FINNISH METEOROLOGICAL INSTITUTE
CONTRIBUTIONS

NO. 69

ADAPTIVE MCMC METHODS WITH APPLICATIONS IN ENVIRONMENTAL AND GEOPHYSICAL MODELS

MARKO LAINE

DEPARTMENT OF MATHEMATICS AND PHYSICS
LAPPEENRANTA UNIVERSITY OF TECHNOLOGY
LAPPEENRANTA, FINLAND

ACADEMIC DISSERTATION in applied mathematics

Thesis for the degree of Doctor of Philosophy to be presented with due permission for public examination and criticism in Auditorium 1383 at Lappeenranta University of Technology, Lappeenranta, Finland at 14th of March, 2008, at noon.

Finnish Meteorological Institute
Helsinki, 2008

ISBN 978-951-697-661-0 (paperback)
ISBN 978-951-697-662-7 (PDF)
ISSN 0782-6117

Yliopistopaino
Helsinki, 2008

FINNISH METEOROLOGICAL INSTITUTE

Published by Finnish Meteorological Institute
(Erik Palménin aukio 1), P.O. Box 503
FIN-00101 Helsinki, Finland

Series title, number and report code of publication
Contributions 69, FMI-CONT-69

Date
March 2008

Authors

Marko Laine

Name of project

Commissioned by

Title

Adaptive MCMC methods with applications in environmental and geophysical models

Abstract

This work presents new, efficient Markov chain Monte Carlo (MCMC) simulation methods for statistical analysis in various modelling applications. When using MCMC methods, the model is simulated repeatedly to explore the probability distribution describing the uncertainties in model parameters and predictions. In adaptive MCMC methods based on the Metropolis-Hastings algorithm, the proposal distribution needed by the algorithm learns from the target distribution as the simulation proceeds. Adaptive MCMC methods have been subject of intensive research lately, as they open a way for essentially easier use of the methodology. The lack of user-friendly computer programs has been a main obstacle for wider acceptance of the methods. This work provides two new adaptive MCMC methods: DRAM and AARJ. The DRAM method has been built especially to work in high dimensional and non-linear problems. The AARJ method is an extension to DRAM for model selection problems, where the mathematical formulation of the model is uncertain and we want simultaneously to fit several different models to the same observations.

The methods were developed while keeping in mind the needs of modelling applications typical in environmental sciences. The development work has been pursued while working with several application projects. The applications presented in this work are: a winter time oxygen concentration model for Lake Tuusulanjärvi and adaptive control of the aerator; a nutrition model for Lake Pyhäjärvi and lake management planning; validation of the algorithms of the GOMOS ozone remote sensing instrument on board the Envisat satellite of European Space Agency and the study of the effects of aerosol model selection on the GOMOS algorithm.

Publishing unit
Earth Observation

Classification (UDK)
519.2, 519.6, 504.064.2, 504,45

Keywords
Markov chain Monte Carlo, adaptive MCMC, Bayesian statistical inference, statistical inversion, model selection, environmental modelling, geophysical modelling, atmospheric remote sensing, Envisat, GOMOS

ISSN and series title
0782-6117 Finnish Meteorological Institute Contributions

ISBN
978-951-697-661-0 (paperback), 978-951-697-662-7 (PDF)

Language
English

Sold by
Finnish Meteorological Institute / Library
P.O.Box 503, FIN-00101 Helsinki
Finland

Pages
146

Price

Note

ILMATIETEEN LAITOS

Julkaisun sarja, numero ja raporttikoodi
Contributions 69, FMI-CONT-69

Julkaisija Ilmatieteen laitos, (Erik Palménin aukio 1)
PL 503, 00101 Helsinki

Julkaisuaika
Maaliskuu 2008

Tekijä(t)

Marko Laine

Projektin nimi

Toimeksiantaja

Nimeke
Adaptiivisia MCMC menetelmiä sovellettuna ympäristötieteellisiin ja geofysikaalisiin malleihin

Tiivistelmä

Tässä työssä esitetään uusia keinoja tehostaa tilastollisessa mallintamisessa käytettäviä Markovin ketju Monte Carlo (MCMC) simulointimenetelmiä. MCMC-menetelmiä käytettäessä mallia simuloidaan toistuvasti, jotta löydettäisiin mallin parametrien ja ennusteiden epävarmuutta kuvaavat todennäköisyysjakaumat. Metropolis-Hastings-algoritmiin perustuvissa adaptiivisissa MCMC-menetelmissä algoritmin tarvitsema ehdotusjakauma mukautuu kohdejakaumaan simuloinnin edetessä. Adaptiiviset MCMC-menetelmät ovat olleet viimeaikoina vilkkaan tutkimuksen kohteena, sillä niiden avulla voidaan menetelmien käyttöä oleellisesti helpottaa. Helppokäyttöisten ohjelmistojen puute onkin ollut suurin este MCMC-menetelmien laajemmalle käytölle. Tässä työssä esitetään kaksi uutta adaptiivista MCMC-menetelmää: DRAM ja AARJ. DRAM-menetelmä on kehitetty toimimaan tehokkaasti erityisesti suuriulotteisissa ja epälineaarisissa ongelmissa. AARJ on puolestaan DRAM-menetelmän laajennus mallinvalintaongelmiin, jossa myös mallin matemaattisessa muodossa on epävarmuutta ja samoihin havaintoihin halutaan yhtä aikaa sovittaa useita eri malleja.

Menetelmiä kehitettäessä on tärkeänä periaatteena on ollut niiden käyttökelpoisuus luonnontieteissä esiintyvissä mallinnussovelluksissa. Menetelmiä onkin kehitetty yhdessä niistä hyötyvien sovellusten kanssa. Erityisesti mallien avulla laskettaviin ennusteisiin liittyvien epävarmuuksien luotettava arvioiminen on tullut mahdolliseksi tavalla, joka ei aikaisemmilla menetelmillä ole onnistunut. Työssä esiintyviä sovelluksia ovat Tuusulanjärven talviajan hapenkulutuksen mallintaminen ja hapettimen adaptiivinen säätö, Säkylän Pyhäjärven ravinnekuormitusmallien estimoiminen ja järvien hoitotoimenpiteiden mitoitus sekä Euroopan avaruusjärjestön Envisat-satelliitin GOMOS-otsonimittalaitteen algoritmien validointi ja GOMOS-algoritmissa käytettävän aerosolimallin valinnan vaikutuksen arvioiminen.

Julkaisijayksikkö
Uudet havaintomenetelmät

Luokitus (UDK)
519.2, 519.6, 504.064.2, 504.45

Asiasanat
Markovin ketju Monte Carlo -menetelmät, adaptiivinen MCMC, bayesläinen tilastopäättely, tilastollinen inversio, mallin valinta, ympäristötieteen mallit, geofysikaalinen mallintaminen, ilmakehän kaukokartoitus, Envisat, GOMOS.

ISSN ja avainnimike
0782-6117 Finnish Meteorological Institute Contributions

ISBN
978-951-697-661-0 (paperback), 978-951-697-662-7 (PDF)

Kieli
Englanti

Myynti
Ilmatieteen laitos / Kirjasto
PL 503, 00101 Helsinki

Sivumäärä 146

Hinta

Lisätietoja

To Tarja

Acknowledgements

This work would not have been finished, or even started, without the support and guidance from my advisor Prof. Heikki Haario at Lappeenranta University of Technology. His ability to provide an endless supply of projects on many diverse areas of science and technology has given me the exceptional opportunity to work with different universities, research and educational institutes and with industrial partners. Thank you, Heikki.

I started the preparation of this work while at Department of Mathematics and Statistics, University of Helsinki. I am in debt to all my colleagues there for the inspiring environment at the department and the ability to work with many world class mathematicians. I especially thank Prof. Esko Valkeila (now at Helsinki University of Technology) who initiated my university career by arranging a job at the University Computing Centre. Esko was also the supervisor of my Licentiate Thesis and was the one who talked me into doing scientific research.

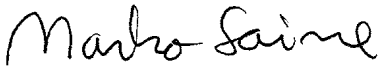
From the year 2006 onwards and during the final preparation of this thesis I have been working at Finnish Meteorological Institute (FMI). I thank all my colleagues for the stimulating atmosphere here. Prof. Jarkko Koskinen and Prof. Tuija Pulkkinen, the former and the present Heads of Earth Observation Unit, are acknowleged for the excellent working conditions at FMI. Warmest thanks go to Dr. Johanna Tamminen, the leader of the Atmospheric Remote Sensing research group. She has given me much needed and appreciated support and encouragement for both my research and my work career.

Thanks to all my co-authors, some of them already mentioned above. Prof. Antonietta Mira, Prof. Eero Saksman and Prof. Markku Lehtinen have showed me the meaning of scientific collaboration at its best. Dr. Olli Malve from Finnish Environment Institute (SYKE) has become a personal friend of mine during the preparation of the two environmental articles in this work, and in addition to studying the ecosystems of the lakes we have had the pleasure of canoeing, skiing and skating on them.

I wish to thank my reviewers Prof. Elja Arjas and Prof. Jukka Corander for their criticism and comments on this work. Likewise, I thank Mr. Malcom Hicks for the proof reading of the introductory part of this thesis.

This work has been financially supported by Finnish Academy's MaDaMe project and by Masi project of Finnish Funding Agency for Technology and Innovation (Tekes).

Thanks to all my friends who have supported me during the years, and to my daughters Maisa and Mimosa for making our family. This work is dedicated to my wife Tarja, who passed away November 2007. I would have wanted so much for her to see me finishing this project, at last.



Marko Laine
Tuusula, Finland
February 18, 2008

Contents

1 Summary of the original articles

This thesis consist of an introductory part, four original refereed articles in scientific journals, and a description of a software for doing the MCMC analysis described in this work.

The author's contributions to the articles in this dissertation are summarized below, where the main scientific results are stated and their relevance and importance for the research field in general are discussed. The software consists of a Matlab toolbox as described in the Appendix.

Paper 1: Lake Tuusulanjärvi

[1] Olli Malve, **Marko Laine**, and Heikki Haario: Estimation of winter respiration rates and prediction of oxygen regime in a lake using Bayesian inference, *Ecological Modelling*, 182(2), pages 183–197, 2005. (doi:10.1016/j.ecolmodel.2004.07.020)

The computational problem of estimating the posterior distributions of a high dimensional parameter vector is solved by means of the Adaptive Metropolis algorithm. A full Bayesian analysis of the unknowns is performed, including errors in the control variables. The trends in the lake respiration rate are studied in a non-parametric way and using predictive posterior inference. A sequential Bayesian analysis is performed, showing how posterior knowledge accumulates as new information becomes available. The work provides tools for adaptive lake management, e.g. the real time control of an artificial aerator.

Paper 2: Lake Pyhäjärvi

[2] Olli Malve, **Marko Laine**, Heikki Haario, Teija Kirkkala, Jouko Sarvala: Bayesian modelling of algae mass occurrences - using adaptive MCMC methods with a lake water quality model, *Environmental Modelling & Software*, 22(7), pages 966–977, 2006. (doi:10.1016/j.envsoft.2006.06.016)

A four-dimensional state space of phytoplankton groups is modelled as a dynamical model with a total of 64 unknowns. MCMC sampling is performed by the DRAM method developed during this work. The data are transformed to achieve normality of the residuals, variance homogeneity and truthful model predictions. Bayesian predictive inference with regard to the effect of the phytoplankton bloom on different scenarios is carried out based on simulated profiles of the control variables.

The study of nutrient loadings is a key area of research envisaged by the EU Water Framework Directive. The model described and developed during the preparation of the article is being actively studied and further applied to other lakes by the Finnish Environment Institute (SYKE).

Paper 3: Atmospheric ozone profile retrieval from a satellite instrument

[3] Heikki Haario, **Marko Laine**, Markku Lehtinen, Eero Saksman, and Johanna Tamminen: MCMC methods for high dimensional inversion in remote sensing, *Journal of the Royal Statistical Society, Series B*, 66(3), pages 591–607, 2004. (doi:10.1111/j.1467-9868.2004.02053.x)

The inversion algorithm of the GOMOS instrument on board the ENVISAT satellite is studied. An alternative to the operational GOMOS algorithm is constructed via parallel MCMC chains, which better takes into account the nonlinear correlations between unknowns and provides means for modelling the error structure more realistically. Smoothness properties of the solution are given as prior information.

The Finnish Meteorological Institute is an active partner in the GOMOS data processing and algorithm development project. GOMOS data are actively used in the modelling of environmental changes. The realistic uncertainty estimates provided by this work facilitate the assimilation of GOMOS data into climate models.

Paper 4: DRAM

[4] Heikki Haario, **Marko Laine**, Antonietta Mira, and Eero Saksman: DRAM: Efficient adaptive MCMC, *Statistics and Computing*, 16(4), pages 339–354, 2006. (doi:10.1007/s11222-006-9438-0)

A novel combination of simulation tools is suggested for MCMC sampling. The method of Delayed Rejection is combined with Adaptive Metropolis. Proof is given of the ergodicity of the new sampler, based on previous work on Adaptive Metropolis. Extensive simulation test runs for targets with non-linear correlations between the components were performed to demonstrate how the methods work in practice. The model of Paper 2 is employed as an example of the use of the DRAM method in a complex modelling application.

DRAM makes it possible to build a general automatic Bayesian inference software tool for a wide variety of modelling problems.

Paper 5: Adaptive RJMCMC

[5] **Marko Laine** and Johanna Tamminen: Aerosol model selection and uncertainty modelling by adaptive MCMC technique, *submitted to Atmospheric Chemistry and Physics*, 2008.

This work shows how the DRAM adaptation can be naturally combined with the automatic reversible jump MCMC of Green [2003]. The new method is called AARJ and the algorithm is applied to the aerosol model selection problem in the GOMOS ozone instrument on board the ENVISAT satellite. The aerosol model is one of the key features in any atmospheric remote sensing instrument. The model averaging approach will take into account the uncertaities due to model selection and make the error estimates more realistic.

Summary of the author's contributions

The author has, in general, been responsible for the statistical analysis and the computer runs in all of the articles. This includes computational implementation, fitting and validation, interpretation of the results, iterative model building, diagnostics and model refinement. The coding for the toolbox used for MCMC simulation and for testing various adaptive strategies was performed by the author. In Papers 1 and 2 the author was responsible for the statistical analysis and predictive simulation runs. He invented various statistical graphs for specific purposes, most notably the diagnostic predictive plot displaying the components of the model uncertainty. The author's original contribution to Paper 3 was the use of parallel MCMC chains to reduce the high dimension of the GOMOS inversion problem. In Paper 4 (DRAM) he performed the test runs for the various strategies and the final implementation for combining DR and AM. He was the corresponding author for Paper 5, in which he suggested the idea of combining automatic RJMCMC with DRAM and AM.

2 Introduction

The main idea of this dissertation is to demonstrate how adaptive MCMC methods, especially the DRAM and AARJ methods developed in this work, can increase the efficiency of the Metropolis-Hastings algorithm and help the utilization of Bayesian modelling for a wider class of models than has been possible with previous MCMC tools.

Examples of the fields where this kind of computational methodology works well include models for dynamical systems in ecology, geophysics and chemical kinetics, to mention only those that are presented in this work. The scope of possible applications is, of course, much wider. For example, the method will work well for classical nonlinear regression models such as the ones treated by Bard [1974] and Seber and Wild [1989].

This work belongs to the domain of applied mathematics and statistics. A new computational methodology is developed to help in the statistical analysis of nonlinear mathematical models. The approach facilitates the statistical analysis of models on a larger scale than with the previous methods, and allows for proper investigation of the uncertainty in the model and in its predictions. The approach is successfully applied to problems in the environmental sciences. The computational algorithms are also discussed. A feature common to all the modelling activities presented here is the use of predictive analysis. The model parameters are usually of indirect interest, and we are mostly interested in the model predictions. The uncertainty of the predictions is calculated using posterior predictive distributions.

In many situations there are not enough data to properly identify the model parameters. The posterior distributions, describing the uncertainties of the estimated values, are too wide and correlated for practical judgements about the actual values of the unknowns, given the available data and the structure of the model. The high dimension of the model can also make the posterior estimation problem hard to solve computationally. Adaptive MCMC tools help to recognize and to some extent circumvent these problems. Moreover, the application articles show how meaningful and easily interpretable inferences can arrived at using the results of MCMC runs. Models should be judged by their predictive power. A model can produce accurate predictions even if its parameters are not well identified. In other words, if the MCMC sampler is working correctly, in spite of the correlations and large posteriors, it is possible make useful predictive analysis on model outcomes. A special predictive plot is introduced as a tool for model diagnostics. It divides the uncertainty in the model predictions into two parts, the part that is due to uncertainty in the model parameters and the part that is due to uncertainty in the observations. This helps to diagnose possible problems, such as the lack of fit or the distributional assumptions of the observational error.

This work started in 2000 as a part of the Academy of Finland's MaDaMe project, called "Development of Bayesian methods with applications in geophysical and environmental research", aimed at applying and further developing the recent adaptive MCMC methods (especially the Adaptive Metropolis, AM [Haario et al., 2001]). Encouraging experience with the use of AM in a satellite instrument modelling and inversion project [Tamminen, 2004] was already available. The statistical inversion problem of the GOMOS instrument data is also one of the main topics of Papers 3 and 5 of this dissertation. The adaptive methods were first applied to an oxygen depletion model for Lake Tuusulanjärvi, with a total of 60 unknowns to be estimated (Paper 1). The next idea for improving the MCMC methodology was to combine the AM adaptation with the Delayed Rejection (DR) method of Mira [2001]. The DRAM method introduced and described in detail in Paper 4 was successfully applied to a lake algal model in Paper 2, and has later proved to be useful in several other applications. An adaptive version of the automatic reversible jump MCMC of Green [2003] was employed for the aerosol model selection problem in the GOMOS inversion (Paper 5).

The structure of the introductory part of the thesis is the following. First, in Section 3, we recall

the basic statistical terminology for the modelling problems of this work. Secondly, in Section 4, we describe the most widely used MCMC algorithm, the Metropolis-Hastings (MH) algorithm. Then the adaptive improvements made to the MH are explained. The main result of this thesis is the introduction of the DRAM method, which improves the efficiency of the basic Metropolis-Hastings algorithm. The implementation details for the adaptive MCMC are discussed in Section 5. The DRAM method can also be used in model determination problems. This is demonstrated in Section 6 by introducing an adaptive version of the reversible jump MCMC algorithm. Finally, in Section 7, the applications in the accompanying articles are described from the perspective of the MCMC analysis conducted.

A set of Fortran 90 subroutines for adaptive MCMC calculations were developed by the author. These were included in the Modest modelling software package [Haario, 1995]. Later, the author developed a general Matlab toolbox for adaptive MCMC calculations [Laine, 2007], which was used together with the Modest software for the calculations in all of the applications described here. The software is documented in the Appendix to this thesis.

3 Statistical inference for models

An important task in science is to understand and predict nature by means of modelling. The phenomena under investigation could be as different as the algal dynamics of a lake or measurements produced by a satellite instrument, as in the applications described in this work, although the mission is universal for many scientific activities.

Statistical analysis studies the uncertainties in scientific inference by means of probabilistic reasoning. For a full statistical treatment of uncertainties, we assume that all the unknown quantities can be described by statistical distributions, whether they are model parameters, unknown states of the system in question, model predictions, or prior information on the structure of the solutions.

We typically have direct or indirect *observations* about the *state* of the system. A *model* is a mathematical description of the process that generates the states and the observations. The model can depend on a set of *model parameters* and it can be driven externally by *control variables* such as temperature or pressure. We also have a separate *error model*, that accounts for the unsystematic variation in the observations not covered by the systematic part of the model.

If we are interested in the model parameters, the inference is called *parameter estimation*. A related problem in geophysics and other similar applied fields is called the *statistical inverse problem*. In inverse problems the target of the estimation is, usually, a discretized version of an unknown function. The unknowns describe the indirectly observed state of the system, and are therefore not in a strict sense model parameters, but statistically we are dealing with the same problem of estimating unknown quantities with the help of a model, data, and prior specifications about the unknowns.

In simple terms, we can write our model as

$$y = f(x; \theta) + \epsilon. \tag{1}$$

Here y stands for the observations whose expected mean behaviour is described by the model f, which depends on certain external control variables x and on some unknown quantities θ. The term ϵ stand for the unsystematic observational errors. The description of the problem would then be completed by giving the observations for the y and x variables, the statistical distribution of the error ϵ and the prior distribution for the unknown θ.

To be more specific and to reveal the types of uncertainties that we will be dealing with in the applications, we present the following more detailed descriptions of the modelling problem. We have n_{set}

sets of data for a system with an unknown state s. For each data set we observe n_{obs} observations of n_{y} separate "y" variables. We could observe some parts of the state s of the system directly or in some indirect way covered by the model. A hierarchical description of the uncertainties considered in the modelling procedure could then be written as

$$y_{ijk} = f(s_k; x_k; \theta_k) + \epsilon_{ijk}, \qquad \text{model for the observations} \tag{2}$$

$$\theta_k \sim p(\theta_0, \tau_\theta^2), \qquad \text{prior for the model parameters} \tag{3}$$

$$x_k \sim p(x_0, \tau_x^2), \qquad \text{error in the control variables} \tag{4}$$

$$s_k \sim p(s_0, \tau_s^2), \qquad \text{prior model for the state} \tag{5}$$

$$\Delta s \sim p(s_\Delta, \tau_\Delta^2), \qquad \text{structural smoothness prior for the state} \tag{6}$$

$$\epsilon_{ijk} \sim p(\sigma_{jk}^2), \qquad \text{error model} \tag{7}$$

$$\sigma_{jk}^2 \sim p(\sigma_{0k}^2, \tau_{\sigma_k}^2), \qquad \text{error model parameters} \tag{8}$$

where the indices go as $k = 1, \ldots, n_{\text{set}}$, $j = 1, \ldots, n_{\text{y}}$, and $i = 1, \ldots, n_{\text{obs}}(k)$. We use a general notation for the prior information. The term $p(\mu_0, \tau_0^2)$ means simply some unspecified distributions with the given "location" and "scale" (or covariance) parameters μ_0 and τ_0^2. In the applications described in this work, however, the priors are typically Gaussian, possibly with extra positivity constraints. Although Bayesian MCMC analysis allows for using any distributions, the Gaussian distribution still predominates in many studies. This is sometimes just for computational convenience, but in many cases there are good theoretical and practical reasons for the error terms to behave, at least approximately, like Gaussian random variables.

There is some notational ambiguity in describing a general Bayesian MCMC modelling framework. In the statistical literature one usually understands the y and x variables as the response variable and the set of control variables, with Greek letters reserved for the unknown model parameters. In the inverse problems literature, x usually stands for the unknown state to be estimated. Moreover, in literature on MCMC theory, the notation x stands for the unknown variable we are trying to estimate. Here we follow the notation common to the statistical literature.

3.1 The Bayes formula

We will here review the terminology used for the Bayesian analysis of uncertainties in modelling. To simplify the notations in this section, let θ stand for all the unknowns in our model. The inference concerning θ is performed using the conditional distribution of the parameter, given the model and the observations, $p(\theta|y)$. This probability distribution is called the *posterior distribution*. By the rules of conditional probabilities, we can invert the posterior and arrive at *the Bayes formula*:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}. \tag{9}$$

The right hand side of the Bayes formula has the following ingredients. The *likelihood function*, $p(y|\theta)$, is the density function of the observations y given θ. When the observed data y are held fixed, $p(y|\theta)$ is considered as a function of the parameters θ. If for two possible parameter values θ and θ' we have $p(y|\theta') > p(y|\theta)$, then the observation y is more likely to happen under θ' than under θ, hence the term "likelihood". In the case of model (2) the likelihood is calculated by combining the model function with the probability density functions of the error term ϵ, see section 5.1.

The unconditional distribution of the unknowns, $p(\theta)$, is called the *prior distribution*. This describes the knowledge about the unknowns that existed before, or exists independent of, the current observations. We discuss the role of the prior more closely in section 3.3.

Again, when the data y are fixed, the term in the denominator, $p(y)$, is a single number, which can be seen as a constant that normalizes the likelihood-times-prior part into a proper probability distribution. When comparing different models for the same data, this unconditional probability of observations can also be seen as a (prior) predictive probability of the observations given the model. For this reason it is sometimes called the *evidence*. By the rule of total probability, we can calculate its value by integrating the product $p(y|\theta)p(\theta)$ over all values of θ

$$p(y) = \int p(y|\theta)p(\theta)\,d\theta. \tag{10}$$

This integral makes the actual computation of the posterior distribution a challenging task in all but some special cases. For most cases we need special numerical algorithms, e.g. the MCMC algorithm described in section 4.

3.2 Predictive distributions

An important element in Bayesian inference is the prediction of future observations y^* given the current ones y. Given the observations y and the model $p(y|\theta)$, the predictions are naturally based on the posterior distributions of θ. If we assume that y and y^* are conditionally independent given the value of θ, we can write

$$p(y^*|y) = \int p(y^*, \theta|y)\,d\theta = \int p(y^*|\theta)p(\theta|y)\,d\theta. \tag{11}$$

Here the predictive distribution of y^* is given as the expectations of $p(y^*|\theta)$ with respect to the posterior distribution $p(\theta|y)$. In this work we use the simulated MCMC chain as a sample from $p(\theta|y)$. If we sample θ from the chain and y^* from $p(y^*|\theta)$, we obtain samples of the predictive distribution $p(y^*|y)$. The predictive distribution is the basis for the validation of the model against reality. This is not possible using only the posterior distribution of the model parameters θ, as the parameters are not directly observable.

3.3 Prior information

Next, we consider the role of prior information and the problem of assigning the prior. In a straightforward classical interpretation of Bayesian inference the prior signifies the modeller's honest opinion about the unknown. In most modelling activities dealing with scientific inference, however, the idea of subjective probability seems restrictive. A common concern has to do with the problem of formulating the prior as "objectively" as possible. The solutions proposed include such concepts as non informative [Box and Tiao, 1973], hierarchical [Gelman et al., 1995], reference [Bernardo and Smith, 2000], vague, or diffuse [Tarantola, 2005] priors.

We will encounter three uses of the prior distribution in the course of this work: uninformative priors, describing a lack of, or unwillingness to provide, prior information on the unknowns, priors that provide structural information about the solution, and finally priors that are based on empirical evidence from previous experiments.

A modeller would usually be happy if the data were to produce informative posteriors "without a prior". Even if we had information about the unknowns from previous similar experiments, we would like the data to verify our conclusions. This usually means using very wide priors, or even setting $p(\theta_i) \equiv 1$ in the Bayes formula (9) for some or all of the components of the unknown. Care must be taken that the posterior becomes a proper distribution.

Sometimes the prior gives the structure of the solution. In inverse problems where the state of the system is estimated, for example, we can have information on some of the properties of the system. In these cases the solution might not exist, or it could very unstable, unless some prior information is used [Kaipio and Somersalo, 2004]. If this regularization of the problem is done via a prior distribution, we can have an intuitive interpretation for the restrictions and a better view of the effects of the premises on the conclusions. This type of prior is used in the GOMOS inversion application of Paper 3, where the inverted gas profiles are given smoothness properties as prior information.

When the data are sparse a great deal of effort should be put into selecting the prior distribution. When possible, the prior should be based on real information. In large-scale problems with a large number of unknowns it is usually not possible to formulate detailed priors. If we assume that the unknowns are a priori independent, we can independently assign one-dimensional densities to each of them. In Paper 1 the temperature dependence follows a traditional form that makes statistical identification hard on the basis of the available data. Separate laboratory tests were conducted to arrive at reasonable bounds for the dependence. After that, a proper Gaussian prior was assigned to the temperature dependence parameter.

If model parameters can be given physical interpretations, it is a lot easier to asses the prior. A natural prior restriction in many models is a requirement of positivity. In the algal kinetics application (Paper 2) the growth rates of an algal population are restricted by the intake of nutrients. This makes it possible to assign (soft) upper bounds for the rate coefficients. Parametric lake quality models are typically overparametricized, as often there are not enough data to identify all the model parameters. The use of reference tables for reaction rate parameters (like in Bowie et al. [1985]) is common. The use of priors makes it possible (for good and evil) to consider models that are more complicated than would be necessary for predictive purposes.

Assessing priors can in practice be seen as an iterative process of the same kind as the building of a model. For example, if a model is described by differential equations, like the lake models in Papers 1 and 2, the initial values of the state variables might be unknown. If we have observations on the state variables at the beginning of the period, then the prior for the initial values should contain information from these observations, and also information about the accuracy of the observations. But the latter is probably available only after a preliminary fit, from the residual variance.

Priors enter the modelling in every step, not only when building the model. In the adaptive management of a lake (Paper 1), the current posterior uncertainty is used as a prior for the new observations added to the model. At the beginning of winter we must use knowledge from previous winters to formulate a prior for the current winter.

The real test of a model is its ability to predict the observations, old and new. Tarantola [2006] describes Bayesian scientific inference as a part of the Popperian paradigm in the philosophy of science [Popper, 1989], where the observations will eventually falsify both badly selected model and badly selected priors.

There exists a longstanding philosophical controversy over the nature of statistical inference and its relation to the "real world", as in the question of whether it is appropriate, or even possible, to assign a statistical prior distribution to the value of an unknown parameter in a model, or for an unknown state in nature, if these are not random variables. We write models to describe the reality, but statistical inference is about the models, not about the reality. In science we must be able to criticize all aspects of our methods and results. This includes the statistical paradigm used. Applied Bayesian statistics works very well in practice. Thanks to the computational MCMC methods, we are able to work with more complicated and realistic model than before, pool prior information from various sources, and easily produce probability statements about the predictions of the model that can be verified by future observations.

4 Markov chain Monte Carlo

High-dimensional estimation problems pose a computational challenge that can be solved only by simulation methods. The normalizing constant – the integral in the denominator of the Bayes formula (9) – makes computation of the posterior distribution in difficult problem, especially in multidimensional cases. A clever algorithm dating back to the 1950s (based on the work of Metropolis et al. [1953][1] and later Hastings [1970]), provides a simple method for simulating values from a distribution that can be calculated only up to a normalizing constant.

We will use the notation $\pi(\theta)$ for the *target distribution* of interest. This is common in the MCMC literature. In most cases the target will be the posterior distribution for the model unknowns, $\pi(\theta) = p(\theta|y)$.

In Markov chain Monte Carlo (MCMC) simulation we produce a sequence of values which are not independent but instead follow a stochastic process called a Markov chain. The algorithm used in the simulation ensures that the chain will take values in the domain of the unknown θ and that its limiting distribution will be the target distribution $\pi(\theta)$. This means that we have a method of sampling values from the posterior distribution and therefore of making Monte Carlo inferences about θ in the form of sample averages and by means of histograms and kernel density estimates.

The MCMC algorithm produces a chain of values in which each value can depend on the previous value in the sequence. For example, a random walk MCMC algorithm advances the sequence by proposing and conditionally accepting or rejecting new values by means of a proposal distribution centred at the current position of the MCMC chain. The values are not independent but instead have some positive autocorrelation. Because of this, the sample averages used as estimates for the corresponding posterior values have error due to the sampling procedure, so called Monte Carlo error, that is larger than in the i.i.d. (independent identically distributed) case. There is no general way to get rid of this problem, however. On the other hand, random walk sampling in a high-dimensional space has advantages over i.i.d. sampling. As Tarantola [2005] points out, high-dimensional spaces are very sparse. If we consider the unit hyper sphere that is located inside a unit hyper cube, we see that the total space consists almost entirely of corners and a very small part of the volume of the space is contained inside the sphere, Figure 1. We are confronted with problems of finding the regions of statistical significant probability and of exploring those regions. Random walk-type methods try to offer solutions to the problem of getting lost in the space. A remedy for larger Monte Carlo errors is to perform longer simulations than would be needed in the i.i.d. sampler case, and also to try to make the MCMC methods as efficient as possible. The latter is one of the main motivations behind the present work.

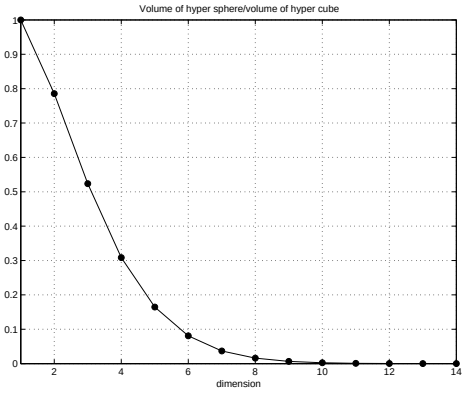


Figure 1: Volume of a hyper sphere, $2\pi^{d/2}r^d/d/\Gamma(d/2)$, divided by the volume of hyper cube, $(2r)^d$ [Tarantola, 2005].

[1] In the list of the Top Ten Algorithms of the Century published by Dongarra and Sullivan in *Computing in Science & Engineering*, Vol. 2, No. 1, 2000, Monte Carlo methods and the Metropolis algorithm are mentioned first.

4.1 The Metropolis-Hastings algorithm

The Metropolis-Hastings (MH) algorithm is currently the most general algorithm for MCMC simulation. Its basic form is easy to explain and implement and it has several useful generalizations and special cases for different purposes. The basic idea depends on the fact that, if instead of computing the values $\pi(\theta)$ we need only compute the ratio of the target at two distinct parameter values $\pi(\theta)/\pi(\theta^*)$, the integral in the Bayes formula cancels out.

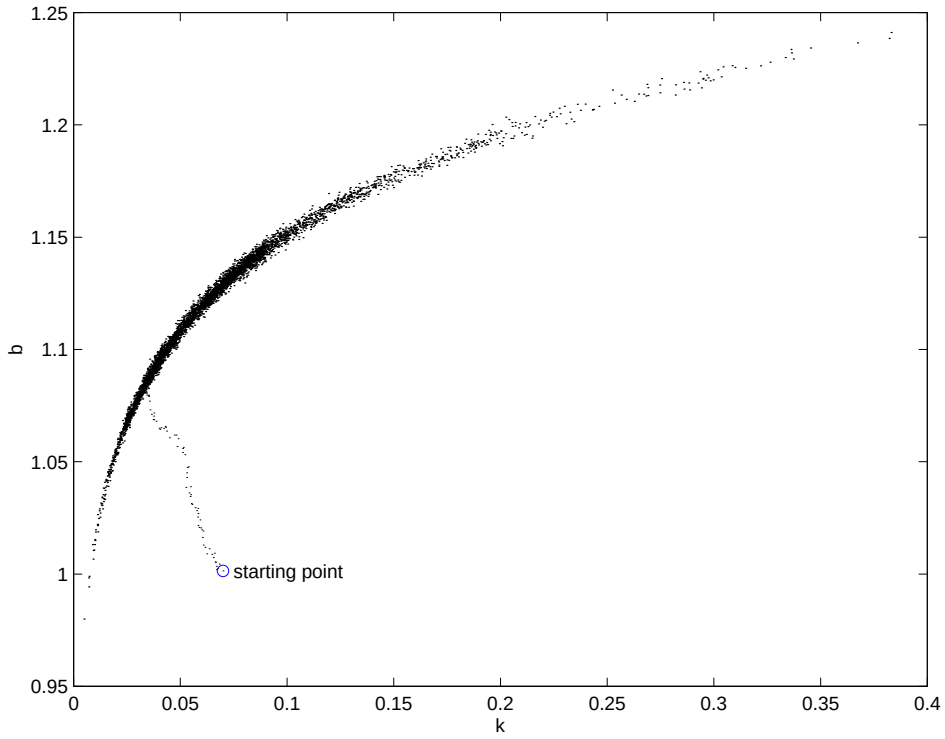


Figure 2: This scatter plot shows sample points of a two dimensional MCMC chain, that is used in Paper 1 to estimate lake oxygen consumption parameters. A heavy nonlinear dependence between the two parameters is shown. The chain starts from values that are not in the core of the posterior distribution π, but as the chain evolves the MCMC algorithm soon finds its way towards π.

With an MCMC algorithm we are generating a chain of values $\theta^0, \theta^1, \ldots, \theta^N$ in such a way that it can be used as a sample of the target density $\pi(\theta)$. In terms of the Markov chain theory [Gamerman, 1997], when using the Metropolis-Hastings algorithm we generate a Markov chain that has a transition kernel according to

$$p(\theta, \theta^*) = q(\theta, \theta^*)\alpha(\theta, \theta^*), \quad \theta \neq \theta^*$$
$$p(\theta, \theta) = 1 - \int q(\theta, \theta^*)\alpha(\theta, \theta^*)\, d\theta$$

for some transition density q, and for an *acceptance probability* α. The density $q(\theta, \cdot)$, with θ being the current location of the chain, is called the *proposal density*. The chain is said to be *reversible* if we have

$$\pi(\theta)q(\theta, \theta^*)\alpha(\theta, \theta^*) = \pi(\theta^*)q(\theta^*, \theta)\alpha(\theta^*, \theta).$$

Reversibility is a sufficient condition for the density π to be the *stationary distribution* of the chain,

$$\int \pi(\theta)\, p(\theta, \theta^*)\, d\theta = \pi(\theta^*),$$

meaning that if the chain were to reach π, it would also follow this distribution for the rest of the simulation. This leads to the choice of the Metropolis-Hastings acceptance probability α as

$$\alpha(\theta, \theta^*) = \min\left\{1, \frac{\pi(\theta^*)q(\theta^*, \theta)}{\pi(\theta)q(\theta, \theta^*)}\right\}. \tag{12}$$

The parameter space Θ is usually a subset of $\mathbb{R}^d$, but the reversibility condition can be formulated for more general state spaces.

We formulate a general Metropolis-Hastings algorithm in the following way.

Algorithm (MH)

i. Start from an initial value θ^0, and select a proposal distribution q.

ii. At each step where the current value is θ^{i-1}, propose a candidate for the new parameter θ^* from the distribution $q(\theta^{i-1}, \cdot)$.

iii. If the proposed value θ^* is better than the previous value θ^{i-1} in the sense that $\pi(\theta^*)q(\theta^*, \theta) > \pi(\theta^{i-1})q(\theta, \theta^*)$, it is accepted unconditionally.

iv. If it is not better in the above sense, θ^* is accepted as the new value with a probability α given by equation (12).

v. If θ^* is not accepted, then the chain stays at the current value, that is, we set $\theta^i = \theta^{i-1}$.

vi. Repeat the simulation from step ii until enough values have been generated.

The proposal distribution from which we choose new values for the chain can be quite arbitrary, but choosing a distribution that most closely resembles the true target distribution can dramatically speed up the convergence of the values generated to the right distribution. The closer the proposal distribution q is to the actual target $\pi(\theta)$, the better the chain mixes and the better a short sequence represents a random draw from the posterior. This is especially true in multidimensional cases and when there is correlation between the components of the parameter vector. In the applications described in this work the proposal density is taken to be the multidimensional Gaussian density.

The algorithm is constructed in such a way that the target distribution π is the stationary distribution of the Markov chain. This means that the values generated will eventually follow the posterior distribution π. In practise, we must allow some burn-in time to let the chain become close enough to the limiting distribution. The MH algorithm can be thought of as travelling uphill towards the peak of the posterior distribution, but occasionally taking steps downhill. The percentage of time spent in each region of the hill corresponding to the probabilities of the target distribution An example of samples from a two dimensional posterior drawn using the MH algorithm is shown in Figure 2.

For the convergence results we need some theory of Markov chains, allthogh with one important simplification: it is known by construction that the stationary distribution π exists. Also, we are able to choose the initial distribution arbitrarily. This provides simple ways of proving important ergodic properties of the MH chain: The Law of Large Numbers type of theorem that says that we can use sample averages as estimates and apply the Central Limit Theorem, which gives us the convergence rate for the algorithms.

We define the *ergodicity* of an MCMC sampler by requiring the following formulation of the strong Law of Large Numbers:

Definition 4.1 Let π be the density function of a target distribution in the Euclidean space $\mathbb{R}^d$. An MCMC algorithm is *ergodic* if it simulates the distribution π correctly in the sense that

$$\lim_{n\to\infty} \frac{1}{n+1}(f(\theta^0) + f(\theta^1) + \ldots + f(\theta^n)) = \int_{\mathbb{R}^d} f(\theta)\pi(\theta)\, d\theta, \tag{13}$$

almost surely, for all bounded and measurable functions $f : \mathbb{R}^d \to \mathbb{R}$ and all initial states θ_0 that belong to the support of π.

We will use a version of the MH in which the proposal will always be such that if θ^i is the current position, then the proposed values are of form $\theta^* = \theta^i + u$ so that the chain performs a random walk with increments given by some random variable u. This is called the *random walk Metropolis-Hastings* algorithm.

Theorem 4.2 Consider a random walk Metropolis-Hastings algorithm with a transition kernel $p(\cdot, \cdot)$. If the support $S \in \mathbb{R}^d$ of the target distribution π is statistically closed, $p(\theta, S) = 1$ for all $\theta \in S$, then the random walk MH algorithm is ergodic.

The proof follows from the basic Markov chain theory. A proof using a minimal amount of theory is given by Nummelin [2002].

The ergodicity theorem states that the empirical distribution calculated from the chain is a consistent estimator for π. It also says that for any measurable function f we can calculate a sample from the posterior distribution $p(f(\theta)|y)$ by sampling θ from the MCMC chain and evaluating $f(\theta)$ at each sample point. This means, for example, that if we calculate model predictions by evaluating the model while sampling parameter values from the chain, the distribution of these sampled predictions will follow the correct posterior predictive distribution.

In this work we use an adaptive version of the random walk Metropolis-Hastings algorithm with a Gaussian proposal distribution. The covariance matrix of the proposal is adapted to the target distribution by the values in the chain generated so far. We also make use of several simultaneous proposals by employing what is known as the delayed rejection method. The proposed method is shown to work well, even though the target posterior distribution may be far from Gaussian and have a nonlinear correlation structure between the unknowns. By using easily calculated proposals we are able to build up a general MCMC tool that can be applied to a variety of problems and be included in a general MCMC software toolbox.

4.2 Adaptive MCMC

The Metropolis-Hastings algorithm depends on the user's ability to choose the proposal distribution in such a way that it will propose reasonable values that have a good chance of being accepted. If we are using the multi dimensional Gaussian distribution as the proposal, then the covariance of the proposal must somehow suit the target distribution. A result of Gelman et al. [1996] shows that if the target is itself a Gaussian distribution, with covariance matrix C, then an efficient sampler can be constructed by scaling the proposal covariance by $2.4^2/d$, where d is the dimension of the target.

If we are able to tune the proposal covariance to the posterior distribution of the unknowns θ, we can usually successfully simulate it with the Metropolis-Hastings algorithm and applying a Gaussian proposal distribution. This tuning of the proposal is important in highly nonlinear situations when there is correlation between the components of the posterior, or when the dimension of the parameter is high.

One problem in adapting the proposal distribution using the chain simulated so far is that when the accepted values depend on the history of the chain, the sampler is no longer Markovian and standard convergence results do not apply. One solution is to use adaptation only for the burn-in period and discard the part of the chain for which adaptation has been used. In this respect adaptation can be thought of as an automatic burn-in. This use of adaptation, however, wastes computational resources and we would like to use as much of the generated values as possible.

It is very easy to construct an adaptive method that is not ergodic and thus will produce erroneous estimates for the posteriors. Robert and Casella [2005] give an example of an adaptive scheme where the chain does not converge to the correct distribution. The idea of diminishing adaptation is that when adaptation works well its effect becomes smaller, and we might be able to prove the ergodicity properties of a chain even when adaptation is used throughout the simulation. After the pioneering work on the ergodic properties of adaptive methods by Haario et al. [2001], adaptive MCMC methodology has become a topic of intensive research [Andrieu and Moulines, 2006, Roberts and Rosenthal, 2007].

In the Adaptive Metropolis method (AM) of Haario et al. [2001] the proposal covariance is adapted by using the history of the chain generated so far. The AM adaptation, given below, is also the basis of the DRAM method which is described later and used in most of the applications in this work.

Algorithm (AM)

i. Start from an initial value θ^0 and initial proposal covariance $C = C_0$. Select a covariance scaling factor s, a small number ε for regularizing the covariance, and an initial non-adapting period n_0.

ii. At each step, propose a new θ^* from a Gaussian distribution centred at the current value $N(\theta^{i-1}, C)$.

iii. Accept or reject θ^* according to the MH acceptance probability.

iv. After an initial period of simulation, say for $i \geq n_0$, adapt the proposal covariance matrix using the chain generated so far by

$$C = \mathrm{cov}(\theta^0, \ldots, \theta^i)s + I\varepsilon.$$

Adapt from the beginning of the chain or with an increasing sequence of values. Adaptation can be done at fixed or random intervals.

v. Iterate from ii until enough values have been generated.

Note that the Gaussian proposal q is symmetric so that it cancels out in equation (12). The regularization factor ε is a small number that prevents the covariance matrix from becoming singular.

Theorem 4.3 If the target distribution π is bounded from above and has support on a bounded measurable subset $S \subset R^d$, then the AM algorithm is ergodic.

Proof: This is **Theorem 1** in Haario et al. [2001].

The requirement for boundedness of the support may seem restrictive at first sight, but we can always approximate the targets with bounded targets to an arbitrary precision. As noted in the remarks in the article cited, it might be possible to prove the results under less stringent assumptions. All the test runs performed so far show that AM method works well with unbounded targets [Haario et al., 2001].

4.3 DRAM

This work introduces a new idea for adaptation, DRAM, which is a combination of two ideas for improving the efficiency of Metropolis-Hastings type Markov chain Monte Carlo (MCMC) algorithms, Delayed Rejection [Mira, 2001] and Adaptive Metropolis [Haario et al., 2001].

One problem with any adaptation method is that to be able to adapt the proposal at all, we need some accepted values to start with. For the AM method it is necessary to have an initial proposal that is feasible enough to initiate the adaptation process. If the initial proposal is too "wide", no new points will be accepted and there will be no history to adapt to. An initial burn-in with scaling of the proposal according to an acceptance ratio is straightforward to implement, but then the initial period of the chain must be discarded. DRAM aims at providing an automatic burn-in with very little initial tuning and short burn-in periods.

4.3.1 Delayed rejection

There is a way to make use of several tries in the MH algorithm after rejecting a value, use a different proposal in each try, and still keep the reversibility of the chain. The delayed rejection method (DR) of Mira [2001] works in the following way. Upon rejection of a proposed candidate point, instead of advancing in time and retaining the same position, a second move is proposed. The acceptance probability of the second stage candidate is computed so that the reversibility of the Markov chain relative to the distribution of interest is preserved. The process of delaying the rejection can be iterated for either a fixed or a random number of times, and the higher-stage proposals are allowed to depend on the candidates so far proposed and rejected. Thus DR allows partial local adaptation of the proposal within each time step of the Markov chain, still retaining the Markovian property and reversibility.

The first stage acceptance probability in DR is the standard MH acceptance, which can be written as

$$\alpha_1(\theta, \theta^*) = \min\left\{1, \frac{\pi(\theta^*)q_1(\theta^*, \theta)}{\pi(\theta)q_1(\theta, \theta^*)}\right\}. \tag{14}$$

Here θ is the current point, θ^* is the proposed new value drawn from the distribution $q_1(\theta, \cdot)$, and π is the target distribution.

If θ^* is rejected, a second candidate θ^{**} is drawn from $q_2(\theta, \theta^*, \cdot)$ using the acceptance probability

$$\alpha_2(\theta, \theta^*, \theta^{**}) = \min\left\{1, \frac{\pi(\theta^{**})q_1(\theta^{**}, \theta^*)q_2(\theta^{**}, \theta^*, \theta)[1 - \alpha_1(\theta^{**}, \theta^*)]}{\pi(\theta)q_1(\theta, \theta^*)q_2(\theta, \theta^*, \theta^{**})[1 - \alpha_1(\theta, \theta^*)]}\right\}. \tag{15}$$

As the reversibility property is preserved, this method also leads to the same stationary distribution π as the standard MH algorithm. The procedure can be iterated further for higher-stage proposals. Paper 4 has the details.

The smaller overall rejection rate of DR guarantees smaller asymptotic variance of the estimates based on the chain, by having smaller autocorrelation. A DR chain can be shown to be asymptotically more efficient than the standard MH chain in the sense of Peskun ordering [Mira, 2001].

4.3.2 Combining AM and DR

By combining the AM and DR algorithms we obtain a method called DRAM (Paper 4). The use of different proposals in DR manner and adapting them as in AM makes it possible to have many different implementations of this idea. We have used one quite straightforward possibility. Firstly, one master proposal is tried. After a rejection, we try with a modified version of the first proposal according to the DR. The second proposal can have a smaller covariance matrix, or a different orientation of the principal axes. The master proposal is adapted using the chain generated so far, and the second stage proposal follows the adaptation in an obvious manner. The use of different Gaussian proposals makes it possible for the sampler to provide better results for non-Gaussian target distributions. We

can interpret the AM as a global adaptation procedure and the DR's ability to use the rejected values as a local adaptation to the current location of the target distribution.

A simple, but useful implementation of DRAM is described in the following.

Algorithm (DRAM)

i. Start from an initial value θ^0 and initial first stage proposal covariance $C^{(1)} = C_0$. Select the scaling factor s, covariance regularization factor ε, initial non-adaptation period n_0, and scalings for the higher-stage proposal covariances $C^{(i)}$, $i = 1, \ldots, N_{\text{try}}$, where N_{try} is the number of tries allowed.

ii. DR loop. Until a new value is accepted, or N_{try} tries have been made:

   (a) Propose θ^* from a Gaussian distribution centred at the current value $N(\theta^{i-1}, C^{(k)})$.

   (b) Accept according to the k'th stage acceptance probability.

iii. Set $\theta^i = \theta^*$ or $\theta^i = \theta^{i-1}$, according whether we accept the value or not.

iv. After an initial period of simulation $i \geq n_0$, adapt the master proposal covariance using the chain generated so far

$$C^{(1)} = \text{cov}(\theta^0, \ldots, \theta^i)s + I\varepsilon.$$

   Calculate the higher-stage proposal as scaled versions of $C^{(1)}$, according to the chosen rule.

v. Iterate from ii onwards until enough values have been generated.

NOTES:

- The above algorithm dos not use information on the rejected points. This information could be used, for example, if the value is rejected by being beyond the bounds of the parameter space, in order to prefer proposal directions leading away from the boundary.
- Although we use a Gaussian distribution for the proposal, the ratio $q_1(\theta^{**}, \theta^*)/q_1(\theta, \theta^*)$ in the second acceptance probability does not cancel out, and must be calculated explicitly.

We have a similar result for the ergodicity of the DRAM algorithm as for the AM.

Theorem 4.4 If the target distribution π is bounded from above and has support on a bounded measurable subset $S \subset R^d$, then the DRAM algorithm is ergodic.

This is **Theorem 4** in Paper 4 (DRAM).

4.3.3 Example

A computer experiment is used here to demonstrate the combinations of the four methods presented : Plain Metropolis-Hastings (MH), Delayed Rejection (DR), Adaptive Metropolis (AM), and DRAM.

We use use a "banana-shaped" 2-dimensional target distribution that can be constructed as follows. We start with a 2-dimensional Gaussian distribution with zero mean, unit variances and a correlation between the components equal to ρ (=0.9 in the example below). The Gaussian coordinates x_1 and x_2 are then twisted to produce a more nonlinear target, using the equations

$$\begin{aligned} y_1 &= ax_1 \\ y_2 &= x_2/a - b(a^2x_1^2 + a^2), \end{aligned} \tag{16}$$

As the determinant of the transformation is 1, we can easily calculate the correct probability regions of the nonlinear target and study the behaviour of the generated chain.

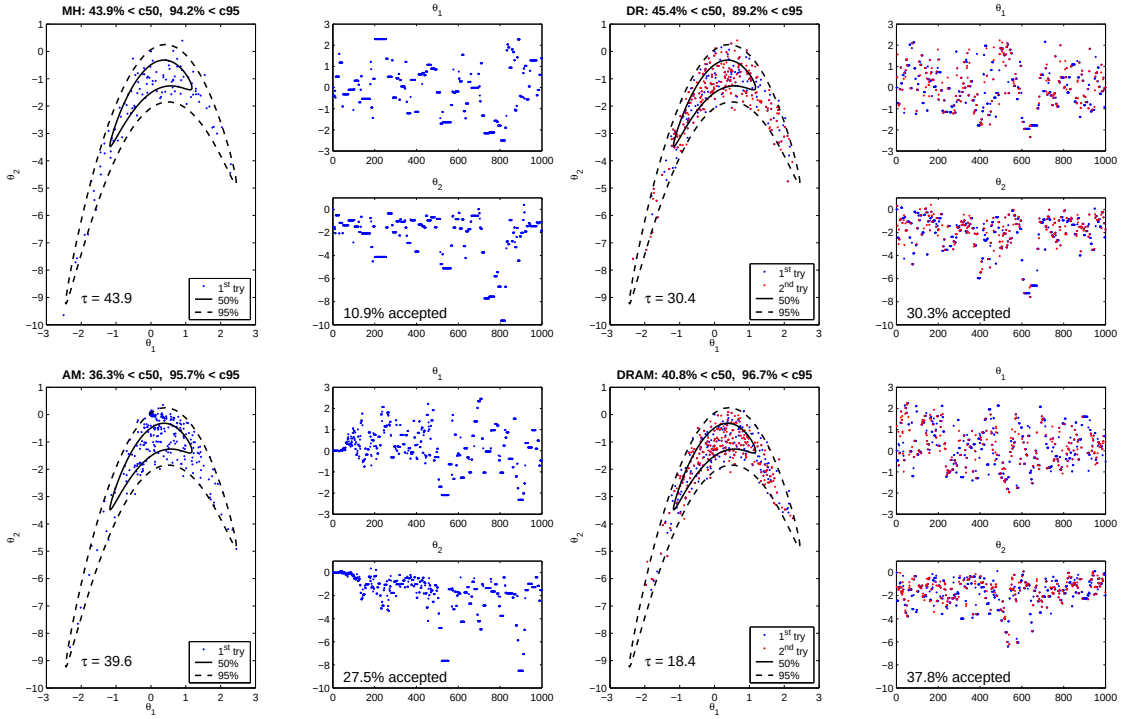


Figure 3: 2-D plots of the chains generated by the four methods. The contours give correct 50% and 95% regions. The percentages in the plot titles show how many points lie inside the regions.

We used Gaussian proposals with the identity matrix as the initial proposal covariance. A chain of length 1000 was generated by each of the methods MH, DR, AM and DRAM. Mixing of the chain was studied by counting the numbers of points that are inside the 50% and 95% probability regions.

As we generated only 1000 points, the percentages observed would vary from simulation to simulation, but we can see that all the methods are working correctly as expected. The initial proposal for the plain MH is not optimal and the acceptance ratio remains at about 10%. In addition, the chain does not have enough time to mix properly. DR without adaptation but with a smaller 2$^{\text{nd}}$ proposal ($N_{\text{try}} = 2$) makes the chain accept more points, which helps the mixing, the acceptance ratio being about 30%. Both AM and DRAM make the chain mix more quickly, due to adaptation, and we see that more distant points of the target are also visited. The covariance shrink factor for the DR proposal is 2, so $C^{(2)} = C^{(1)}/2^2$.

The number τ in the plots is the integrated autocorrelation time, giving an estimated increase in the asymptotic variance of the chain-based estimates relative to an i.i.d. sample [Sokal, 1996]. The number τ also varies from run to run, DRAM being the most efficient in general. The result $\tau = 18.4$ obtained by the DRAM method means approximately that if every 18$^{\text{th}}$ point is taken from the chain the sample will behave like an i.i.d. sample from the target distribution.

5 Models and data

As pointed out in the Introduction, the main theme of this work is to develop adaptive MCMC methods and to apply them for the statistical analysis of certain nonlinear models. The statistical analysis is employed to assess both the uncertainty in the model parameters and the uncertainty in the model-based

predictions. To explain the implementation of the adaptive MCMC, consider the following version of the basic model equations. We have observations y from the model f with additive independent Gaussian errors having an unknown variance σ^2

$$y = f(x, \theta) + \epsilon, \quad \epsilon \sim N(0, I\sigma^2). \tag{17}$$

The vector $\theta \in \mathbb{R}^d$ contains all the unknown variables in the model, and matrix x the known control variables. We have n independent observations. We assume independent Gaussian prior specifications for θ:

$$\theta_i \sim N(v_i, \eta_i^2), \tag{18}$$

For the error variance a Gamma distribution is used as a prior for its inverse

$$p(\sigma^{-2}) \sim \Gamma\left(\frac{n_0}{2}, \frac{n_0}{2} S_0^2\right). \tag{19}$$

We consider first the implementation of the adaptive MCMC for this basic model and later discuss its immediate extensions.

5.1 Implementing MCMC

The likelihood function $p(y|\theta, \sigma^2)$ for n i.i.d. observations from model equation (17) with a Gaussian error model is

$$p(y|\theta, \sigma^2) = (2\pi)^{-\frac{n}{2}} \sigma^{-n} \exp\left\{\frac{-1}{2\sigma^2} SS(\theta)\right\}, \tag{20}$$

where $SS(\theta)$ is the sum of squares function

$$SS(\theta) = \sum_{i=1}^{n} (y_i - f(x_i; \theta))^2. \tag{21}$$

Modelling of the error term ϵ is an important part of the modelling process. Although independent Gaussian errors are well suited in many situations, their use should be justified. The error variance σ^2 can often be treated as a "nuisance" parameter independent of the other unknowns in θ. Sometimes it is even considered fixed, but in most cases a prior distribution for it should be used and the corresponding posterior distribution calculated.

One practical way of specifying a prior for σ^2 depends on what is called the conjugacy property. The form of the Gaussian likelihood (20) when considered as a function of σ^2 is seen to correspond to a Gamma-type distribution for $1/\sigma^2$. If the prior for $1/\sigma^2$ is also a Gamma distribution, then the product of the likelihood and the prior will again have the same form. This means that if we use a prior for σ^2, as in equation (19), then the conditional distribution $p(\sigma^{-2}|y, \theta)$ will also be a Gamma with

$$p(\sigma^{-2}|y, \theta) = \Gamma\left(\frac{n_0 + n}{2}, \frac{n_0 S_0^2 + SS(\theta)}{2}\right). \tag{22}$$

This conditional conjugacy property makes it possible to sample and update σ^2 within each Metropolis-Hastings simulation step for the other parameters. In general, this kind of conditional MCMC sampling of one component of the unknowns is called Gibbs sampling.

The prior parameters S_0^2 and n_0 in (19) can be interpreted as the prior mean for σ^2 and the prior accuracy as imaginary observations. The distribution of $1/\sigma^2$ is called an inverse χ^2 distribution in

Gelman et al. [1995]. This type of prior is also reasonable because in many cases we build the prior for σ^2 using the residual variance from the previous experiments (or from "imaginary" experiments). It is also in line with the classical analyses of error variance using χ^2 confidence intervals.

If the prior distribution for θ is Gaussian, it can be treated as an extra sum-of-squares. If all the priors are independent, as in equation (18), then we calculate the prior sum-of-squares for the given θ according to

$$SS_{\text{pri}}(\theta) = \sum_{i=1}^{p} \left(\frac{\theta_i - \nu_i}{\eta_i} \right)^2 . \tag{23}$$

Then, assuming σ^2 has a fixed value, the posterior for θ has the form

$$p(\theta|y, \sigma^2) \propto \exp\left\{ -\frac{1}{2} \left(\frac{SS(\theta)}{\sigma^2} + SS_{\text{pri}}(\theta) \right) \right\}, \tag{24}$$

and the posterior ratio needed in the Metropolis-Hastings acceptance probability can be written as

$$\frac{p(\theta^2|y, \sigma^2)}{p(\theta^1|y, \sigma^2)} = \exp\left\{ -\frac{1}{2} \left(\frac{SS(\theta^2)}{\sigma^2} - \frac{SS(\theta^1)}{\sigma^2} \right) + \frac{1}{2} \left(SS_{\text{pri}}(\theta_2) - SS_{\text{pri}}(\theta_1) \right) \right\}. \tag{25}$$

For computation purposes we need a routine to return the sum-of-squares $SS(\theta)$ for a given parameter and data. In some cases this means numerically solving the differential equations describing the system, which will be the most computation-intensive part. The same routine would also be needed when carrying out classical nonlinear least squares minimization estimation.

Algorithm (Adaptive Metropolis for the basic model)

i. Initialization: Choose initial values for the model parameters θ, error variance σ^2, and a proposal covariance C. Choose prior parameters for θ and error variance prior parameters n_0 and S_0^2.

ii. MH step: Let θ be the current value. Generate a new value θ^* from $N(\theta, C)$, calculate the sum-of-squares for it, $SS(\theta^*)$, and calculate the prior sum-of-squares, $SS_{\text{pri}}(\theta^*)$. The new value is accepted if

$$u \leq \exp\left\{ -\frac{1}{2\sigma^2} (SS(\theta^*) - SS(\theta)) + \frac{1}{2} \left(SS_{\text{pri}}(\theta^*) - SS_{\text{pri}}(\theta) \right) \right\},$$

where u is generated from a uniform $U(0, 1)$ distribution and σ^2 is the current value of the error variance.

iii. Update σ^2 with a direct draw from the conditional distribution of $p(\sigma^2|\theta, y)$, i.e. draw $1/\sigma^2$ from Gamma ditribution

$$1/\sigma^2 \sim \Gamma\left(\frac{n_0 + n}{2}, \frac{n_0 S_0^2 + SS(\theta)}{2} \right).$$

iv. Adapt C as in AM.

v. Return to ii until enough values have been sampled.

The algorithm in its basic form is amazingly simple, considering that it replaces the nonlinear least squares optimization, it works in relatively high dimensions, and it provides much more information about the unknowns than the classical methods. Even in this basic form it can be used to analyse large numbers of quite complicated problems. All the applications described in this work were analysed with

this algorithm (with the DRAM additions discussed below). The calculations and posterior analyses were performed with the Matlab [Mat, 2000] programs described in the Appendix. For some model calculations Fortran code was used to speed up the computations.

The adding of DRAM acceptance makes the algorithm appear a little more complicated. The modifications for the basic DRAM of section 4.3.2 are the following.

To calculate the higher-stage acceptance probabilities we need the values of additional backward probabilities. For example, for $\alpha_2(\theta, \theta^{**})$ we need to calculate the backward probability $\alpha_1(\theta^{**}, \theta^*)$. The computational complexity of DR increases as a factorial of the number of stages. So, for a 5-stage proposal, for example, we need to evaluate 120 acceptance probabilities and the benefits are usually not worth the trouble. One or two extra tries would typically be enough. All these acceptance probabilities depend on similar posterior ratios to the one in equation (25).

In the DRAM proposal ratios, even when using symmetric proposals in each individual step, the proposals do not cancel out. This will cause some extra matrix vector product calculations and a need for the inverse of the proposal covariance. For the second stage acceptance, for example, we need to calculate

$$\frac{q_1(\theta^{**}, \theta^*)}{q_1(\theta, \theta^*)} = \exp\left\{-\frac{1}{2}(\theta^{**} - \theta^*)'(C^{(1)})^{-1}(\theta^{**} - \theta^*) + \frac{1}{2}(\theta - \theta^*)'(C^{(1)})^{-1}(\theta - \theta^*)\right\}. \quad (26)$$

A Matlab code with a recursive function for the DRAM acceptance probabilities is given in the Appendix.

NOTES:

- Initial values for θ and for the proposal covariance can sometimes be acquired by an initial least squares fit of the sum-of-squares function, usually even without using prior information. If the least squares fit can be done, an initial covariance can be formed from the numerical Jacobian matrix J of the model near the least squares point, as $(J'J)^{-1}s^2$, where s^2 is an estimate of the residual variance.
- The initial proposal covariance can even be a diagonal matrix with some guesses on the (relative) sizes of the posterior uncertainties. We can then rely on the DRAM method to adjust the proposal covariance matrix.
- The positivity constraints in the unknowns of the models can be implemented by rejecting those proposed values for which any component is nonpositive. If the unknowns are identified sufficiently well, the adaptation ensures that only a few tries are made with out-of-bounds unknowns. The DRAM sampler is also useful in this respect, as a new DR try can be made after a rejection due to a prior restriction. In this case $p(\theta^*) = 0$ for the rejected value, and some simplifications for the next stage proposal formula are available.
- We have written the formulae for Gaussian likelihoods given by the sum-of-squares function, $SS(\theta)$, as this formulation has been used in all of the applications. For a general likelihood function the sum-of-squares corresponds to twice the log likelihood, $-2\log(p(y|\theta))$. For a Poisson likelihood, for instance, and with a model that returns the expected Poisson counts, $E(N_i) = f(x_i; \theta)$, $i = 1, \ldots, n$, the "sum-of-squares" function would be

$$SS(\theta) = 2\sum_{i=1}^{n}\left[f(x_i; \theta) - N_i \log(f(x_i; \theta))\right], \quad (27)$$

  where N_i, $i = 1, \ldots, n$ are the observed counts. Then, by setting $\sigma^2 = 1$, the algorithm would work as such for the Poisson case. Many other standard distribution can be used similarly.

- To obtain a general non-Gaussian prior for θ, we have to calculate minus twice the log of the prior density, corresponding to the "sum-of-squares" in the algorithm. For the log-normal prior density $\theta_i \sim \log N(\log(\nu_i), \eta_i^2)$, for example, we would need to calculate

$$SS_{\text{pri}}(\theta) = \sum_{i=1}^{p} \left(\frac{\log(\theta_i/\nu_i)}{\eta_i} \right)^2 . \tag{28}$$

- The code also works when the observation y has several components, e.g. when y is a matrix with different error variances for each column. This is the case, for example, for the algae model on Paper 2.

5.2 Using the MCMC chain

After the MCMC run we have a chain of values of the unknown at our disposal. According to the theory behind MCMC, the chain can be used as a sample from the posterior distribution of the unknowns. If we think of the generated chain as a matrix where the number of rows corresponds to the size of the MCMC sample and the number of columns corresponds to the number of unknowns in the model, then each row is a possible realization of the model, and the rows appear in the correct proportions corresponding to the posterior distribution. Many useful calculations can be based on this sample. The mean of the posterior, the best Bayesian point estimate (with respect to the square error loss function), is calculated in terms of the column means of the MCMC chain matrix, the posterior standard deviations of the estimates are the column standard deviations of the matrix, and so on. The actual posterior distribution of the unknowns can also be explored using the MCMC chain. For example, plotting two-dimensional scatter plots of the sampled values in the chain produces a representation of the corresponding two dimensional marginal posterior density. More refined estimates of the underlying posteriors can be acquired by kernel density estimation methods [Silverman, 1986]. An example of a two dimensional marginal chain with posterior density estimates obtained by the kernel density method, is shown in Figure 4.

If we calculate some characteristic of the model depending on the observations and the parameters for each row of the MCMC matrix, then this sample will again follow the right distribution, called the predictive distribution. To illustrate this use of the MCMC chain, consider the oxygen consumption model of Paper 1. One winter period (1980–1981) is shown in Figure 5. The set of lines is a sample of fitted models from the MCMC chain. One line corresponds to one sampled value of the parameter vector. From these model lines we calculated a 90% envelope within which the oxygen concentration predicted by the model will lie. These limits, together with the mean prediction, are given by the thick black lines.

5.2.1 Uncertainty in the predictions

In the model formulation $y = f(x;\theta) + \epsilon$ the uncertainty of the predictions for y comes from two main sources, the error arising from the uncertainty in the model parameters θ and the error arising from the noise term ϵ. A third source, error in the model, will be also considered.

The uncertainty due to θ is covered by the posterior distribution of θ and the variability due to ϵ is covered by the (Gaussian) error model and the posterior distribution for σ^2. We must consider them both, when making model based predictions.

If we think about the model as a true description of the underlying phenomenon, we can take ϵ to be uninteresting noise. The predictions are then based solely on the model. Usually, however, the

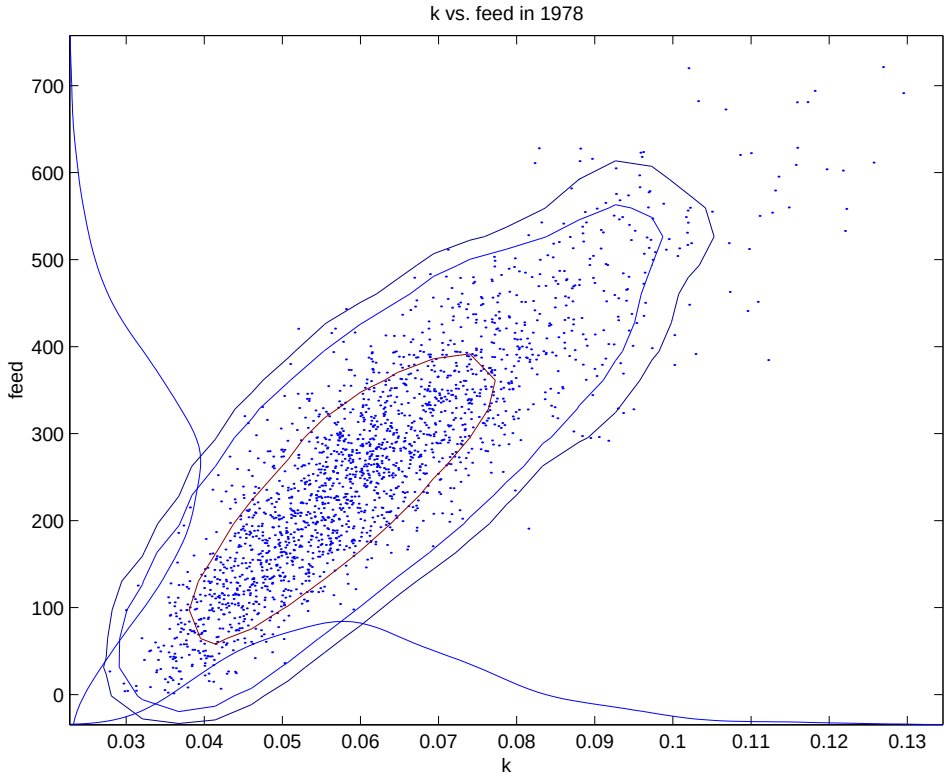


Figure 4: Two-dimensional posterior density of the respiration rate constant in 1978 versus the feed of extra oxygen pumped [kg d^{-1}] in the oxygen consumption model of Paper 1. The small dots are points in the MCMC chain for which the density contour lines corresponding to 50%, 90% and 95% levels are calculated using the kernel density estimation method. The marginal densities of the parameters are also shown by the x and y axes.

error term ϵ contains unmodelled features of the system and we must assume that the variation in the observations due to these unmodelled features will be non-systematic and smaller than the variation due to the systematic part of the model. In this case the observations might be accurate but the model is not. When the model f is just an approximation of the true model, then the ϵ term will also contain sources of error due to the modelling error. When predicting algal concentrations in Paper 2, for example, we are interested in the uncertainty attached to new algae observations. Then the prediction error must take account of the ϵ term in the model.

On the other hand, if the observational error comes mainly from the measuring device or from the sampling procedure, then the underlying mean behaviour governed by the model will be of interest. This would be the case in modelling local lake oxygen concentrations or in the inversion of atmospheric gas profiles (Papers 1 and 3).

The fit and predictive regions for the algae model of Paper 2 are illustrated in Figure 6. In predictive plots for such dynamical system models we draw two posterior regions: the 95% posterior uncertainty due to the model parameters and a larger 95% posterior uncertainty for new observations. The accuracy and fit diagnostics of the model can be studied by considering the width of these regions. The 95% "predictive envelope" for new observations, for example, should contain approximately 95% of the observations. Failure to do this in some parts of the data signals a lack of fit.

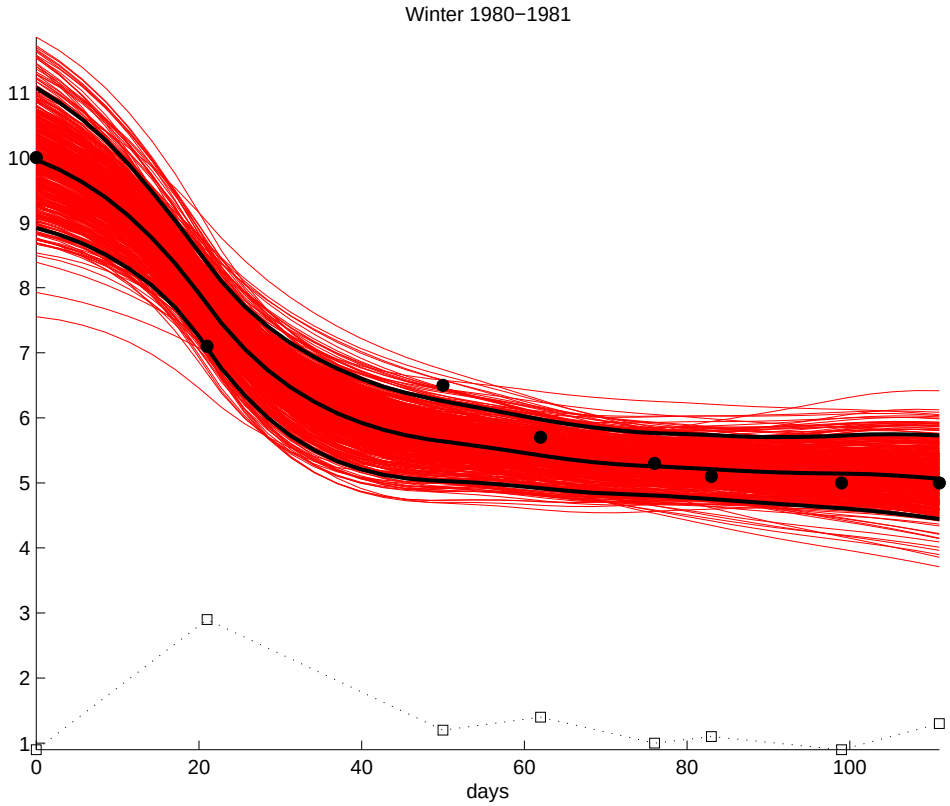


Figure 5: The ice period in winter 1980-1981 in the oxygen consumption model for Lake Tuusulanjärvi (Paper 1). The dots are the observed oxygen concentrations [mg l^{-1}], the set of narrow solid lines is a sample of fitted models from the MCMC chain. The thick solid lines gives a 90% envelope within which the fitted values will lie. The thick solid line in the middle gives the posterior mean values of the parameters k and b. The squares with a dotted line correspond to the observed water temperatures [°C], with values given on the y axis.

6 Model selection

In the Bayes formula (9) we have implicitly assumed that the likelihood, and thus the model itself, are known. The error in choosing the right model is taken to be small with respect to the accuracy of the observations. Models are always approximations, but we assume that the model we choose will give us sufficiently accurate predictions. Model choice, model comparison, and goodness of fit are important topics that also have been objects of a great deal of research [Dellaportas et al., 2002, Spiegelhalter et al., 2002].

One direct way of including the uncertainty in the model in the Bayesian modelling framework is to regard the model as an unknown quantity of the same kind as an unknown parameter. We assign prior probability to it and calculate various posterior quantities for the inference. When comparing two models M_1 and M_2, the posterior odds

$$\frac{p(M_1|y)}{p(M_2|y)} = \frac{p(y|M_1)}{p(y|M_2)} \frac{p(M_1)}{p(M_2)}, \tag{29}$$

would provide a direct comparison of their posterior probabilities depending on the ratio of their probabilities and the *Bayes factor* $p(y|M_1)/p(y|M_2)$. Calculation of the conditional probabilities $p(y|M_i)$ requires integration over the space of the unknowns θ, for which the MCMC methods can be used.

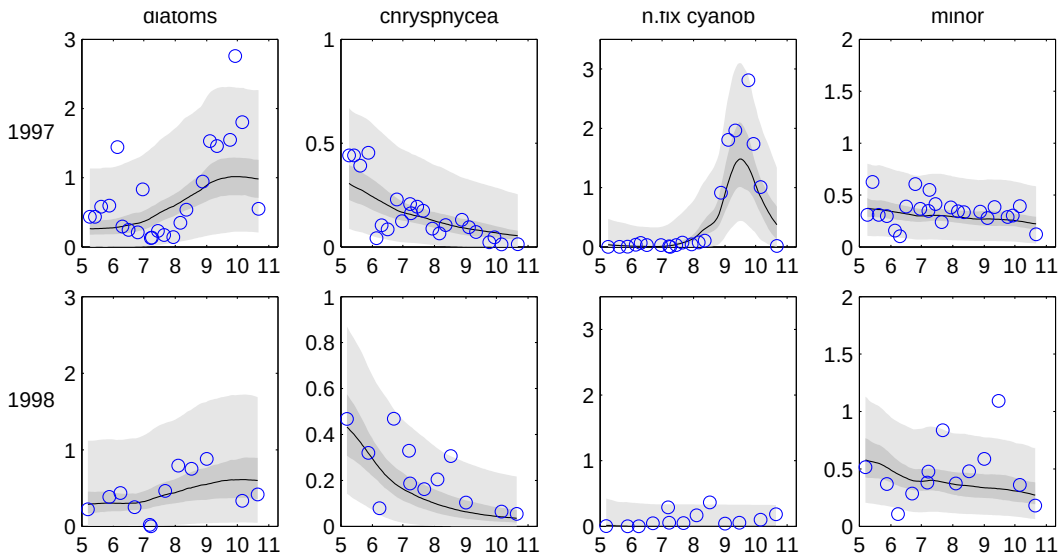


Figure 6: Plots of the fitted algae model together with the uncertainties for two years 1997 and 1998 (Paper 2). Circles (o) present the observed algae wet biomass concentrations [mg L^{-1}]. The solid lines show the median fits. Darker areas correspond to the 95% posterior limits of the model uncertainty, while the lighter areas present the same uncertainty level in predicting new observations. The horizontal axis specifies the months of the year.

In principle, the number of possible models is infinite and we cannot expect a finite data set to provide information for very detailed model determination. Thus there will clearly be no complete solution to the model selection problem. Science advances by proposing models that are put to the test with observations. Often it is reasonable, however, to compare models and make probability statements about their relative merits with respect to the available data and the prior considerations.

6.1 Reversible Jump MCMC

If we want to use MCMC for a modelling problem where the model itself is also an unknown, we need to consider a general state space which allows the chain to move, or "jump", from one model to another. Typically, each model can have a different number of unknowns. We would then need eitherto consider a large common model that includes all the specific models as special cases, or to try to make an MCMC sampler that can handle changes of dimension in the unknown.

A general formulation of the Metropolis-Hastings algorithm that allows the chain to explore spaces of varying dimensions, thus allowing for a model choice, is called Reversible Jump MCMC (RJMCMC) [Green, 1995, Tierney, 1998]. The change of dimension is handled by defining the model jumps by means of a suitable parameter transformation and with auxiliary random variables in such a way that the reversibility condition of the sampler is retained.

The RJMCMC method has been used successfully for many complicated models, but in its general formulation, the parameter transformations for the jumps between models and the corresponding Jacobians must be calculated explicitly, which makes it hard to implement. Green [2003] has proposed a simple version of the RJMCMC method, called automatic RJMCMC, in which the jumps between the models are implemented via Gaussian approximations of the targets. This allows for an easy calculation of the transformations and Jacobians. These approximations are used only for the calculation of the transformations, as the sampler itself uses the exact targets. The method is quite general, and no

model specific tuning is needed for implementing the move between models. As Green points out, this method will probably be practical only for model selection problems where the number of models is small. We find it useful, however, in many environmental models where we have to choose between, say, 2 to 10 competing models to describe the same phenomena and want the data to decide which ones are supported best by the available data. For example, a suitable aerosol cross-section model for the GOMOS inversion depends on the size distribution of the aerosols at a particular height. We have a set of possible approximate models for describing the different situations, and the RJMCMC can be used to select the one that has the best evidence in its favour in the current location. Or we can use an averaged model, in which all the available cross-section models are used according to their posterior weights.

6.2 Adaptive automatic RJMCMC – AARJ

The automatic RJMCMC [Green, 2003] constructs a random walk Metropolis-Hastings sampler that jumps between a number of models. Suppose that for each model M_i, with an unknown model parameter vector $\theta^{(i)}$, the conditional target posterior distribution $p(\theta^{(i)}|i)$ is approximated by a mean vector μ_i and a covariance matrix $C_i = R_i^T R_i$, where R_i denotes the Cholesky decomposition factor, and let the dimension of $\theta^{(i)}$ be d_i. A scaled and normalized version of the unknown parameter can be computed as

$$z^{(i)} = (\theta^{(i)} - \mu_i) R_i^{-1}. \tag{30}$$

When model M_j has the same dimension as model M_i, a transformation from $\theta^{(i)}$ to $\theta^{(j)}$ can be written as

$$\theta^{(j)} = \mu_j + z^{(i)} R_j. \tag{31}$$

When the dimensions of the two models do not match, we either drop some components or add a new one using independent Gaussian random variables, $u \sim N(0, I)$, and arrive at

$$\theta^{(j)} = \begin{cases} \mu_j + \left[z^{(i)}\right]_1^{n_j} R_j & \text{if } d_i > d_j \\ \mu_j + z^{(i)} R_j & \text{if } d_i = d_j \\ \mu_j + \begin{bmatrix} z^{(i)} \\ u \end{bmatrix} R_j & \text{if } d_i < d_j. \end{cases} \tag{32}$$

Here $[z]_1^i$ means the first i components of the vector z.

The acceptance probability for a move from model M_i to model M_j is calculated according to the RJMCMC. Let $p(i, j)$ be the probability of proposing a jump to model M_j when the chain is currently at model M_i. If model M_j, with $i \neq j$, is drawn, then the current parameter vector is transformed to the new model according to equation (32). The acceptance probability for the automatic RJMCMC sampler can be written as

$$\alpha(\theta^{(i)}, \theta^{(j)}) = \max\left(1, \frac{p(y|\theta^{(j)}, j)p(\theta^{(j)}, j)p(j, i)}{p(y|\theta^{(i)}, i)p(\theta^{(i)}, i)p(i, j)} \frac{|R_j|}{|R_i|} g\right), \tag{33}$$

where $p(y|\theta^{(j)}, j)$ is the likelihood of model M_j and $p(\theta^{(j)}, j) = p(\theta^{(j)}|j)p(j)$ is the joint prior for the unknowns of model M_j and for model M_j itself. $|R|$ is the determinant of the matrix R. The last term g depends on the auxiliary Gaussian random vector u:

$$g = \begin{cases} \phi(u) & \text{if } d_i > d_j, \\ 1 & \text{if } d_i = d_j, \\ \phi(u)^{-1} & \text{if } d_i < d_j, \end{cases} \tag{34}$$

where ϕ is the probability density function of independent multi-dimensional Gaussian values, $N(0, I)$.

For a move inside the same model, we use a Gaussian proposal distribution and the standard MH acceptance probability. The proposal distribution for moves inside models and the Gaussian approximation for the moves between models are closely linked. We can use a scaled version of the covariance of the Gaussian approximation for the proposal covariance. As in AM or DRAM, we use scaling according to Gelman et al. [1996] with the Cholesky factor of the proposal covariance equal to $R2.4/\sqrt{d}$, where R is the Cholesky factor of the Gaussian approximation and d is the dimension of the target distribution.

We propose the following version of an *adaptive automatic reversible jump MCMC* for the model selection and model averaging problem for a fixed number of models $\{M_1, \ldots, M_k\}$.

Algorithm (AARJ)

i. Run separate DRAM chains for all the proposed models. Collect the mean vectors $\mu^{(i)}$ and the Cholesky factors $R^{(i)}$ of the covariance matrices of the chains, $i = 1, \ldots, k$.

ii. Run automatic RJMCMC using the target approximations calculated in step i.

iii. If the current model is kept, use the standard random walk MH with Gaussian proposal distribution, such that the proposal covariance depends on R_i in the current model.

iv. After the given (random or fixed) intervals, adapt each model approximations independently by the AM method using those parts of the chain generated so far that belong to the particular model.

The AARJ algorithm needs very little extra computation relative to the separate MH or DRAM runs for each model. We already have to maintain the mean of the chain μ and the Cholesky factor R of the covariance matrix of the chain for the adaptation of the proposal distribution. For the RJMCMC we only need to calculate the model likelihoods, or sum-of-squares functions, separately for each of the models. These routines are therefore the same as those used for the separate runs.

The success of this algorithm depends on how well the Gaussian approximations are able to provide decent proposals for moves from one model to another. Our experience so far has been promising. If a general DRAM-type MCMC sampler performs well for the individual models, then it is expected that the AARJ sampler should work as well for the coresponding model selection problem. The AARJ algorithm is used in Paper 5 to select between a set of aerosol cross-section models in the GOMOS inversion problem.

7 Applications

The methods described here have been used in three large modelling applications, as described in detail in the accompanying articles. Short introductions to the application are giwen below, with a view to the particular model and corresponding Bayesian MCMC analysis. The development of the process of modelling is also described in more detail than in the articles.

7.1 Oxygen depletion model

The aim of this work was to estimate the temporal evolution of winter respiration in Lake Tuusulanjärvi and to assess the long-term impacts of artificial aeration and a reduction in loading. Adaptive MCMC methods for studying the uncertainties in the results were developed and tested.

The data consisted of 30 years of temperature and oxygen concentration values during the ice-covered period in the interval 1970–2002 (the data for 1972 and 1976 were not used). The effect of artificial oxygenation was studied using the following oxygen consumption model:

$$\frac{dC_{O_2}}{dt} = k_{\text{year}} C_{O_2} b^{T_{\text{obs}} - T_{\text{ref}}} + \frac{\text{Pump}}{\text{Vol}}, \tag{35}$$

with

C_{O_2}	oxygen concentration in the lake (mg l^{-1})
k_{year}	yearly total respiration rate constant (d^{-1})
b	temperature coefficient of the respiration rate
T_{obs}	observed temperature of lake water (°C)
T_{ref}	reference temperature (4°C)
Pump	pumped oxygen flux (kg O_2 d^{-1})
Vol	volume of aerator impact (m^3)

As the system was modelled with ordinary differential equations, the initial concentrations $C_{O_2}(t_0)$ were also taken as unknowns. Thus the initial concentrations, yearly rate coefficients k_{year}, common temperature coefficient b and error variance σ^2 together involved a total of 62 unknowns. The Adaptive Metropolis algorithm was used for the MCMC analysis, in addition to which an MCMC run was performed with the plain Metropolis-Hastings using the eventual proposal covariance that the AM had ended up with in order to confirm the adaptive method. The estimated posterior distributions of the yearly rate coefficients are shown as box plots[2] in Figure 7.

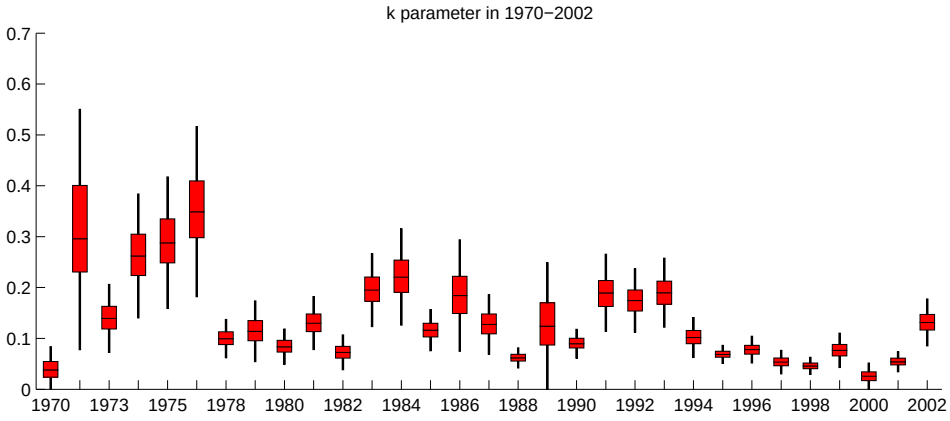


Figure 7: Box plots of the posterior distribution of the fitted respiration rate constant k [d^{-1}] of the lake model (35) for each year.

One special feature of the model was that the control variable Pump and the volume of aerator impact, Vol, were only known imprecisely based on the manufacturer's estimates. A Gaussian prior was attached to the oxygen feed term separately for each of the four aeration periods in the data.

Two yearly values were of interest: the respiration rate k_{year} and the actual average respiration. The former is a model parameter and the latter can be calculated from the model given the parameters. The trends in respiration and in the respiration rate coefficient time series were of interest because they showed the evolution of the trophic state of the lake with and without the external influences of temperature and aeration. For this, each individual time series calculated from the chain was smoothed

[2]In the box plot the middle vertical line inside the box gives the median value of the distribution, the box box reaches from 25% to 75% probability limits and the black line gives approximately the 95% limits of the distributions.

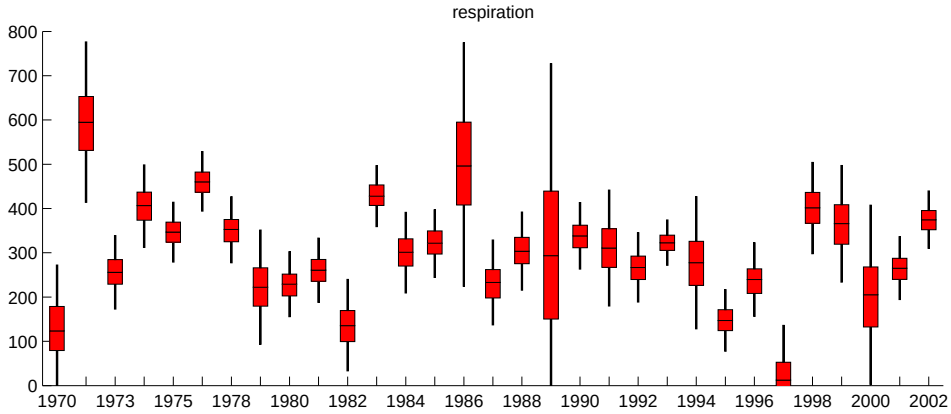


Figure 8: Average respiration [mg m^{-2} d^{-1}] calculated from the MCMC chain. Box plots correspond to the predictive posterior distribution of the estimated respiration.

using a LOWESS smoother [Cleveland, 1985]. The predictive analysis of the smoothed time series thus formed a non-parametric model for the trends with a predictive posterior distribution, see Figure 9.

Lastly, we studied this problem from the perspective of adaptive management, calculating the predictive distribution of the oxygen flux in the aerator needed to keep the amount of oxygen in the water above the level of 4 mg m^{-2} d^{-1}. We illustrated how this predictive posterior changes and becomes narrower as winter advances and more observations become available.

7.2 Phytoplankton growth and nutrition limitations

Paper 2 discusses a lake phytoplankton model in which the algal kinetics are modelled as functions of zooplankton and nutrients (P and N, phosphorus and nitrogen), and with controlling environmental variables such as temperature, light and in and out flows.

The model is in principle similar to the first mechanistic water quality models, which date back to 1970's. The new features here include the division of the phytoplankton into four functional groups, diatoms, chrysophyceae, cyanobacteria and a minor group, and the performing of a full Bayesian analysis on all the unknowns. The main interest in the modelling is focused on one algal group, nitrogen-fixing cyanobacteria, which have a severe effect on water quality, both recreational and commercial, when they bloom at the end of the summer. The building of water quality models with realistic estimates of uncertainties is required by the EU water framework directive.

The modelling procedure as a whole can be seen as a combination of empirical and mechanistic modelling. The overall behaviour is described in terms that have a concrete biological meaning. The food web used to describe the system is a meaningful concept that can be interpreted biologically. Not all of the details of the system can be modelled, however, due a to lack of available observations.

In the first stage of the modelling a more complex model with both algal and nutrital (N and P) state variables was considered. The dynamics of the nutrition were described in terms of nitrification from the bottom sediment and external loads. After an initial fitting of the model, it was realised that there were not enough data to estimate the internal parameters for the processes operating between the sediment and the water body, nor did there seemed to be agreement among specialists in the area about the mechanisms and external forces that should be taken into account in the modelling.

The next step was to simplify the model to such an extent that it would still be possible to get reason-

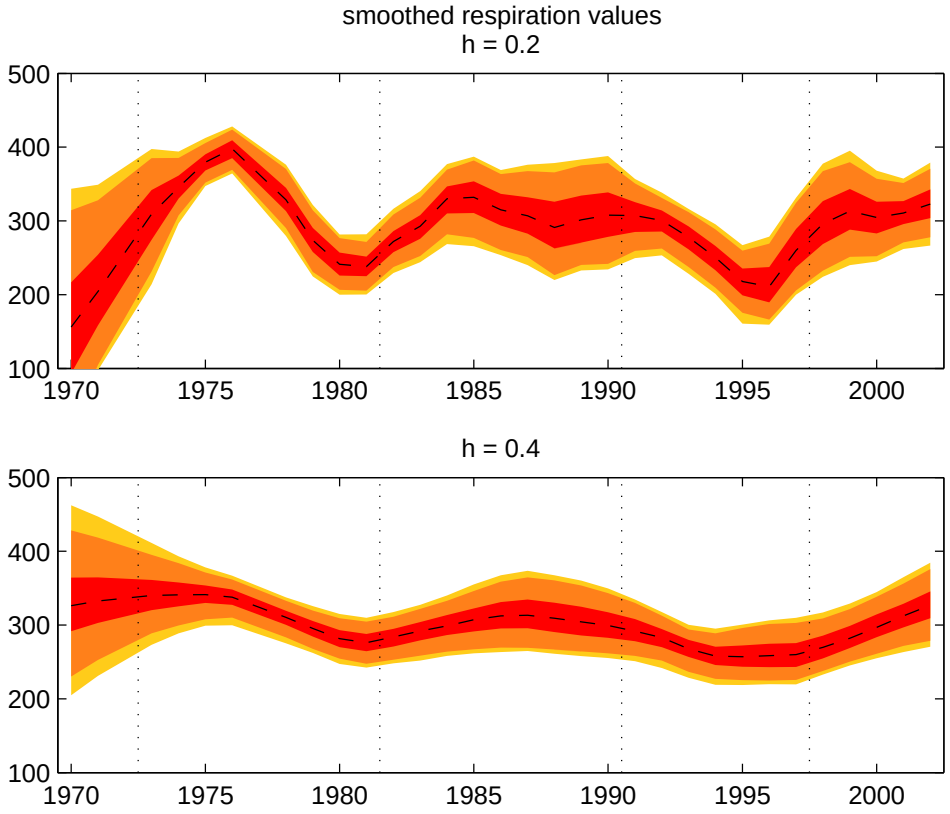


Figure 9: Smoothed respiration with two levels of smoothing. The upper plot with LOWESS parameter $h = 0.2$ corresponds approximately to a 6-year trend, and the lower one with $h = 0.4$ to a 12-year trend. The coloured levels give the 50%, 90% and 95% limits of the posterior.

able fits, with realistic predictions. The solution was to include N and P as control variables. Later, simulation-based scenarios were run with typical nutrient profiles. Zooplankton, Z, which is an important ingredient in any algal food web, was taken as an external control variable from the beginning, as the inclusion of its dynamics would have increased the need to include data on the amount of planktivorous fish. Lake Pyhäjärvi is a shallow, weakly stratified lake which can be handled with a model in which spatial variations are not taken into account. Thus we arrived at the model for the dynamics of the concentrations of the four algal groups C_{A_i}, $i = 1, \ldots, 4$, shown below.

$$\frac{dC_{A_i}}{dt} = (\mu_i - \rho_i - \frac{\sigma_i}{h} - \frac{Q}{V} - p_i C_{ZH})C_{A_i}, \qquad i = 1, 2, 3, 4 \tag{36}$$

with

$$\begin{aligned}
\mu_i &= \mu_i^* \theta_{\mu_i}{}^{T-T_{\text{ref}}} \frac{I}{K_{I_i}+I} \frac{P}{K_{P_i}+P} \frac{N}{K_{N_i}+N} \frac{K_{CS_i}}{K_{CS_i}+CS} && \text{growth} \\
\rho_i &= \rho_i^* \theta_{\rho_i}{}^{T-T_{\text{ref}}} && \text{dying} \\
\sigma_i &= \sigma_i^* (\text{a} - \text{b}\, T)^{T-T_{\text{ref}}} && \text{settling} \\
P &= P_{\text{tot}} - \textstyle\sum_{i=1}^{4} \alpha_i C_{A_i} && \text{phosphorus} \\
N &= N_{\text{tot}} - \textstyle\sum_{i=1}^{4} \beta_i C_{A_i} && \text{nitrogen}
\end{aligned}$$

The data consist of observations for the 4 algal groups over the period of 8 years and with 8 control variables, including temperature T, irradiation I, solids CS and nutrients N, P. More simplifications to the model were introduced in the model in Paper 2, e.g. the settling and dying of the algae were combined into a single non-predatory loss term by removing of the term ρ_i, a common parameter

value for the temperature dependence of the loss terms was set for all four groups, and the effect of suspended solids, CS, was not considered. Taking into account the unknown initial concentrations and four error variances for each algal group, we again had more than 60 parameters to estimate. [In fact, we have simplified the model still further in a more recent work [Haario et al., 2007].]

As the observations are derived from counts, it can be anticipated that direct Gaussian modelling for the error will not necessarily be appropriate. It was seen that the error level increases with the response, which is typical of count and Poisson-type data. A square root transformation stabilizes the error variance, and we can write the model as

$$y = (\sqrt{f_\theta} + \epsilon)^2, \quad \epsilon \sim N(0, I\sigma^2). \tag{37}$$

The sum of squares function needed for the MCMC is written in the form

$$SS(\theta) = \sum_{i=1}^{n} (\sqrt{y_i} - \sqrt{f(x_i; \theta)})^2. \tag{38}$$

Square root transformation also ensures that the model predictions for new observations cannot be negative. The sum of squares is calculated for all four algal groups and a different error variance is fitted for each group.

Important questions from a lake management perspective include the following. How does the nutrient loading affect the amount of cyanobacteria? What is the effect of fishing for plantivorous fish on the cyanobacteria, since this could leave room for larger concentrations of zooplankton? These questions were answered by the predictive analysis of simulations of different scenarios. The model described here is currently being studied further and applied to other lakes in ongoing work at the Finnish Environment Institute (SYKE) [Malve, 2007].

7.3 Ozone profile inversion from remote sensing data

The ozone layer has been an object of intense scientific study for decades now, especially since the discovery of the ozone hole over Antarctica in 1985. In 2002 the European Space Agency launched its ENVISAT satellite, which contains 10 instruments for monitoring the earth's environment and atmosphere. Among them was GOMOS (Global Ozone Monitoring of Occultation of Stars, [ESA 2002]) which studies ozone and other minor trace constituents in the atmosphere in a range from 10 to 100 km. GOMOS inversion methods are being actively developed at the Finnish Meteorological Institute (FMI), and one of the main motivations behind the Adaptive Metropolis (AM) methodology was to develop MCMC methods for GOMOS inversion and validation [Tamminen, 2004].

Below is a short description of the model, while the measurement principles are explained in Figure 10. The transmission spectrum of a star is given by

$$T(\lambda, \ell) = \frac{I(\lambda, \ell)}{I^{ref}(\lambda)}, \tag{39}$$

where $I(\lambda, \ell)$ is the stellar spectrum measured at tangent height ℓ and I^{ref} is a reference spectrum measured above the atmosphere. The transmission $T(\lambda, \ell)$ tells us how the stellar light is absorbed and scattered in the atmosphere, and without error it should have values between zero and one. The transmission at the wavelength λ along the ray path ℓ includes $T^{\text{abs}}_{\lambda,\ell}$ due to absorption and scattering by gases and $T^{\text{ref}}_{\lambda,\ell}$ due to refractive attenuation and scintillation. By gases we mean atmospheric constituents like ozone, NO_2, NO_3, but also neutral air density and aerosols. The absorption $T^{\text{abs}}_{\lambda,\ell}$ is

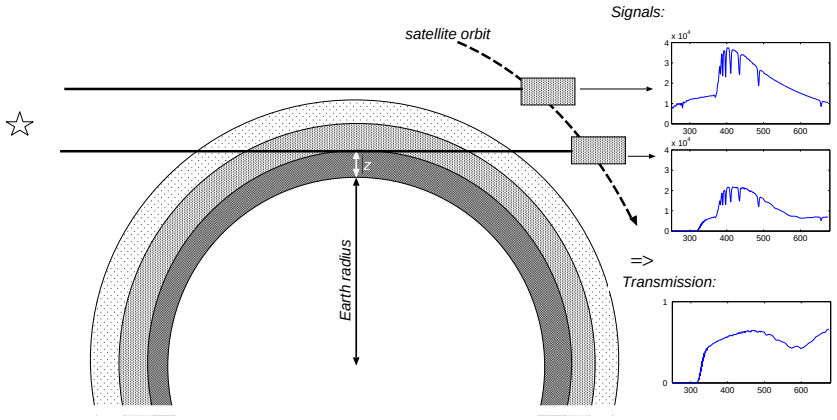


Figure 10: The principle of stellar occultation measurement. The reference spectrum is measured above the atmosphere and the attenuated spectrum through the atmosphere. By dividing the latter by the former we get the transmission spectrum. The tangential altitude of the measurement is denoted with z. The atmosphere is (locally) presented as spherical layers around the Earth. The thickness of the atmosphere is greatly overestimated in the figure. (Figure by Johanna Tamminen, FMI.)

proportional to the amount of gases in the atmosphere and follows Beer's law:

$$T^{\text{abs}}_{\lambda,\ell} = \exp\left[-\int_{\ell} \sum_{\text{gas}} \alpha^{\text{gas}}_{\lambda}(z(s))\rho^{\text{gas}}(z(s))\, ds \right], \tag{40}$$

where $\rho^{gas}(z)$ gives the gas density at altitude z and α denotes the cross sections. Each gas has typical wavelength ranges where the gas is active either by absorbing, scattering or emitting light. The cross sections reflect this behavior and their values are considered to be known from laboratory measurements. The inversion problem is to estimate the gas profiles ($\rho^{\text{gas}}(z)$) from the measurements

$$y_{\lambda,\ell} = T^{\text{abs}}_{\lambda,\ell} T^{\text{ref}}_{\lambda,\ell} + \epsilon_{\lambda,\ell}.$$

Although there is a high variability in the signal-to-noise ratio, the noise is approximately Gaussian, due to the large amount of dark charge, and also approximately uncorrelated between different altitudes and wavelengths. The scintillation and dilution term $T^{\text{ref}}_{\lambda,\ell}$ is estimated from separate measurements. Here we are working with so called Level 1 data on $T^{\text{abs}}_{\lambda,\ell}$, with the term $T^{\text{ref}}_{\lambda,\ell}$ removed.

In the operational algorithm the cross-sections are assumed to be constant on each ray path and the noise uncorrelated, in which case the inversion is separated into

$$T^{\text{abs}}_{\lambda,\ell} = \exp\left[-\sum_{\text{gas}} \alpha^{\text{gas}}_{\lambda,\ell} N^{\text{gas}}_{\ell} \right], \quad \lambda = \lambda_1, \ldots, \lambda_\Lambda, \tag{41}$$

with

$$N^{\text{gas}}_{\ell} = \int_{\ell} \rho^{\text{gas}}(z(s))\, ds, \quad \ell = \ell_1, \ldots, \ell_M. \tag{42}$$

The process of solving equation (41) for the line densities N^{gas}_{ℓ} is called *spectral inversion*, while *vertical inversion* is the problem of solving equation (42) for the local constituent densities $\rho^{\text{gas}}(z)$. By discretizing the atmosphere into layers and assuming constant (or linearly interpolated) gas densities inside the layers, the problem can be solved separately for each gas as a linear inversion problem

$$N^{\text{gas}}_{\ell} = A\rho^{\text{gas}}. \tag{43}$$

The matrix A contains the lengths of the line of sights in the layers and depends on the discretization. In the present operational retrieval algorithm the discretization is fixed so that the number of layers is the same as the number of measurement lines in each occultation.

Two ways of trying to overcome the problems in the operational algorithm are demonstrated in Paper 3: the one-step MCMC algorithm and a computationally very efficient parallel chains algorithm.

In the one-step version of the algorithm all the unknowns are estimated from the same MCMC run. This allows the temperature dependence of the cross-section to be taken into account, for example, in order to obtain a proper prior specification of the profiles and to explore the correlations between the heights. An adaptive single component Metropolis algorithm (SCAM, [Haario et al., 2005]) is used for the one-step solution in the paper, although an alternative could be the use of DRAM, a direction that has not yet been studied further.

In addition to the one-step method, a method is available that is similar to the operational algorithm in its speed and ease of implementation, called here the parallel chains algorithm. In this approach the line densities for each height are first fitted separately, or in a parallel manner, using MCMC, and the line density samples in the parallel chains are then combined for vertical inversion, which is performed for each set of line densities. This produces a chain for the actual gas densities. As vertical inversion is a linear operation, the latter part can be performed very efficiently. Also the parallel chains algorithm handles the nonlinear inversion part better than does the nonlinear least squares fitting in the operational algorithm, by using MCMC. It also makes it easy to apply smoothness priors and positivity constraints directly to the unknown profiles.

With dim stars, the signal-to-noise ratio is low, especially at low altitudes, making the solution unstable, as is typical of many inverse problems. The GOMOS solution immediately becomes unstable if the discretization of the profile has more points than there are measurements. The use of smoothness priors regularizes the problem. The smoothness properties are given in the article in such a way that they are easily interpretable and independent of the solution grid. This makes it easy to use "empirical" priors, i.e. priors that are based on independent observations, see Figure 11 for an example.

Aerosol cross-section model selection

In Paper 5 Bayesian model selection and averaging is applied to the GOMOS aerosol model selection problem. Although taken as known in the operational GOMOS algorithm, the aerosol cross-section is just an approximation of the underlying aerosol extinction process. It is typically modelled by using a function that behaves like $1/\lambda$, where λ is the wavelength. Several alternative formulations are possible, however, depending on the types of the aerosols at a given location. Four different aerosol cross-section models are used and an AARJ run is performed to study the effect of the aerosol model on the estimated values of the other constituents and to show how the uncertainty of the aerosol model can be included in the error estimates.

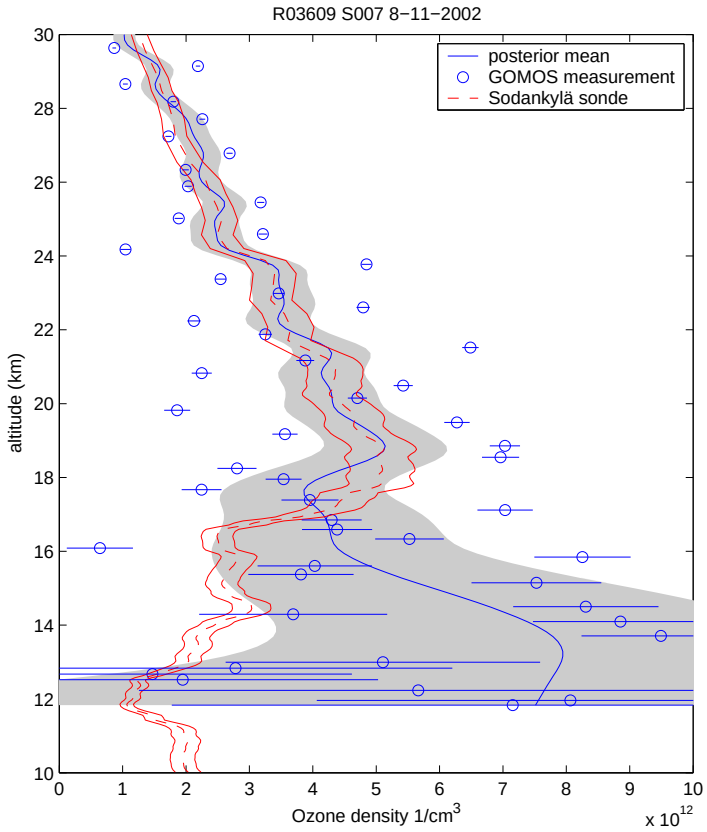


Figure 11: Smoothness priors for GOMOS. The original GOMOS ozone retrievals with error bars are shown by the blue circles. These are compared with the Sodankylä balloon sonde measurements at nearby locations and times (red lines). An empirical prior for the smoothness of the ozone profile is constructed from a history of sonde measurements, and this prior information is added to the GOMOS retrieval to form a new posterior estimate of the profile, given by the blue line, with 95% posterior limits given by the grey area.

8 Conclusions

Environmental models are often characterized by being over-parametrized and based on vague data, and specialists can seldom agree on all the details of their functional form. This leaves uncertainties regarding the conclusions that create a challenge for any statistician. Models must be judged by their predictive abilities. We should, of course, use the simplest acceptable model, but sometimes there are good reasons for preferring models that are in a way "too complicated".

Bayesian statistical inference provides a powerful tool for modelling in the environmental sciences. MCMC methods provide ways of performing the model calculations in a unified, practical and intuitively plausible manner. In contrast to classical statistical analysis, a Bayesian analysis provides a direct way to study all the uncertainties in modelling. A predictive analysis of the model is easily available after an MCMC run and the results of the analysis are more easily presented.

Adaptation is a key tool for building general purpose software for MCMC with a minimal need for case-dependent "hand tuning". This is demonstrated by the fact that all the applications described in this thesis were modelled using the same general software toolbox that implements the Metropolis-Hastings MCMC algorithm with a Gaussian proposal distribution and DRAM adaptation. The AM and DRAM algorithms can handle correlated parameters and even diagnose singular posterior covariance. Their use helps to identify potential problems in the model and facilitates the analysis of models with a large number of unknowns.

The DRAM algorithm introduced in this work has several advantages. It uses delayed rejection to help the adaptation to get started by providing accepted points right from the start of the MCMC simulation. The mixing of the chain is improved by combining proposals giving large global steps and smaller local steps. DR works in a theoretically correct manner, so that the ergodicity properties are preserved. DRAM provides proposal distributions that build upon simple multivariate Gaussian distributions but can handle non-Gaussian targets much more efficiently than when using a single Gaussian proposal. It is relatively easy to implement and may be employed in applications of many different kinds.

The AARJ algorithm is an easy to use method for model selection and model averaging. By using AM, DRAM and the automatic RJMCMC as the building blocks it allows for implementation of a general tool that further expands the scope of MCMC methods in modelling applications.

MCMC is really an excellent tool for a modeller, and adaptive methods make its use easier for a wider audience. Statistics in general is going through very interesting times as MCMC methods are widening the applicability of statistical analysis. At the same time, MCMC has shifted the paradigm from classical statistics towards the Bayesian approach. It is hoped that the accompanying articles will provide further evidence of the strength of MCMC in general and of adaptive MCMC in particular.

References

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

Yonathan Bard. *Nonlinear Parameter Estimation*. Academic Press, New York, 1974.

José M. Bernardo and Adrian F. M. Smith. *Bayesian Theory*. Wiley, 2000.

G.L. Bowie, W.B. Mills, D.B. Porcella, C.L. Campbell, J.R. Pagenkopf, G.L. Rupp, K.M. Johnson, P.W.H. Chan, S.A. Gherini, and C.E. Chamberlin. Rates, constants, and kinetic formulations in surface water modeling. Technical Report EPA/600/3-85/040, U.S. Environmental Agency, ORD, Athens, GA, ERL, 1985.

George E. P. Box and George C. Tiao. *Bayesian Inference in Statistical Analysis*. Addison-Wesley, 1973.

William S. Cleveland. *The Elements of Graphing Data*. Wadsworth, Monterrey, California, 1985.

Petros Dellaportas, Jonathan J. Forster, and Ioannis Ntzoufras. On Bayesian model and variable selection using MCMC. *Statistics and Computing*, 12:27–36, 2002.

J. J. Dongarra, C. B. Moler, J. R. Bunch, and G. W. Stewart. *LINPACK Users' Guide*. SIAM, Philadelphia, 1979.

ESA 2002. *EnviSat GOMOS Product Handbook*. European Space Agency, December 2002. URL `http://envisat.esa.int/dataproducts/`. Issue 1.1.

Dani Gamerman. *Markov Chain Monte Carlo – Stochastic simulation for Bayesian inference*. Chapman & Hall, 1997.

A. Gelman, G. O. Roberts, and W. R. Gilks. Efficient Metropolis jumping rules. *Bayesian Statistics*, 5, 599–607 1996.

Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman & Hall, 1995.

Peter J. Green. Trans-dimensional Markov chain Monte Carlo. In Peter J. Green, Nils Lid Hjort, and Sylvia Richardson, editors, *Highly Structured Stochastic Systems*, number 27 in Oxford Statistical Science Series. Oxford University Press, 2003.

Peter J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

Heikki Haario. *MODEST User Guide*. ProfMath Oy, 1995.

Heikki Haario, Eero Saksman, and Johanna Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242, 2001.

Heikki Haario, Eero Saksman, and Johanna Tamminen. Componentwise adaptation for high dimensional MCMC. *Computational Statistics*, 20(2):265–274, 2005.

Heikki Haario, Leonid Kalachev, and Marko Laine. Reduced models for algae growth. Submitted to *Bulletin of Mathematical Biology*, in revision, 2007.

David Hastie. *Towards Automatic Reversible Jump Markov Chain Monte Carlo*. PhD thesis, University of Bristol Department of Mathematics, 2005.

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

Sir Harold Jeffreys. *Theory of Probability*. Oxford University Press, third edition, 1961.

Jari P. Kaipio and Erkki Somersalo. *Computational and Statistical Methods for Inverse Problems*. Springer, 2004.

Marko Laine. *MCMCSTAT Toolbox documentation*, 2007. URL `http://www.helsinki.fi/~mjlaine/mcmc/`.

D.J. Lunn, A. Thomas, N. Best, and D. Spiegelhalter. WinBUGS – a Bayesian modelling framework: concepts, structure, and extensibility. *Statistics and Computing*, 10:325–337, 2000.

Olli Malve. *Water Quality Prediction for River Basin Management*. PhD thesis, Helsinki University of Technology, Department of Civil and Environmental Engineering, Water Resources Laboratory, 2007.

Andrew D. Martin and Kevin M. Quinn. Applied Bayesian inference in R using MCMCpack. *R News*, 6(1):2–7, 2006. URL `http://cran.r-project.org/doc/Rnews`.

Using MATLAB. The MathWorks, Inc., 2000. Version 6.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

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

Esa Nummelin. MC's for MCMC'ists. *International Statistical Review*, 70(2):215–240, 2002.

Karl R. Popper. *Conjectures and Refutations – The Growth of Scientific Knowledge*. Routledge, fifth edition, 1989.

Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, second edition, 2005.

G.O. Roberts and J.S. Rosenthal. Coupling and ergodicity of adaptive MCMC. *Journal of Applied Probability*, 44(2):458–475, 2007.

G. A. F. Seber and C. J. Wild. *Nonlinear Regression*. Wiley, 1989.

B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, 1986.

A. D. Sokal. Monte Carlo methods in statistical mechanics: foundations and new algorithms. Lecture notes, 1996. Cours de Troisième Cycle de la Physique en Suisse Romande.

David J. Spiegelhalter, Nicola G. Best, Bradley P. Carlin, and Angelika van der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society, Series B*, 64(4): 583–639, 2002.

Johanna Tamminen. *Adaptive Markov chain Monte Carlo algorithms with geophysical applications*. PhD thesis, University of Helsinki Department of Mathematics and Statistics, 2004.

Albert Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Siam, Philadelphia, 2005.

Albert Tarantola. Popper, Bayes and the inverse problem. *Nature Physics*, 2:492–494, August 2006.

Andrew Thomas, Bob O Hara, Uwe Ligges, and Sibylle Sturtz. Making BUGS open. *R News*, 6(1): 12–17, 2006. URL `http://cran.r-project.org/doc/Rnews`.

Luke Tierney. Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4): 1701–1728, 1994.

Luke Tierney. A note on Metropolis-Hastings kernels for general state spaces. *The Annals of Applied Probability*, 8(1):1–9, 1998.

David Williams. *Weighting the Odds – A Course in Probability and Statistics*. Cambridge University Press, 2001.

A MCMC toolbox for Matlab

A.1 Introduction

For the MCMC to become an everyday tool for scientists, software must be developed. There are several examples already, such as BUGS [Lunn et al., 2000, Thomas et al., 2006] and MCMCpack [Martin and Quinn, 2006]. The MCMC procedures described in this thesis are implemented as a set of Matlab files. The code is available from `http://www.helsinki.fi/~mjlaine/mcmc/`.

To use the code, the user must submit her own Matlab code to calculate either the model or the sum-of-squares (that is $-2\log(\text{likelihood})$) function given the parameters and the "y" and "x" data sets. The MCMC code does not depend on any extra Toolboxes, only the base Matlab software is needed. The code implements the DRAM method of adaptation. This includes as special cases the plain Metropolis-Hastings, the Delayed Rejection, and the Adaptive Metropolis algorithms described in the thesis. Below follows a short introduction and some examples.

A.2 MCMC functions

The main Matlab subroutine is the function `mcmcrun`. Here is the help text:

```
MCMCRUN Metropolis-Hastings MCMC simulation for nonlinear Gaussian models
 properties:
  multiple y-columns, sigma2-sampling, adaptation,
  Gaussian prior, parameter limits, delayed rejection, dram

 [RESULTS,CHAIN,S2CHAIN,SSCHAIN] = MCMCRUN(MODEL,DATA,PARAMS,OPTIONS)
 MODEL   model options structure
    model.ssfun    -2*log(likelihood) function
    model.sigma2   initial error variance
    model.N        total number of observations
    model.S20      prior for sigma2
    model.N0       prior accuracy for S20
    model.nbatch   number of datasets

     sum-of-squares function 'model.ssfun' is called as
     ss = ssfun(par,data) or
     ss = ssfun(par,data,local)
     instead of ssfun, you can use model.modelfun as
     ymodel = modelfun(data{ibatch},theta_local)

 DATA    the data, passed directly to ssfun

 PARAMS  theta structure
   {  {'par1',initial, min, max, pri_mu, pri_sig, targetflag, localflag}
      {'par2',initial, min, max, pri_mu, pri_sig, targetflag, localflag}
      ... }

 OPTIONS mcmc run options
    options.nsimu            number of simulations
    options.updatesigma      update error variance (=1)
    options.qcov             proposal covariance
    options.method           'dram','am','dr' or 'mh'
    options.verbosity        level of information printed
    options.waitbar          use graphical waitbar?
    options.burnintime       burn in before adaptation starts
    options.adaptint         interval for adaptation
```

```
Output:
 RESULTS   structure that contains results and information about
           the simulations
 CHAIN, S2CHAIN, SSCHAIN
          parameter, sigma2 and sum-of-squares chains
```

Other basic function included is `mcmcplot` that produces some standard plots of the chain, such as one and two dimensional chain scatter plots, histograms and kernel density estimates of the posteriors.

A.3 Examples

The code is best illustrated by some examples.

A.3.1 Monod model

This example is from P. M. Berthouex and L. C. Brown: *Statistics for Environmental Engineers*, CRC Press, 2002. We fit the Monod model

$$y = \theta_1 \frac{t}{\theta_2 + t} + \epsilon \quad \epsilon \sim N(0, I\sigma^2)$$

to observations

```
x (mg / L COD):  28    55    83    110   138   225   375
y (1 / h):       0.053 0.060 0.112 0.105 0.099 0.122 0.125
```

First clear some variables from possible previous runs.

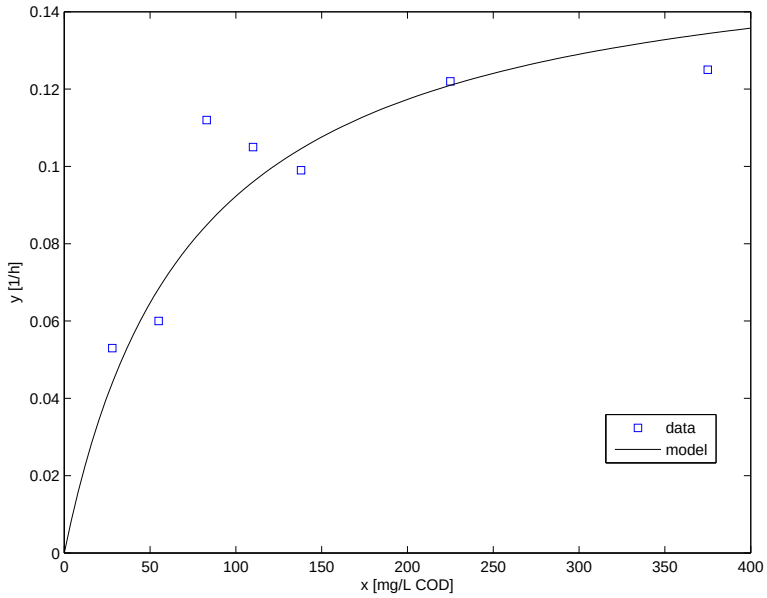
```
clear data model options
```

Next, create a data structure for the observations and control variables. Typically one could make a structure `data` that contains fields `xdata` and `ydata`.

```
data.xdata = [28    55    83    110   138   225   375]';   % x (mg / L COD)
data.ydata = [0.053 0.060 0.112 0.105 0.099 0.122 0.125]'; % y (1 / h)
```

Here is a plot of the data set.

```
figure(1); clf
plot(data.xdata,data.ydata,'s');
xlim([0 400]); xlabel('x [mg/L COD]'); ylabel('y [1/h]');
```



For the MCMC run we need the sum of squares function. For the plots we shall also need a function that returns the model. Both the model and the sum of squares functions are easy to write as one line anonymous functions using the @ construct.

```
modelfun = @(x,theta) theta(1)*x./(theta(2)+x);
ssfun    = @(theta,data) sum((data.ydata-modelfun(data.xdata,theta)).^2);
```

We have to define three structures for inputs for the `mcmcrun` function, parameter, model, and options. Parameter structure has a special form and it is constructed as Matlab cell array with curly brackets. Minimal structure has the name of the parameter and the initial value of it.

```
params = {
    {'theta1', 0.17}
    {'theta2', 100}
    };
```

In general, each parameter line can have up to 7 elements: 'name', initial_value, min_value, max_value, pri_mu, pri_sigma, and targetflag.

The `model` structure holds the information about the model. Minimally we need to set `ssfun` for the sum of squares function and the initial estimate of the error variance `sigma2`.

```
model.ssfun  = ssfun;
model.sigma2 = 0.01^2;
```

The `options` structure has settings for the MCMC run. We need at least the number of simulations in `nsimu`. Here we also set the option `updatesigma` to allow automatic sampling and estimation of the error variance.

```
options.nsimu = 4000;
options.updatesigma = 1;
```

The actual MCMC simulation run is done using the function `mcmcrun`.

```
[res,chain,s2chain] = mcmcrun(model,data,params,options);
```

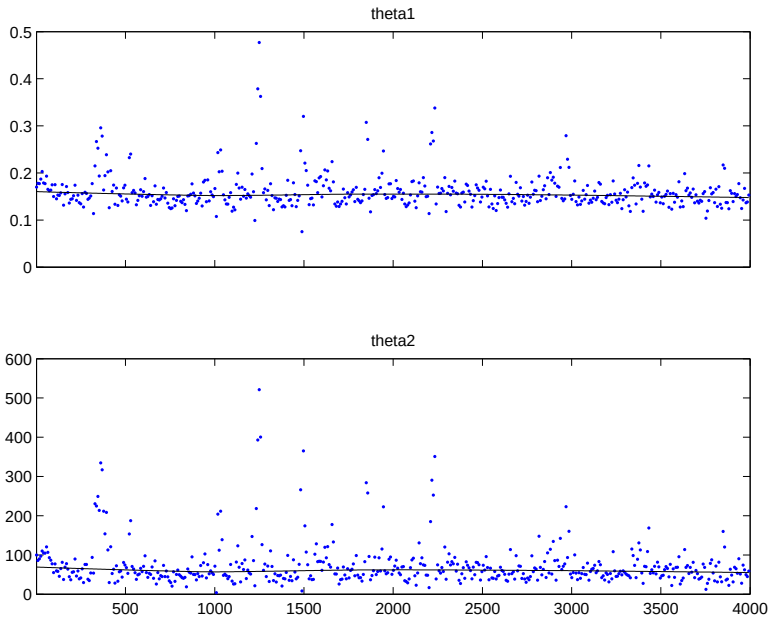
```
  Sampling these parameters:
  name    start  [min,max]  N(mu,s^2)
```

```
theta1: 0.17 [-Inf,Inf] N(0,Inf^2)
theta2: 100 [-Inf,Inf] N(0,Inf^2)
```

After the run the we have a structure `res` that contains some information about the run, and a matrix outputs `chain` and `s2chain` that contain the actual MCMC chains for the parameters and for the observation error variance.

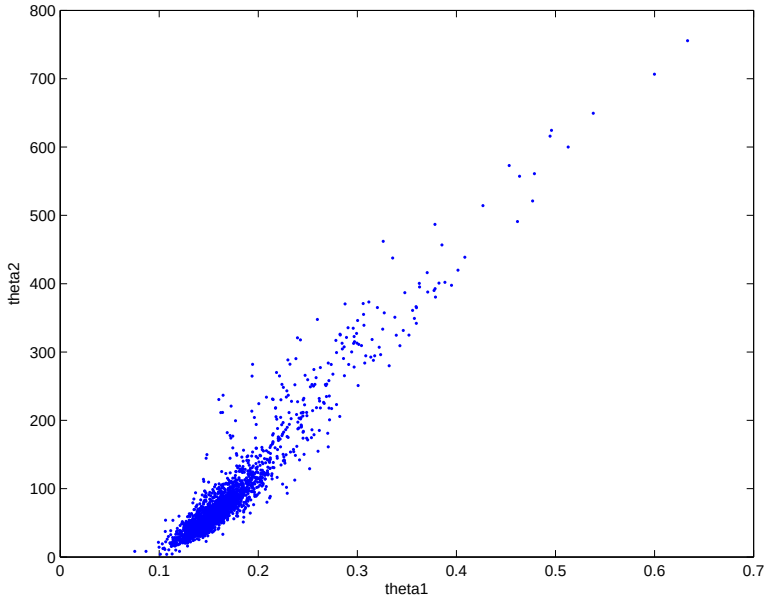
The `chain` variable is `nsimu×npar` matrix and it can be plotted and manipulated with standard Matlab functions. Function `mcmcplot` can be used to make some useful chain plots and also plot 1 and 2 dimensional marginal kernel density estimates of the posterior distributions.

```
figure(2); clf
mcmcplot(chain,[],res,'chainpanel');
```



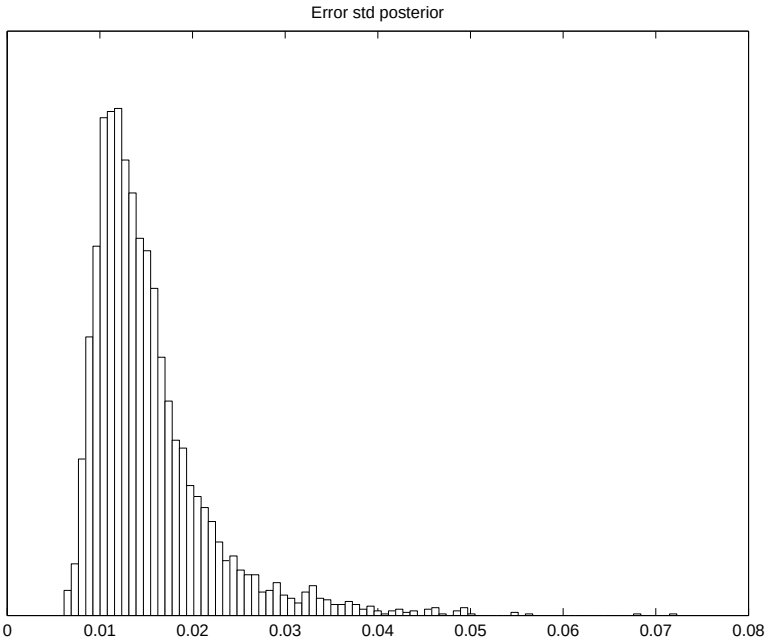
The `'pairs'` options makes pairwise scatter plots of the columns of the `chain`.

```
figure(3); clf
mcmcplot(chain,[],res,'pairs');
```



If we take square root of the `s2chain` we get the chain for error standard deviation. Here we use `'hist'` option for the histogram of the chain.

```
figure(4); clf
mcmcplot(sqrt(s2chain),[],[],'hist')
title('Error std posterior')
```



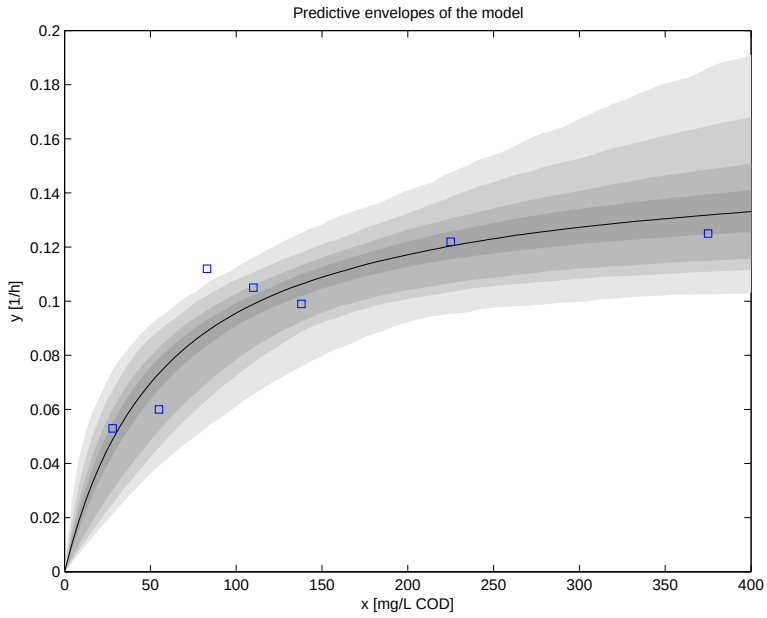
A point estimate estimate of the model parameters can be calculated from the mean of the `chain`. Here we plot the fitted model over the data set.

```
x = linspace(0,400)';
figure(1)
hold on
plot(x,modelfun(x,mean(chain)),'-k')
hold off
```

```
legend('data','model',0)
```

Instead of just a point estimate of the fit, we should also study the predictive posterior distribution of the model. The `mcmcpred` and `mcmcpredplot` functions can be used for this purpose. By them we calculate the model fit for a randomly selected subset of the chain and calculate the predictive envelope of the model. The cray areas in the plot correspond to 50%, 90%, 95%, and 99% posterior regions.

```
figure(5); clf
out = mcmcpred(res,chain,[],x,modelfun);
mcmcpredplot(out);
hold on
plot(data.xdata,data.ydata,'s'); % add data points to the plot
xlabel('x [mg/L COD]'); ylabel('y [1/h]');
hold off
title('Predictive envelopes of the model')
```



A.3.2 Gaussian target distribution

In this example, we generate Gaussian target with known covariance matrix. The target distribution has a known form and could be calculated explicitly, so this simulation is mainly for testing of the algorithms.

```
clear model data params options;

nsimu = 4000;  % number of simulations
npar  = 4;     % dimension of the parameter

data.x0 = zeros(1,npar); % mean vector
```

We create the parameter structure in a loop:

```
for i=1:npar
  params{i} = {sprintf('\\theta_{%d}',i),data.x0(i),-Inf,Inf,NaN,Inf,1};
end
```

Create covariance and precision matrises.

```
[Sig,Lam] = covcond(100,ones(npar,1));
```

Store the precision matrix in `data` so we can use it in the `ssfun`.

```
data.Lam = Lam;
```

The `ssfun` for `mcmcrun` is the quadratic form in the Gaussian distribution.

```
model.ssfun     = @(x,data) (x-data.x0)*data.Lam*(x-data.x0)';
model.N         = 1;
```

For `mcmcrun` we use scaled versions of the known target covariance as the proposal covariance. This is known to be the optimal proposal.

```
options.nsimu   = nsimu;
options.qcov    = 2.4^2/npar*Sig;
options.method  = 'dram';  % use the (default) DRAM method

[results,chain] = mcmcrun(model,data,params,options);
```

```
  Setting nbatch to 1
  Sampling these parameters:
  name   start [min,max] N(mu,s^2)
  \theta_{1}: 0 [-Inf,Inf] N(0,Inf^2)
  \theta_{2}: 0 [-Inf,Inf] N(0,Inf^2)
  \theta_{3}: 0 [-Inf,Inf] N(0,Inf^2)
  \theta_{4}: 0 [-Inf,Inf] N(0,Inf^2)
```

From the generated chain we calculate the relative distances of the chain points from the origin and count the points that are inside given probability limits. We plot the first two dimensions of the chain together with the correct probability contours.

The title of the 2d plot shows the rejection rate and the proportion of points inside the ellipsoids. Number $\tau * t$ in the title tells how many seconds it takes to generate 1000 independent samples according to the integrated autocorrelation time (`iact`) estimate.

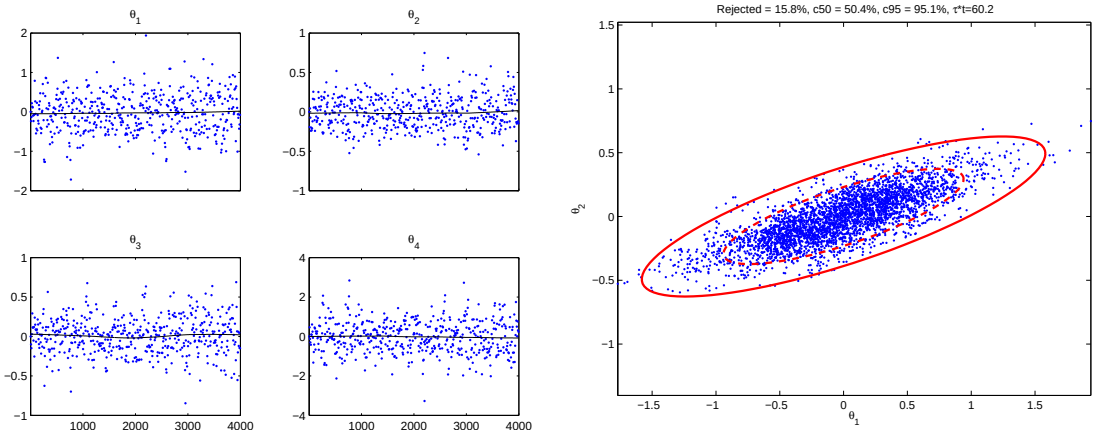
```
d    = mahalanobis(chain(:,1:npar),data.x0,Lam,1);
c50  = chiqf(0.50,npar);
c95  = chiqf(0.95,npar);
cc50 = sum(d<c50)./nsimu;
cc95 = sum(d<c95)./nsimu;

figure(1); clf
mcmcplot(chain,[1:4],results.names,'chainpanel')
figure(2); clf
mcmcplot(chain,[1,2],results.names,'pairs',0)

title(sprintf('Rejected = %.1f%%, c50 = %.1f%%, c95 = %.1f%%, \\tau*t=%.1f', ..
              results.rejected*100, cc50*100, cc95*100, ...
              results.simutime/results.nsimu*1000*mean(iact(chain))))

hold on
ellipse(data.x0(1:2),c50*Sig(1:2,1:2),'r--','LineWidth',2);
```

```
ellipse(data.x0(1:2),c95*Sig(1:2,1:2),'r-','LineWidth',2);
axis equal
hold off
```



A.4 Computational details

A.4.1 Recursive formulas for mean, covariance and the Cholesky factor

For the covariance matrix C and the mean vector $\overline{x}$ we have recursive formulas for adding a new observation x. Below w_{old} is the old sum of weights and w is the weight for the new observation x.

$$C_{\text{new}} = \frac{w_{\text{old}} - 1}{w + w_{\text{old}} - 1} C_{\text{old}} + \frac{w\ w_{\text{old}}}{(w + w_{\text{old}} - 1)(w + w_{\text{old}})} (x - \overline{x}_{\text{old}})'(x - \overline{x}_{\text{old}}). \tag{44}$$

For the mean vector we have

$$\overline{x}_{\text{new}} = \overline{x}_{\text{old}} + \frac{w}{w + w_{\text{old}}} (x - \overline{x}_{\text{old}}). \tag{45}$$

For updating the Cholesky factor R of a covariance matrix, we use a routine for rank 1 Cholesky update, for example the routine `DCHUD` in the LINPACK Fortran subroutine library [Dongarra et al., 1979], or `cholupdate` in Matlab. We get the Cholesky factor R_{new} of C_{new} when we update

$$\sqrt{\frac{w_{\text{old}} - 1}{w + w_{\text{old}} - 1}} R_{\text{old}}$$

with

$$\sqrt{\frac{w\ w_{\text{old}}}{(w + w_{\text{old}} - 1)(w + w_{\text{old}})}} (x - \overline{x}_{\text{old}})'.$$

The Cholesky update is $O(p^2)$ operation while calculating the Cholesky from the start is $O(p^3)$, where p is the dimension of the parameter vector. So if we update more seldom that at every p'th iteration, it is computationally wise to recalculate the Cholesky factor and not to use the update formula.

```
function [xcov,xmean,wsum,R]=covupd(x,w,oldcov,oldmean,oldwsum,oldR)
%COVUPD covariance update
% [xcov,xmean,wsum,R]=covupd(x,w,oldcov,oldmean,oldwsum,oldR)
% optionally updates also the Cholesky factor R
```

```
% Marko Laine <Marko.Laine@Helsinki.FI>
% $Revision: 1.3 $  $Date: 2006/09/06 09:15:16 $

[n,p]=size(x);
if n == 0 % nothing to update with
  xcov = oldcov; xmean = oldmean; wsum = oldwsum;
  return
end

if nargin<2 | isempty(w)
  w = 1;
end
if length(w) == 1
  w = ones(n,1)*w;
end

if nargin < 6 | isempty(oldR)
  R = [];
else
  R = oldR;
end

if nargin>2 & ~isempty(oldcov) % update

  for i=1:n
    xi     = x(i,:);
    wsum   = w(i);
    xmeann = xi;
    xmean  = oldmean + wsum/(wsum+oldwsum)*(xmeann-oldmean);

    if ~isempty(R)
      R = cholupdate(sqrt((oldwsum-1)/(wsum+oldwsum-1))*R, ...
                     (xi-oldmean)'* ...
                     sqrt((wsum*oldwsum)/(wsum+oldwsum-1)/(wsum+oldwsum)));
    end

    xcov =  (oldwsum-1)./(wsum+oldwsum-1).*oldcov + ...
            wsum.*oldwsum/(wsum+oldwsum-1)./(wsum+oldwsum) .* ...
             ((xi-oldmean)' *(xi-oldmean));
    wsum    = wsum+oldwsum;
    oldcov  = xcov;
    oldmean = xmean;
    oldwsum = wsum;
  end

else % no update

  wsum  = sum(w);
  xmean = zeros(1,p);
  xcov  = zeros(p,p);
  for i=1:p
    xmean(i) = sum(x(:,i).*w)./wsum;
  end
  if wsum>1
    for i=1:p
      for j=1:i
        xcov(i,j) = (x(:,i)-xmean(i))' * ((x(:,j)-xmean(j)).*w)./(wsum-1);
        if (i ~= j)
          xcov(j,i) = xcov(i,j);
        end
```

```
      end
    end
  end

  if nargout>3
    [R,p] = chol(xcov);
    if p~=0
      R=[];
    end
  end

end
```

A.4.2 DRAM

The DR method can be coded directly as given in the formulas for the acceptance probabilities. This is feasible if we just want to one extra try after a rejection. For several tires a recursive function can be build. In the code this is implemented as three auxiliary functions, `alphafun` for the recursive calculations of the acceptance probabilities, `qfun` for the needed proposal ratios, and `lfun` for the logarithm of the posterior ratios. Below is a listing of function `alphafun`.

```
function y=alphafun(varargin)
% alphafun(x,y1,y2,y3,...)
% recursive acceptance function for delayed rejection
% x.p, y1.p, ... contain the parameter value
% x.ss, y1.ss, ... the sum of squares
% x.a, y1.a, ... past alpha probabilities

stage = nargin - 1; % The stage we're in, elements in varargin - 1

% recursively compute past alphas
a1 = 1; a2 = 1;
for k=1:stage-1
  a1 = a1*(1-varargin{k+1}.a); % already have these alphas
  a2 = a2*(1-alphafun(varargin{(stage+1):-1:(stage+1-k)}));
end

y = lfun(varargin{1},varargin{end}); % log posterior ratio
for k=1:stage
  y = y + qfun(k,varargin{:}); % log proposal ratios
end

y = min(1, exp(y)*a2/a1);
```

For the proposal ratios, and in case of Gaussian proposals, we need to calculate ratios of type

$$\frac{q_1(\theta^{**},\theta^{*})}{q_1(\theta,\theta^{*})} = \exp\left\{-\frac{1}{2}\left(\left\|\theta^{**}-\theta^{*}\right\|_2 R_1^{-1}\right)^2 + \frac{1}{2}\left(\left\|\theta-\theta^{*}\right\|_2 R_1^{-1}\right)^2\right\}. \tag{46}$$

This is implemented in function `qfub`:

```
function z=qfun(iq,varargin)
% Gaussian n:th stage log proposal ratio
% log of q_i(y_n,..,y_n-j) / q_i(x,y_1,...,y_j)

global invR

stage = nargin-1-1;
if stage == iq
```

```
  z = 0;                                % we are symmetric
else
  iR = invR{iq};                        % proposal^(-1/2)
  y1 = varargin{1}.p;           % y1
  y2 = varargin{iq+1}.p;        % y_i
  y3 = varargin{stage+1}.p;     % y_n
  y4 = varargin{stage-iq+1}.p;  % y_(n-i)
  z = -0.5*(norm((y4-y3)*iR)^2-norm((y2-y1)*iR)^2);
end
```

Finally, for the logarithm of the posterior ratios we have Matlab function `lfun`:

```
function z=lfun(x,y)
% calculates log posterior ratio from sum-of-squares
z = -0.5*( sum((y.ss-x.ss)./x.s2) + y.pri - x.pri );
```

Now the simulation loop for the DRAM can be written as:

```
for i=2:nsimu

  y.p  = x.p+randn(1,npar)*R{1}; % proposal, random walk
  y.ss = feval(ssfun,y.p);

  acce = 0;
  itry = 1;
  xyz = {x,y}; % current try path

  while acce == 0 & itry <= Ntry
    alpha = alphafun(xyz{:});
    xyz{end}.a  = alpha; % save alpha
    if rand(1,1) < alpha     %  accept
      x    = y;
      acce = itry;
    elseif itry < Ntry % try a new one
      y.p  = x.p + randn(1,npar) * R{itry+1}; % proposal
      y.ss = feval(ssfun,y.p);
      xyz  = {xyz{:},y};
    end
    itry = itry+1;
  end

  chain(i,1:npar) = x.p;

  %% adaptation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
     adapt Cholesky factor R{i} here
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

end % of nsimu
```