$ explicitly using a similar loop as above and then take sums of the vectors, weighted by random numbers $\omega_i$. 
REFERENCES

405. TANSKANEN, ANNA. Analysis of electricity distribution network operation business models and capitalization of control room functions with DMS. 2010. Diss.


408. KÄMÄRI, VESA. Kumppanuusohjelman strateginen johtaminen – Monitapaututkimus puolustushallinnossa. 2010. Diss.


412. SADOVNIKOV, ALBERT. Computational evaluation of print unevenness according to human vision. 2010. Diss.


415. ZAKHVALINSKII, VASILII. Magnetic and transport properties of LaMnO_{3+\delta}, La_{1-x}Ca_{x}MnO_{3}, La_{1-x}Ca_{x}Mn_{1-y}Fe_{y}O_{3}, and La_{1-x}Sr_{x}Mn_{1-y}Fe_{y}O_{3}. 2010. Diss.


420. CHIPOFYA, VICTOR. Training system for conceptual design and evaluation for wastewater treatment. 2010. Diss.


425. VIRTA, MAARIT. Knowledge sharing between generations in an organization – Retention of the old or building the new 2011. Diss.
428. MARKELOV, DENIS. Dynamical and structural properties of dendrimer macromolecules. 2011. Diss.
446. HYNYLENEN, KATJA. Broadband excitation in the system identification of active magnetic bearing rotor systems. 2011. Diss.