Open your mind. LUT.
Lappeenranta University of Technology

LAPPEENRANTA UNIVERSITY OF TECHNOLOGY
Faculty of Technology
Department of Mathematics and Physics

MCMC Analysis for Optimization of Stochastic Models.

The topic of this Master's thesis was approved by the departmental council of the Department of Mathematics and Physics on 30 August, 2011.

The examiners of the thesis were Professor Heikki Haario and PhD Tuomo Kauranne. The thesis was supervised by Professor Heikki Haario.

In Lappeenranta November 1, 2011

Maombi Mkenyeleye
Punkkerikatu 2 A 6
53850 Lappeenranta
Phone: +358465963672
Maombi.Mkenyeleye@lut.fi

Abstract

Lappeenranta University of Technology
Department of Mathematics and Physics

Maombi Mkenyeleye

MCMC Analysis for Optimization of Stochastic Models
Master's thesis

2011

48 pages, 22 figures, 4 tables and 1 appendix.

Key words: Bayesian Inference, Markov chain Monte Carlo (MCMC), Stochastic Models, Optimization, Chemical Kinetics.

In any decision making under uncertainties, the goal is mostly to minimize the expected cost. The minimization of cost under uncertainties is usually done by optimization. For simple models, the optimization can easily be done using deterministic methods. However, many models practically contain some complex and varying parameters that can not easily be taken into account using usual deterministic methods of optimization. Thus, it is very important to look for other methods that can be used to get insight into such models.

MCMC method is one of the practical methods that can be used for optimization of stochastic models under uncertainty. This method is based on simulation that provides a general methodology which can be applied in nonlinear and non-Gaussian state models. MCMC method is very important for practical applications because it is a unified estimation procedure which simultaneously estimates both parameters and state variables. MCMC computes the distribution of the state variables and parameters of the given data measurements. MCMC method is faster in terms of computing time when compared to other optimization methods.

This thesis discusses the use of Markov chain Monte Carlo (MCMC) methods for optimization of Stochastic models under uncertainties .The thesis begins with a short discussion about Bayesian Inference, MCMC and Stochastic optimization methods. Then an example is given of how MCMC can be applied for maximizing production at a minimum cost in a chemical reaction process. It is observed that this method performs better in optimizing the given cost function with a very high certainty.

Acknowledgement

I wish to thank the Department of Mathematics and Physics of Lappeenranta University of Technology (LUT) for the scholarship granted to me for the whole time of my studies here at LUT. Without this scholarship I could not by any other means be able to join this innovative University.

Thanks to Professor, Ph.D. Matti Heiliö, the Coordinator of Technomathematics Program for his initiatives and collaboration with the University of Dar Es Salaam, Tanzania and other East African Universities.

I am very grateful to my supervisor and Head of Mathematics and Physics Department at LUT, Professor, Ph.D. Heikki Haario for his close and continuous supervision for this work to be completed. My gratitude also goes to Ph.D. Tuomo Kauranne, for all his assistance to me during my studies and for examining this work.

My sincere gratitude goes to Isambi Sailon, my friend, for his moral, material and practical support, and his encouragement throughout the time of my being in Lappeenranta and for this work to be completed. He has always been giving me valuable and constructive ideas about how to go in completing this work. I would also like to thank Antti Solonen for his assistance in the practical work.

I thank my wife Neema Mkenyeleye, my son Brighton Mkenyeleye, and my mother Esther Balosha ('*nawapenda sana*') for their patience and tolerance during the whole time of my absence and for their prayers. Lastly but not least, I thank all my friends and classmates in Lappeenranta.

May God bless you all.

Lappeenranta; November 1, 2011.

Maombi Mkenyeleye

Contents

1 Introduction **1**

1.1 Structure of the thesis 2

2 Theoretical background **3**

2.1 Bayesian Inference 3

2.2 Monte Carlo Method and Markov chain Overview 4

2.2.1 Monte Carlo Integration 5

2.2.2 Markov Chains 5

2.2.3 Markov Chain Monte Carlo Methods (MCMC) 6

2.2.4 The Metropolis Algorithm 7

2.2.5 The Metropolis-Hastings Algorithm (MH) 10

2.2.6 The Gibbs Sampler 13

2.2.7 Adaptive Metropolis (AM) 14

2.2.8 Delayed Rejection (DR) 15

2.2.9 Delayed Rejection Adaptive Metropolis (DRAM) 16

2.3 Convergence Diagnostics and Chain Length 17

3 Stochastic Optimization **26**

3.1 Stochastic Optimization Problems 27

3.2 Problem Formulation 27

3.3 Methods of Optimization 28

3.3.1 Mean Criterion 28

3.3.2 Sample Average Approximation (SAA) 28

3.3.3 Risk Models 29

3.3.4 Worst Case Criterion 30
3.3.5 Response Surface Method 30
3.3.6 Process Optimization 31

4 Numerical Results **32**

4.1 Chemical Reaction 32
4.1.1 Parameter Estimation 33
4.1.2 Statistical Analysis 35
4.2 Optimization 41

5 Conclusions **45**

References **47**

List of Tables

1 Mean and Variance for α and η 13

2 Data Measurements 17

3 Response surface technique 31

4 MCMC Parameter values 38

List of Figures

1 The illustration of Bayesian inference; Prior distribution, likelihood and posterior distribution computed from Gaussian distribution. 4

2 Theoretical density and Posterior distribution of cauchy distribution obtained by Metropolis algorithm 9

3 Sequence of samples 10

4 Histograms for the posteriors of α and η. The left panel is a Histogram for posterior of α while the right panel in a Histogram for the posterior of η 12

5 The posterior predictive distribution for the components A and B in our example model 20

6 The Autocorrelation Function for the parameters k_1 and k_2. The left panel if the The Autocorrelation Function for k_1 and the right panel is the Autocorrelation Function both with lags 30 21

7 Trace Plots for the parameters k_1 and k_2. 23

8 Sampled parameter values plotted pairwise for the parameters k_1 and k_2 in our example model. 24

9 A scatter plot of a parameter pair with confidence regions based on kernel densities k_1 and k_2 in our example model. 25

10 Reactants A and B are fed in the pipe of length L to produce C. 32

11 Solution of the ODE system at different Temperatures. The dots in circle form are the measurements. 33

12 Solution at two temperatures with the generated data 34

13 Sample Autocorrelation Function for the Parameters 35

14 Trace Plots of Parameters: The trace plots for both K_{ref} and E show relatively good mixing as the values are moving like a snake around the mean, though that of E starts to show the mixing after a short time. Generally, we can conclude that the mixing of the parameters was relatively good. 36

15 Scatter Plot of the parameters: From this plot, we observe that K_{ref} and E are positively correlated 37

16 A scatter plot of a parameter pair with confidence regions based on kernel densities. 38

17 Error Standard Deviation of the Samples 39

18 Simulated data and Predictive Distributions calculated from MCMC at $T = 20°C$ 40

19 Simulated data and Predictive Distributions calculated from MCMC at $T = 40°C$ The gray colors in the plots correspond to 50%, 80% 95% and 99% confidence envelopes. 40

20 Time (a point in red circle) at which B has decreased under a critical value with 99% certainty at 45°C 42

21 Sampled B values 43

22 Histogram of the minimum length values 44

1 Introduction

A stochastic model may be defined as a model in which ranges of values for each variable (in the form of probability distribution) are used. A stochastic model describes the sequence of outcomes from a particular, initial event in terms of the probability of each set of developments occurring through time. In contrast, deterministic models use single estimates to represent the value of each variable[24]. Stochastic models[3] can be considered in communication, transportation and marketing.

Optimization is central to any problem involving decision making[3]. The optimization of stochastic models is required in different fields being engineering or economics. In manufacturing, queuing models are used for modeling production processes. The task of decision making entails choosing between various alternatives. This choice is governed by the desire to make the best decision. The measure of goodness of the alternatives is described by an objective function or performance index. Optimization theory and methods deal with selecting the best alternative in the sense of the given objective function.

Practically, most optimization problems depend mostly on several model parameters, noise factors, uncontrollable parameters, etc, which are not given fixed quantities at the planning stage. Due to several types of stochastic uncertainties (physical uncertainty, statistical uncertainty, and model uncertainty) these parameters must be modelled by random variables having a certain probability distribution. A basic procedure to cope with these uncertainties is to replace first the unknown parameters by some chosen nominal values, e.g. estimates or guesses, of the parameters. With MCMC methods, we can then find the statistical distributions of these parameters and test for their deviations from their true values using some statistical methods.

1.1 Structure of the thesis

To achieve the objectives of this thesis, it is divided into four main parts, namely; introduction, theoretical background, numerical results and conclusion.

In the introductory part, a summary about optimization of stochastic models is given. The fields like Engineering, Finance and Chemical reactions in which the optimization of stochastic is mainly applicable are mentioned with some examples.

In the theoretical background part, the Bayesian inference, Monte Carlo integration and Markov chain Monte Carlo (MCMC) methods are discussed. In the Bayesian inference method, we discuss the meaning, importance and shortcomings of Bayesian inference for sampling from statistical distributions. As the results of weaknesses of Bayesian inference, in this part we then discuss the introduction of MCMC methods as alternative sampling methods especially from high dimensional and complicated statistical distributions. Several MCMC algorithms such as Metroplolis algorithm (MA), Metropolis Hastings algorithm (MH), Delayed rejection algorithm (DR), Adaptive metropolis algorithm (AM) and Delayed Rejection adaptive Metropolis algorithm (DRAM) are discussed. We also discuss about Gibbs sampler as one of the strong and useful sampling method.

Several methods of stochastic optimization are discussed in this theoretical part. The methods discussed here assume one single objective function with the decision variable x and the vector θ containing some parameters to be estimated. The methods of stochastic optimization such as mean criterion, sample average approximation, worst case criterion and response surface are described. However, in our example, only the worst case criterion and sample average approximation method are used.

In the numerical results part, the application of the theories are discussed with an example in a chemical reaction process. We consider a chemical reaction in which two components are allowed to flow through a pipe of a certain length to produce a new component. The aim is to minimize the cost in the production by avoiding the possibility of using a long pipe unnecessarily. We study how we can minimize the length of the pipe with optimal production of the new component using MCMC methods. Since we do not have real data in this case, we use the random data values generated by random number generators using MATLAB. The generated MCMC samples are then studied by observing some of their parameter statistics in relation to one another.

The conclusion part a summary of our numerical results is given, challenges encountered in using MCMC methods for optimization are discussed, suggestions for improving the methods and the future work to be done when time and resources are available are mentioned.

2 Theoretical background

2.1 Bayesian Inference

In Bayesian inference [15], the estimation of a parameter $\theta \in \mathbb{R}^d$ amounts to computation of its posterior distribution $p(\theta \,|\, \mathbf{y}_1, \ldots, \mathbf{y}_M)$, where $\{\mathbf{y}_1, \ldots, \mathbf{y}_M\}$ are the observed measurements. The posterior distribution can be computed with Bayes' rule:

$$p(\theta \,|\, \mathbf{y}_1, \ldots, \mathbf{y}_M) = \frac{p(\mathbf{y}_1, \ldots, \mathbf{y}_M \,|\, \theta)\, p(\theta)}{p(\mathbf{y}_1, \ldots, \mathbf{y}_M)} \tag{1}$$

where $p(\mathbf{y}_1, \ldots, \mathbf{y}_M \,|\, \theta)$ is the observation model, or the likelihood function, $p(\theta)$ is the prior distribution of parameters and $p(\mathbf{y}_1, \ldots, \mathbf{y}_M)$ is the normalization constant, which can be calculated as

$$p(\mathbf{y}_1, \ldots, \mathbf{y}_M) = \int_{\mathbb{R}^d} p(\mathbf{y}_1, \ldots, \mathbf{y}_M \,|\, \theta)\, p(\theta)\, d\theta. \tag{2}$$

For purposes of analyzing the posterior distribution, one is usually interested in computing the marginal distributions of parameters defined as:

$$p(\theta_j \,|\, \mathbf{y}_1, \ldots, \mathbf{y}_M) = \int_{\mathbb{R}^{d-1}} p(\theta \,|\, \mathbf{y}_1, \ldots, \mathbf{y}_M)\, d\theta_{-j}. \tag{3}$$

where θ_{-j} is a vector after removing the element θ_j from vector θ, or computing the moments of the form

$$\mathrm{E}[\mathbf{g}(\theta) \,|\, \mathbf{y}_1, \ldots, \mathbf{y}_M] = \int_{\mathbb{R}^d} \mathbf{g}(\theta)\, p(\theta \,|\, \mathbf{y}_1, \ldots, \mathbf{y}_M)\, d\theta, \tag{4}$$

and $\mathbf{g}(\theta)$ is some function of the parameter.

Inference is usually performed ignoring the normalizing constant $p(Y)$, thus utilizing $p(\theta|Y) \propto p(Y|\theta)p(\theta)$. Unfortunately, the closed form computation of integrals in Equations (2), (3), or (4) is possible only in very simple special cases.

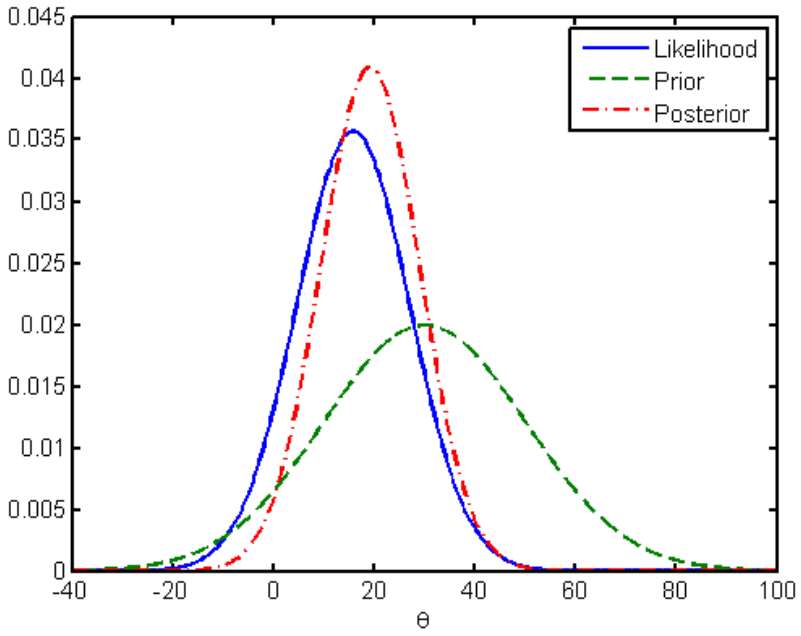

Figure 1: The illustration of Bayesian inference; Prior distribution, likelihood and posterior distribution computed from Gaussian distribution.

The relationship between the ingredients of Bayesian inference is shown in Figure 1. In this example, the distribution is Gaussian for which the variance is assumed to be known. The parameter values between -10 and 40 are favoured by the likelihood function while those below -20 an above 50 are not favoured. The posterior distribution combines the information from the prior distribution and the likelihood function using the Baye's Theorem.

Bayesian inference is an important tool in statistical analysis[8] because it provides

- parameter estimates with good statistical properties
- parsimonious descriptions of observed data
- predictions for missing data and forecasts of future data
- a computational framework for model estimation, selection and validation.

2.2 Monte Carlo Method and Markov chain Overview

A major limitation to the implementation of Bayesian inference approaches is that of obtaining the posterior distribution. The process often requires the integration of

the normalizing constant which is very difficult to calculate especially when dealing with complex and high-dimensional models. Because of this limitation in Bayesian approaches, researchers introduced Markov chain Monte Carlo (MCMC) methods, which attempt to simulate direct draws from some complex distribution and high-dimensional distributions of interest without utilizing the normalizing constant.

2.2.1 Monte Carlo Integration

Monte Carlo approach was originally developed for the purpose of computing integrals using random number generation. If for example we consider a complex integral

$$\int f(x)dx \tag{5}$$

If $f(x)$ can be decomposed into two functions, a function $g(x)$ and a probability density function $p(x)$ defined over the interval (a, b), then the integral in equation 5 can be expressed as an expectation of $g(x)$ over the density $p(x)$. That is

$$\int_a^b f(x)dx = \int g(x)p(x)dx = \mathbb{E}_{p(x)}[g(x)] \tag{6}$$

Thus, if a large number of random variables $x_1, x_2, \ldots, x_n$ is drawn from the density $p(x)$, then

$$\int_a^b f(x)dx = \mathbb{E}_{p(x)}[g(x)] \simeq \frac{1}{n}\sum_{i=1}^{n} g(x_i) \tag{7}$$

The Monte Carlo integration can be used to approximate posterior distributions required for a Bayesian analysis.

2.2.2 Markov Chains

If we let X_t to denote the value of a random variable at time t, and let the state space to be the range of possible X values. The random variable is a *Markov process* if the transition probabilities between different values in the state space depend only on the random variable's current state, i.e.,

$$Pr(X_{t+1} = S_j | X_0 = S_k, \ldots, X_t = S_i) = Pr(X_{t+1} = S_j | X_t = S_i) \tag{8}$$

This means that, the only information needed to predict the future of a Markov random variable is the current state of the random variable, knowledge of the values of the past states do not change the transition probability. *A Markov chain* can therefore be defined as a sequence of random variables $(X_0, X_1, \ldots, X_n)$ generated by a Markov process.

2.2.3 Markov Chain Monte Carlo Methods (MCMC)

Markov chain Monte Carlo (MCMC) methods are a class of algorithms for sampling from probability distributions based on constructing a Markov chain that has the desired distribution as its equilibrium distribution[24].

The MCMC algorithms generate a chain of parameter values $\theta_1, \theta_2, \cdots$, whose empirical distribution, in the histogram sense, asymptotically approaches the posterior distribution. That means that, candidate points are generated, and suitably accepted or rejected. A correct distribution is generated by favoring points with high values of the probability of posterior distribution, $\pi(\theta)$. The generation of parameters in the chain $\theta_n, n = 1, 2, \cdots$, is done by random number generators[2]. Each new point θ_{n+1} may only depend on the previous point θ_n. This is the *Markov property*. MCMC techniques are often applied to solve integration and optimization problems in large dimensional spaces.

With MCMC, it is possible to examine the distribution of unknown parameters in non-linear models, whereas traditional fitting techniques only produces single estimates for the parameters. Researchers are able to observe the shape and size of the distribution of the parameters that give valuable information related to correlations, uncertainty and identifiability of parameters.

In the Bayesian approach[5], the unknown parameter vector is considered as a random variable. The aim is to find its distribution. Before experiment, the parameter θ has a *prior distribution* $p(\theta)$. The measurements y update $p(\theta)$ to the *posterior distribution* by the *Bayes formula*

$$\pi(\theta) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta)d\theta} \tag{9}$$

where $p(y|\theta)$ is the *likelihood function* that gives the likelihood of the data y for given parameter value θ, $\pi(\theta) = p(\theta|y)$ is the posterior distribution that gives the probability distribution of parameter values, with the measured data y and $\int p(y|\theta)p(\theta)d\theta$ is the normalizing constant at which $\int_\theta \pi(\theta)d\theta = 1$.

The parameter vector θ and measurements y are connected by the model $y = f(x, \theta) + \epsilon$, where the experimental error ϵ is normally distributed, independent with the standard deviation of size σ, i.e $\epsilon \sim N(0, \sigma^2 I)$. It can easily be shown that

$$p(y|\theta) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\sum_{i=1}^{n} (y_i - f(x_i, \theta))^2 / 2\sigma^2 \right). \tag{10}$$

where n is the number of observations.

2.2.4 The Metropolis Algorithm

The most simplest MCMC method is the *Metropolis algorithm*[4] with the following steps:

(i) *Initialize* by choosing a starting point θ_1

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

(iii) *Accept* the candidate with probability

$$\alpha(\theta_n, \hat{\theta}) = \min\left(1, \frac{\pi(\hat{\theta})}{\pi(\theta_n)}\right) \tag{11}$$

If rejected, repeat the previous point in the chain. Go back to item (ii).

In this case, points with $\pi(\hat{\theta}) > \pi(\theta_n)$ are always accepted (go 'upwards'). Points with $\pi(\hat{\theta}) < \pi(\theta_n)$ (go 'downwards'), may still be accepted, with probability that is given by the ratio of the π values. In practice, this is done by generating a uniformly distributed random number $u \in [0, 1]$ and accepting $\hat{\theta}$ if $u \leq \pi(\hat{\theta})/\pi(\hat{\theta}_i)$.

We can note that only the ratios of π at different points are needed, so the 'difficult' of calculating the normalizing constant in the Bayes formula cancels out and is not needed.

The choice of the appropriate proposal distribution is very important in this case. It should be chosen in such a way that the 'sizes' of the proposal distribution q and target distribution suitably match. Choosing a suitable proposal for this purpose is not easy. An unsuitable proposal leads to inefficient sampling, typically due to

(i) the proposal is too large. Then the new candidates mostly miss the essential region π; they are chosen at points where $\pi \simeq 0$ and only rarely accepted.

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

In simple cases, the proposal might be relatively easy to find by hand-tuning. However, the 'size' of the proposal distribution is not a sufficient specification. In higher dimensions, especially, the shape and orientation of the proposal are crucial. The most typical proposal is a multidimensional Gaussian (Normal) distribution. In the (most typical) random walk version, the center point of the Gaussian proposal is chosen to be

the current point of the chain. The problem then is to find a covariance matrix that produces efficient sampling.

We can study the convergence of Metropolis algorithm to the target distribution. If we start the algorithm at time $t-1$ with a draw θ^{t-1} from the target distribution $p(\theta|y)$. We consider any two such points θ_a, θ_b drawn from $p(\theta|y)$ and labeled so that $p(\theta_b|y) \geq p(\theta_a|y)$. The unconditional probability density of a transition from θ_a to θ_b is

$$p(\theta^{t-1} = \theta_a, \theta^t = \theta_b|y) = p(\theta_a|y)J_t(\theta_b|\theta_a) \tag{12}$$

where $J_t(\theta_b|\theta_a)$ is the jumping distribution, and the acceptance probability is 1.

The unconditional probability density of a transition from θ_b to $theta_a$ is from

$$p(\theta^{t-1} = \theta_b, \theta^t = \theta_a|y) = p(\theta_b|y)J_t(\theta_a|\theta_b)\frac{p(\theta_a|y)}{p(\theta_b|y)} = p(\theta_a|y)J_t(\theta_a|\theta_b).$$

In Metropolis algorithm, it is required that $J_t(\cdot|\cdot)$ be symmetric and also the joint distribution $p(\theta^{t-1}, \theta^t|y)$ is symmetric. Therefore θ^t and θ^{t-1} have the same marginal distribution and so $p(\theta|y)$ is the stationary distribution of the Markov chain of θ.

Example:[28] Suppose we want to generate random samples from the Cauchy distribution using Metropolis algorithm. The probability density of the Cauchy is given by:

$$f(\theta) = \frac{1}{\pi(1+\theta^2)} \tag{13}$$

Because we do not need any normalizing constants in the Metropolis sampler, we can rewrite this to:

$$f(\theta) \propto \frac{1}{(1+\theta^2)} \tag{14}$$

where the proportional constant is $\frac{1}{\pi}$. If θ^* is a starting point, then the Metropolis acceptance probability becomes

$$\alpha = \min\left(1, \frac{1+[\theta^{(t)}]^2}{1+[\theta^*]^2}\right) \tag{15}$$

We use the Normal distribution as the proposal distribution. Our proposals are generated from a Normal $(\theta^{(t)}, \sigma)$ distribution. Therefore, the mean of the distribution is centered on the current state and the parameter σ, which needs to be set by the modeler, controls the variability of the proposed steps.

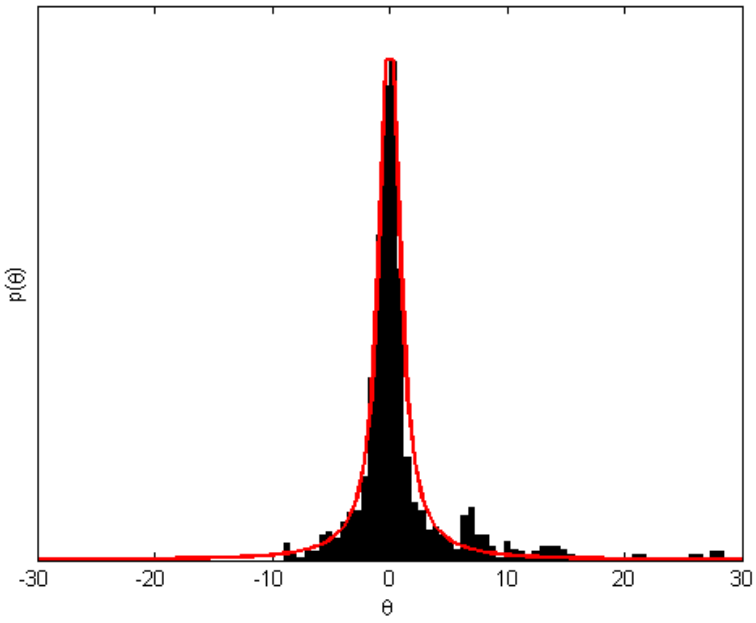

Figure 2: Theoretical density and Posterior distribution of cauchy distribution obtained by Metropolis algorithm

Figure 2 shows the simulation results for a single chain run for 500 iterations. The figure shows the theoretical density in the red line and the histogram shows the distribution of all 500 samples. We can see from the figure how Metropolis algorithm performs better in generating posterior distribution. Figure 3 shows the sequence of samples of one chain.

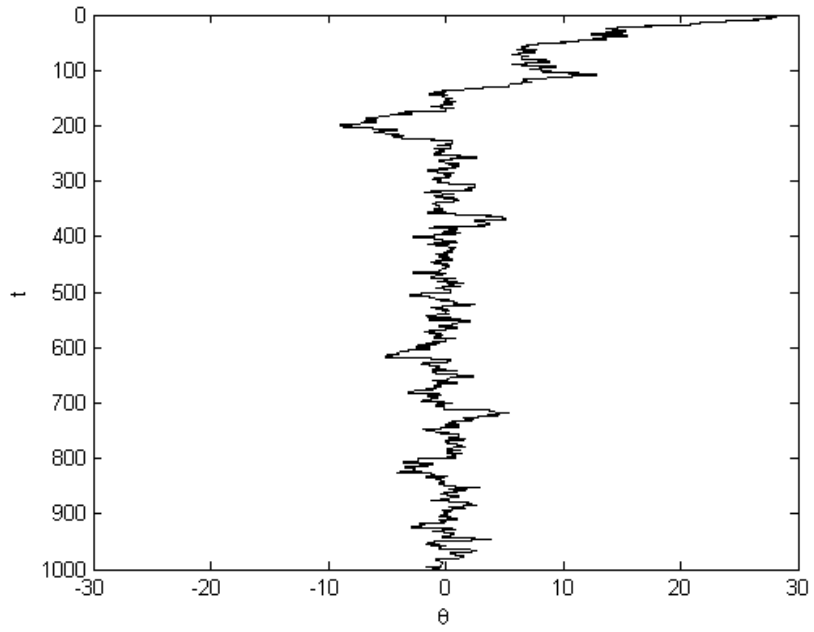

Figure 3: Sequence of samples

If one uses a different proposal however, say $t-$distibution, Beta distribution or Gamma distribution, the samples obtained will be different and sometimes deviates away from the correct posterior distribution. it is therefore important to choose a suitable proposal for this algorithm to produce samples that converge to the targeted distribution.

2.2.5 The Metropolis-Hastings Algorithm (MH)

The Metropolis-Hastings Algorithm (MH)[18] is a powerful Markov chain method to simulate multivariate distributions. If we have a posterior $p(\theta|y)$ that we want to sample from such that

(i) the posterior does not look like any distribution we know

(ii) the posterior consists of more than two parameters (grid approximations intractable)

(iii) some (or all) of the full conditionals do not look like any distributions we know,

then we can use the Metropolis-Hastings algorithm to sample from that distribution. The Metropolis-Hastings algorithm generalizes the basic Metropolis algorithm presented above in two ways. First, J_t needs no longer be symmetric; Second, to correct for the

asymmetry in the jumping rule, the ratio r is replaced by

$$r = \frac{p(\theta^*|y)/J_t(\theta^*|\theta^{(t-1)})}{p(\theta^{(t-1)}|y)/J_t(\theta^{(t-1)}|\theta^*)}.$$

The Metropolis-Hastings algorithm follows the following steps:

(i) Choose a starting value $\theta^{(0)}$.

(ii) At iteration t, draw a candidate θ^* from a jumping distribution $J_t(\theta^*|\theta^{(t-1)})$. The original Metropolis algorithm requires that $J_t(\theta^*|\theta^{(t-1)})$ be a symmetric distribution (such as the normal distribution), that is

$$J_t(\theta^*|\theta^{(t-1)}) = J_t(\theta^{(t-1)}|\theta^*)$$

With the Metropolis-Hastings algorithm the symmetry is unnecessary.

(iii) Compute an acceptance ratio (probability):

$$r = \frac{p(\theta^*|y)/J_t(\theta^*|\theta^{(t-1)})}{p(\theta^{(t-1)}|y)/J_t(\theta^{(t-1)}|\theta^*)}. \tag{16}$$

(iv) Accept θ^* as $\theta^{(t)}$ with probability $min(r, 1)$. If θ^* is not accepted, then $\theta^{(t)} = \theta^{(t-1)}$.

(v) Repeat steps number $(ii) - (iv)$ M times to get M draws from $p(\theta|y)$.

The ideal Metropolis-Hastings jumping rule is $J(\theta^*|\theta) = p(\theta^*|y)$ for all θ. However, iterative algorithm is applied to problem for which direct sampling is not possible. A good jumping distribution has the following properties[26]:

- For any θ, it is easy to sample from $J(\theta^*|\theta)$
- It is easy to compute the ratio r.
- Each jump goes a reasonable distance in the parameter space, otherwise the random walk moves too slowly.
- The jumps are not rejected too frequently, otherwise, the random walk wastes too much time standing still.

Weibull Example[27]: The Weibull distribution is used extensively in reliability, queueing theory, and many other engineering applications, partly for its ability to

describe different hazard rate behavior and partly for historic reasons. The Weibull distribution parameterized by α - the shape or slope, and $\eta^{-\frac{1}{\alpha}}$ - the scale,

$$f(x|\alpha,\eta) = \alpha\eta x^{\alpha-1}e^{-x^{\alpha}\eta},$$

is not a member of the exponential family of distributions if the shape of parameter varies, and explicit posteriors for α and η are impossible.

If we consider the prior $\pi(\alpha,\eta) \propto e^{-\alpha}\eta^{\beta-1}e^{-\beta\eta}$ and observations $data = [0.200 \quad 0.100 \quad 0.250]$, we can construct MCMC based on the Metropolis-Hastings algorithm and approximate posteriors for α and η by assuming the hyperparameter $\beta = 2$ and proposal distribution

$$q(\alpha',\eta'|\alpha,\eta) = \frac{1}{\alpha\eta}\exp\left\{-\frac{\alpha'}{\alpha} - \frac{\eta'}{\eta}\right\}$$

(product of two exponentials with means α and η).

Note that $q(\alpha',\eta'|\alpha,\eta) \neq q(\alpha,\eta|\alpha',\eta')$ and q does not cancel in the expression for p.

The proposals from independent exponential distributions for α and η are 3.0335 and 1.9037 respectively. "Burn in" 5000 out of 10000 simulations (usually $100-500$ is enough) to make sure that there is no influence of the initial values for α and η, and plot the histograms of their posterior distributions.

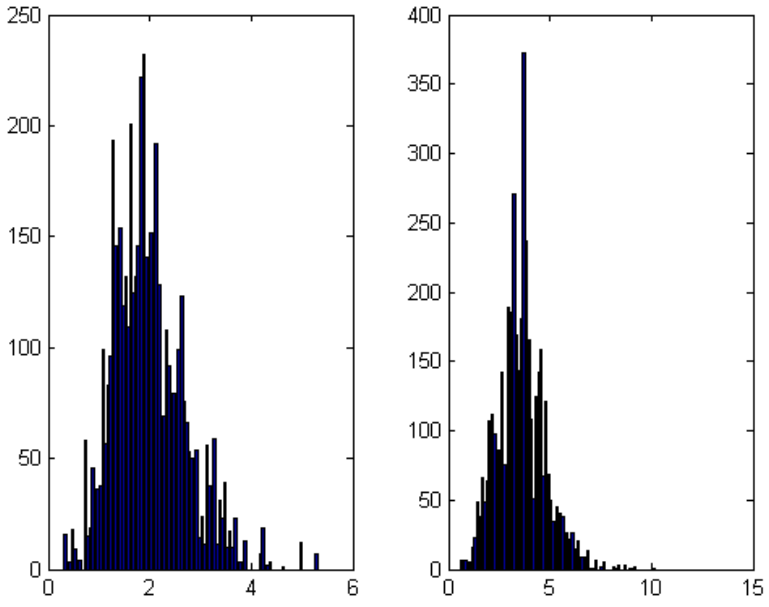

Figure 4: Histograms for the posteriors of α and η. The left panel is a Histogram for posterior of α while the right panel in a Histogram for the posterior of η

The mean and variance of α and η are shown in Table 1. These are desired Bayes estimators with their posterior precisions.

	α	η
Mean	1.9837	3.5213
Variance	0.5207	1.3249

Table 1: Mean and Variance for α and η

2.2.6 The Gibbs Sampler

The Gibbs sampler or Gibbs sampling a is an algorithm that generates a sequence of samples from the joint probability distribution of two or more random variables. The purpose of such a sequence is to approximate the joint distribution; to approximate the marginal distribution of one of the variables, or some subset of the variables (for example, the unknown parameters or latent variables); or to compute an integral (such as the expected value of one of the variables)[24]. Gibbs sampling is applicable when the joint distribution is not known explicitly or is difficult to sample from directly, but the conditional distribution of each variable is known and is easy (or at least, easier) to sample from.

Gibbs sampling can be viewed as a special case of the Metropolis-Hastings algorithm. If we define the iteration t to consist of a series of d steps, with step i of iteration t corresponding to an update of the sub-vector θ_i conditional on all the other elements of θ, then the jumping distribution $J_i, t(\cdot|\cdot)$ at step i of iteration t only jumps along the ith sub-vector and does so with the conditional posterior density of θ_i given θ_{-i}^{t-1}:

$$J_{i,t}^{Gibbs}(\theta^*|\theta^{t-1}) = \begin{cases} p(\theta_i^*|\theta_{-i}^{t-1}, y), & \text{if} \quad \theta_{-i}^* = \theta_{-i}^{t-1} \\ 0 & \text{otherwise} \end{cases}$$

The possible jumps are to parameter vectors θ^* that match θ^{t-1} on all components other than the ith. Under this jumping distribution, the acceptance rate at the ith step of iteration t is

$$\begin{aligned} \alpha(\theta^*|\theta^{t-1}) &= \min\left\{1, \frac{p(\theta^*|y)J_{i,t}^{Gibbs}(\theta^{t-1}|\theta^*)}{p(\theta^{t-1}|y)J_{i,t}^{Gibbs}(\theta^*|\theta^{t-1})}\right\} \\ &= \min\left\{1, \frac{p(\theta^*|y)p(\theta_i^{t-1}|\theta_{-i}^{t-1}, y)}{p(\theta^{t-1}|y)p(\theta_i^*|\theta_{-i}^{t-1}, y)}\right\} \\ &= \min\left\{1, \frac{p(\theta_{-i}^{t-1}|y)}{p(\theta_{-i}^{t-1}|y)}\right\} \\ &= 1. \end{aligned}$$

and thus every jump is accepted.

What we observe from Gibbs sampling is that, given a multivariate distribution it is simpler to sample from a conditional distribution than to marginalize by integrating over a joint distribution. The samples then approximate the joint distribution of all variables. Furthermore, the marginal distribution of any subset of variables can be approximated by simply examining the samples for that subset of variables, ignoring the rest. In addition, the expected value of any variable can be approximated by averaging over all the samples.

2.2.7 Adaptive Metropolis (AM)

In the MCMC methods described above, for example the Metropolis Hastings algorithm, the convergence of the algorithm requires a proper choice of a proposal distribution. It is often necessary to tune the scaling and other parameters before the algorithm will converge efficiently. However, choosing the proposal distribution in such a way that it will propose reasonable values that have a good chance of being accepted is not an easy task. In some extremely complicated target distribution, it is even difficult to know how to tune the corresponding parameters. In adaptive Metropolis (AM) algorithm [19], the problem of choosing a proposal can easily be solved because in this algorithm the Gaussian proposal distribution is updated along the process using the full information cumulated. In this method the proposal covariance is adapted by using the history of the chain generated so far. Adaptive MCMC methods modify the transitions on the way forward, in an effort to automatically tune the parameters and improve convergence, mainly basing on the historical information.

The Adaptive Metropolis algorithm follows the following steps [20]

(i) Choose 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 centered at the current value $N(\theta^{i-1}, C)$.

(iii) Accept or reject θ^* according to the Metropolis-Hastings (MH) acceptance probability in equation 16.

(iv) After an initial period of simulation, say for $i \geq 0$, adapt the proposal covariance matrix using the chain generated so far by $C = cov(\theta^0, \ldots, \theta^i)s + I\epsilon$. 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 step (*ii*) until enough values have been generated.

2.2.8 Delayed Rejection (DR)

Delayed rejection MCMC method[21] is a way of modifying the standard Metropolis-Hastings algorithm to improve efficiency of the resulting MCMC estimators. Delayed rejection method works in such a way that, upon rejection of a proposed candidate using a Metropolis Hastings (MH) algorithm, instead of advancing time and retaining the same position, a second move is proposed. The acceptance probability of the second stage candidate is computed so that reversibility of the Markov chain relative to the distribution of interest is preserved. The second stage proposal is allowed to depend on the current position of the chain and also on what have just been proposed and rejected. The process of delaying rejection can be iterated for a fixed or random number of stages. In its basic formulation, DR employs a given number of fixed proposals that are used at the different stages. The DR allows partial local adaptation of the proposal within each time step of the Markov chain, still retaining the Markov property and reversibility.

DR algorithm uses the standard Metropolis Hastings probability of acceptance in the first stage, which can be written as:

$$\alpha_1(\theta, \theta^*) = \min\left\{1, \frac{\pi(\theta^*)q_1(\theta^*, \theta)}{\pi(\theta)q_1(\theta, \theta^*)}\right\} \tag{17}$$

where θ is the current point, θ^* is the new proposed 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 using the acceptance probability [20]

$$\alpha_2(\theta, \theta^*, \theta^{**}) = \min\left\{1, \frac{\pi(\theta^{**})q_1(\theta^{**}, \theta^*)q_2(\theta^{**}, \theta^*, \theta)[1 - \alpha_1(\theta^{**}, \theta^*)]}{\pi(\theta)q_1(\theta, \theta^*)q_2(\theta, \theta^*, \theta^{**})[1 - \alpha_1(\theta, \theta^*)]}\right\}. \tag{18}$$

Since the reversibility property of the MCMC chain is preserved, DR method also leads to the same stationary distribution π as the Metropolis hastings Algorithm.

2.2.9 Delayed Rejection Adaptive Metropolis (DRAM)

The Delayed Rejection Adaptive Metropolis (DRAM) method of MCMC[20] is a combination of two Markov chain Monte Carlo (MCMC) algorithms, delayed rejection and Adaptive Metropolis. Most of the methods discussed above have the problem of adapting the proposal, thus the initial accepted values to start with are to be chosen at the beginning. For example, in Adaptive Metropolis, the choice of an initial proposal which is good enough for the adaptation to take place is very important. If the proposal is too wide for example, no new points will be accepted as there will be no history to adapt to.

With DRAM method, the problem of proposal adaptation and selection of initial proposal can be solved since this method uses ideas from the Adaptive Metropolis and Delayed Rejection method. The use of different proposals using delayed rejection and adapting them using adaptive metropolis enables the possibility of having different ways of getting a good proposal. One of the ways is to obtain a master proposal[20]. In this way, a master proposal is tried. After a rejection, another modified version of the first proposal is tried using delayed rejection algorithm. This 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. It is also possible to get better results for non-Gaussian target distributions by using different Gaussian proposals.

The algorithm for the DRAM can be described as follows

(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 $C^{(i)}, i = 1, \ldots, N_{try}$, 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:

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

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

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

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

2.3 Convergence Diagnostics and Chain Length

After generating MCMC samples using either of the algorithms described above, we need to study how our samples converge to the target distribution. It is also important to determine how long the chain should be in order that our targeted distribution is attained. That is we need to set a stopping criteria that will enable us to obtain sufficient length of the chain. There are some methods devised to address how the stopping criteria can be set. These methods used to address the issue of convergence and chain length belong to the field of MCMC convergence diagnostics[16].

It is not easy to assess the convergence in MCMC algorithms in general, because almost every MCMC algorithm has a different rate of convergence depending on the target distribution. It is hard to construct effective analytical estimates for the convergence rate and accuracy of MCMC algorithms. That is, there is no any analytical formula or stopping criteria for the algorithm, that would uniquely determine the run length. The MCMC algorithms can be falsified, but not verified; we can never be totally sure that the sample created is a comprehensive representation of the posterior distribution[16]. Convergence diagnostics are methods for making educated guesses about the convergence of the algorithms.

The most simple and straightforward methods seem to be based on monitoring the created chain visually using some statistical plots that show the behavior of the chain with time or with some statistical tools. Some of the methods used are as described below.

As an example to demonstrate how we can visualize and monitor the properties of the generated MCMC samples we consider a chemical reaction[4] described below:

To determine the unknown rate coefficients k_1 and k_2 in radioactive decay reaction $A \xrightarrow{k_1} B \xrightarrow{k_2} C$; the components A, B were measured starting with the initial values $A = 1$; $B = 0$ at $t = 0$. The data below was obtained

Time	0	1	2	3	4	5	6	7	8	9
A	1	0.504	0.185	0.217	0.023	0.101	0.058	0.064	0.000	0.082
B	0	0.415	0.488	0.594	0.505	0.493	0.457	0.394	0.334	0.309

Table 2: Data Measurements

The system is modeled as Ordinary Differential Equation (ODE) system:

$$\begin{aligned} \frac{dA}{dt} &= -k_1 A \\ \frac{dB}{dt} &= k_1 A - k_2 B \\ \frac{dC}{dt} &= k_2 B \end{aligned}$$

We can create an MCMC chain to produce samples from the posterior probability distribution of the parameters k_1 and k_2. We first solve the system using ODE solvers in MATLAB and estimate the values for k_1 and k_2 by least squares fitting by first guessing the original values of the parameters to be [1 1]. We then compute the covariance proposal of the distribution using jacobian matrix (see appendix for more details). The MCMC samples of the parameters is then generated using Metropolis Hastings algorithm for 5000 iterations.

Posterior Predictive Distribution

The posterior predictive distribution[22] is the distribution of unobserved observations (prediction) conditional on the observed data. It is one of the best and most flexible approaches of examining the fit of the model. The posterior predictive distribution for a model is the distribution of future observations that could arise from the model under consideration. It takes into account both parametric uncertainty and sampling uncertainty from the original model. Parametric uncertainty is captured via the posterior distribution for the parameters, a sample of which is the result of simulation using MCMC methods. Sampling uncertainty is captured via the specification of the sampling density for the data.

If y is the observed data, θ the parameter, and y_{pred} is unobserved data; the posterior predictive distribution can be defined as:

$$\begin{aligned} p(y_{pred}|y) &= \int p(y_{pred}|y,\theta)p(\theta|y)d\theta \quad (19) \\ &= \int p(y_{pred}|\theta,y)p(\theta|y)d\theta \quad (20) \end{aligned}$$

If it is assumed that the observed and unobserved data are conditional independent given θ , the posterior predictive distribution can be simplified to:

$$p(y_{pred}|y) = \int p(y_{pred}|\theta)p(\theta|y)d\theta \tag{21}$$

The posterior predictive distribution is an integral of the likelihood function $p(y_{pred}|\theta)$ with respect to the posterior distribution $p(\theta|y)$.

The model prediction curves of the parameters are often more interesting than the distributions of the parameters. These will also be used in the optimization part when choosing the critical point at below which one the components in the chemical reaction vanishes. The posterior predictive distribution gives the $1-\alpha$ confidence intervals for model prediction, and can be formed as follows[16].

1. Generate MCMC chain for unknown parameters
2. Calculate the model prediction for the chain in order to create a chain for the model prediction
3. Calculate the confidence interval for the prediction values separately for every time point
   - Sort the values
   - Take the $\alpha/2$ and $1-\alpha/2$ (see appendix for more details) empirical percentile of the samples (through interpolation)
4. Plot the limits for every (time) point

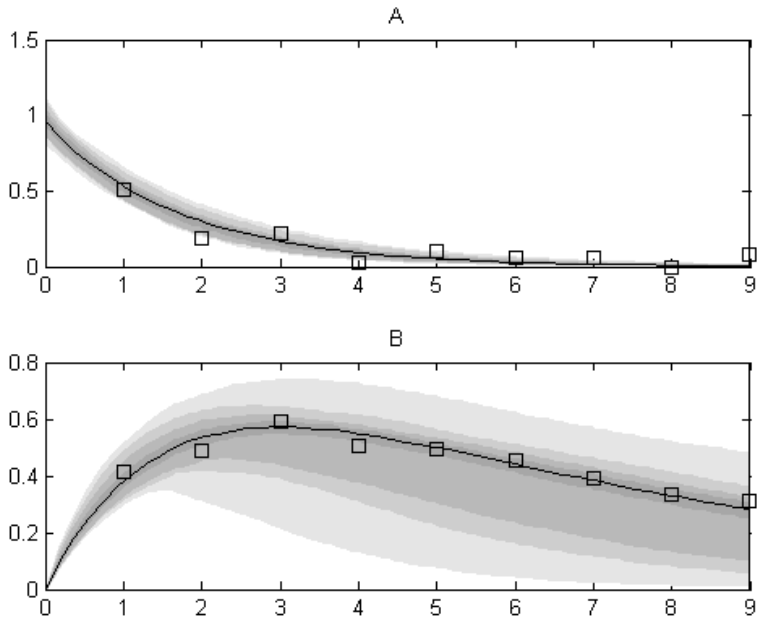

Figure 5: The posterior predictive distribution for the components A and B in our example model

The posterior predictive distribution for the components A and B in our example model is shown in figure 5. The confidence interval marked with darker gray in the figure is produced by calculating the response curves with the parameter values given in the sampled chain for given time points.

Posterior predictive distribution is different from the prior predictive distribution[28]. The prior predictive distribution $p(y_{pred})$, also known as the marginal distribution of the data is an integral of the likelihood function with respect to the prior distribution. That is;

$$p(y_{pred}) = \int p(y_{pred}|\theta)p(\theta)d\theta \tag{22}$$

This distribution is not conditional on observed data. It is the distribution the model predicts over the observed variables, before any data is considered. These are the predictions the model makes when there is no observed data to condition on. Prior predictive distribution can be used to predict data patterns that the modeler knows will either not occur in an experiment or occur only rarely.

The prior predictive distribution can also be used to create synthetic data sets for testing purposes. Before applying any probabilistic model to real data, it is normally helpful to do posterior inference on the model on data that were produced by itself. This helps

to check the inference procedures that were used. If the inference works properly, the model should infer distributions over the latent variables that are similar to the ones that were sampled when producing the artificial data.

Autocorrelation Function

Samples produced from distributions by MCMC methods are not independent. Instead, MCMC algorithms produce samples that are autocorrelated. The autocorrelation function (ACF) measures how correlated the values in the chain are with their close neighbors. The lag is the distance between the two chains to be compared. The higher the autocorrelation in the chain, the larger the MCMC variance and the worse the approximation is. That is, when the parameters are highly correlated in the model the sampling to explore the entire posterior distribution will be slow.

Empirical auto-correlation functions, on the other hand, can be used to determine how correlated successive draws of the chain are, and also how quickly there is convergence to stationarity[6]. If the auto-correlations decay very slowly, then the Monte Carlo estimates based on the whole sample would not be reliable. This is mostly caused by the initial transient period. In this case it the initial transient period should be discarded before Monte Carlo estimates are formed.

For example, for the chemical reaction problem above, the autocorrelation function for the parameters k_1 and k_2 can be observed as shown in Figure 6

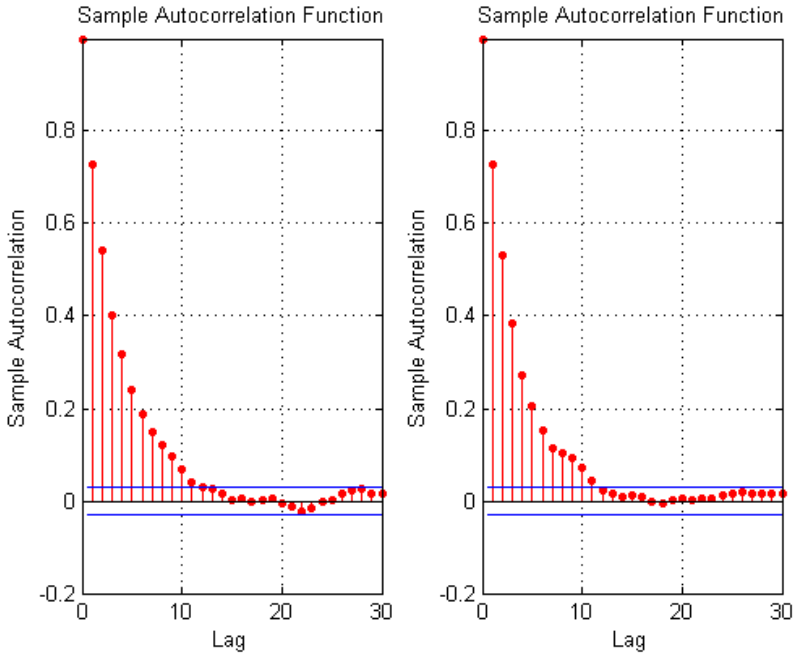

Figure 6: The Autocorrelation Function for the parameters k_1 and k_2. The left panel if the The Autocorrelation Function for k_1 and the right panel is the Autocorrelation Function both with lags 30

From the figure, we can see that the Autocorrelation Function for both parameters converges and stabilizes to zero with small lags. This means that the generated MCMC chain for the parameters are highly correlated.

Trace Plots of Parameters

The trace plot for the parameter (or history plot) shows the parameter value at time t against the iteration number. We can observe whether our chain gets stuck in certain areas of the parameter space, which indicates bad mixing. Trace plots are frequently used to try to assess informally whether the Markov chain has converged[23]. If the model has converged, the trace plot will move like a snake around the mean of the distribution.

The trace plots for the parameters k_1 and k_2 in the chemical reaction problem described above can be visualized in Figure 7.

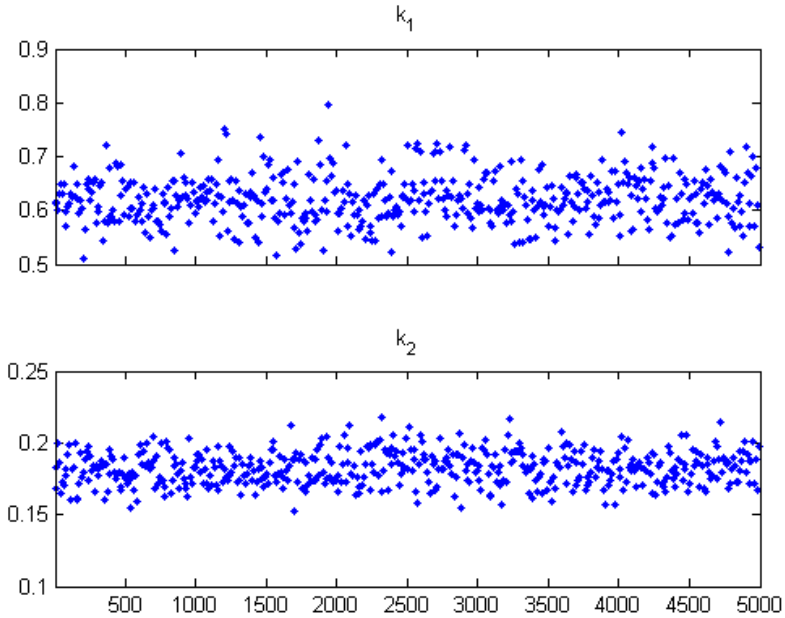

Figure 7: Trace Plots for the parameters k_1 and k_2.

From the figure, we observe that, the mixing of the chain for the both parameters k_1 and k_2 begins just at the beginning. The plots shows how the generated chain for the parameters move around the mean. The trace plot is therefore an important way of visualizing the behavior of the chain, and hence if needed we can make an improvement. If for example the mixing begins after a certain time, we can extend the chain by increasing the length and neglecting the part where mixing is not good.

Scatter Plots

A scatter plot for the parameter values is a graph of plotted points that show the

relationship between the parameters. It can be used to monitor how the parameters from the MCMC are correlated to one another. In this case, one might want to construct a confidence region, that would approximately include a certain percentile of the two-dimensional marginal distribution mass. One could use the histogram approach in two dimensions as well by assigning a grid on the axes and see how many points fall into each box in the grid.

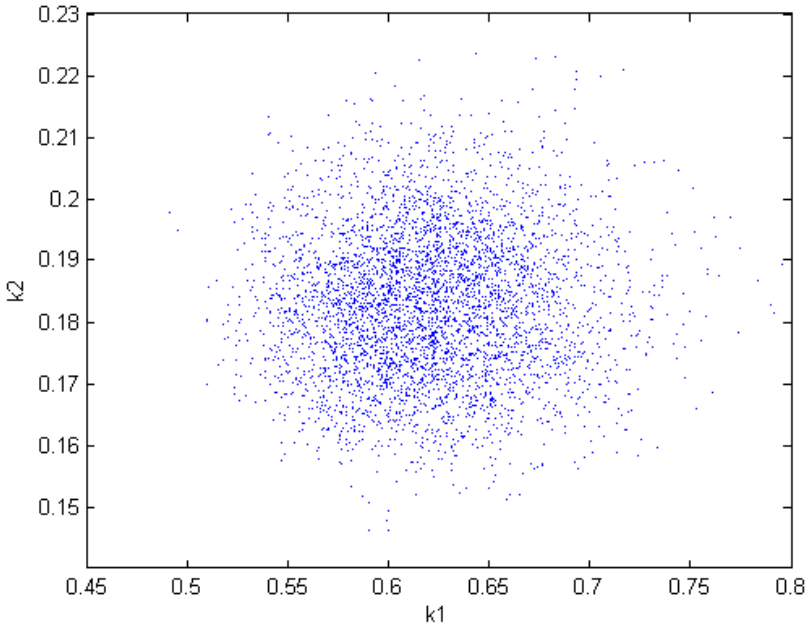

Figure 8: Sampled parameter values plotted pairwise for the parameters k_1 and k_2 in our example model.

The scatter plot for the parameters k_1 and k_2 for the chemical reaction above is shown in Figure 8. For one-dimensional density plots and two-dimensional scatter plots, we can estimate the density in a desired point using a sum of kernel functions at every data point as shown in Figure 9.

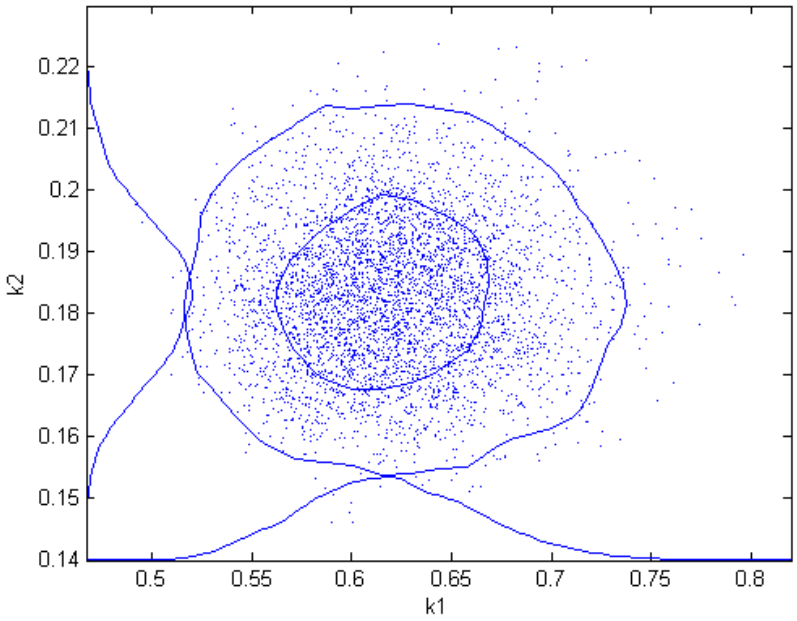

Figure 9: A scatter plot of a parameter pair with confidence regions based on kernel densities k_1 and k_2 in our example model.

The kernel density approach as illustrated in Figure 9 shows both pairwise scatter plot and one-dimensional marginal distributions. The kernel method is applied to one-dimensional parameter plots to get a smooth-looking PDF for the parameters. In two-dimensional density estimation, a grid is set on the axes and the density is estimated at each point.

3 Stochastic Optimization

Stochastic optimization plays a significant role in the analysis, design, and operation of modern systems. Methods for stochastic optimization provide a means of coping with inherent systems noise and with models or systems that are highly nonlinear, high dimensional, or inappropriate for classical deterministic methods of optimization[11].

As stated in the introductory part of this thesis, optimization is the art of finding the best among several alternatives in decision making. Any optimization problem is characterized by a set of possible decisions from which one best solution has to be chosen based on the purpose of the problem.

Suppose S is the set of possible solutions or the feasible set and x is the decision variable. If $x \in S$, then x is a feasible set, otherwise infeasible. The net costs caused by decision x are measured by a real valued objective function $F(x, \theta)$ where θ is a vector of variables which have to be estimated. The goal is to find the best decision with minimal costs. In multi-criteria decision making problems, the optimization problem has several competing objective functions. In our case, a one single objective function will be considered.

An optimization problem has the general form of

$$\underset{x \in S}{\text{Minimize}}\, F(x, \theta)$$

The minimization problem can be changed to maximization problem by considering $-F(x, \theta)$ instead of $F(x, \theta)$.

Solution approaches to stochastic models optimization are driven by the type of probability distributions governing the random parameters. In stochastic optimization, the best decision x of the objective function is determined by comparing the probability distributions of all available alternatives, and not fixed values as the case of deterministic optimization.

Whereas deterministic optimization problems are formulated with known parameters, real world problems almost invariably include some unknown parameters, and hence an attention on how to take into account these unknown parameters is required. There are different approaches to choose x in the stochastic situation depending on the nature of the used model.

3.1 Stochastic Optimization Problems

A stochastic optimization problem[3] is characterized by the fact that not all decision-relevant data are exactly known at the time, when the decision has to be made.

Mathematically, uncertainty is described by random variables, which appear in the optimization model. The distributions of all random variables in the model are assumed to be known but their concrete realizations are not known. To find the statistical distributions of these variables, the data must be collected, parameters must be estimated or models must be validated. In the situations where no data available, data may be generated using random number generators.

3.2 Problem Formulation

In any given problem to be optimized, it is very important to formulate the model depending on the the purpose of the problem to be solved. Decision problems are often formulated as optimization problems, and thus in many situations decision makers wish to solve optimization problems which depend on parameters which are unknown. Formulation and solving optimization problems, both conceptually and numerically is not always easy especially when the model to be formulated needs to take the uncertainty into account. The difficulty starts at the conceptual stage of modeling. Usually there are a variety of ways in which the uncertainty can be formalized. In the formulation of optimization problems, one usually attempts to find a good trade-off between the realism of the optimization model, which usually affects the usefulness and quality of the obtained decisions, and the tractability of the problem, so that it could be solved analytically or numerically.

There are different approaches that can be used for formulating and solving optimization problems under uncertainty. However, there are steps to be followed during problem formulation and model optimization.

Given a nonlinear model $\mathbf{s} = f(\mathbf{x}, \theta, \sigma\xi)$, where ξ is the random noise, with measurements $\mathbf{y} = g(\mathbf{s}) + \epsilon$.
To optimize such a model, we need to specify and estimate the parameters incorporated in the model, fit the model to the data and then use the optimized parameters to optimize the given cost function. It is not always that we have measured data for the problem. Sometimes we need to generate data using random number generators.

In this work, we only consider the optimization task with uncertain model parameter values. In summary, the optimization of stochastic models follows the following steps:

1. Formulate model: $\mathbf{s} = f(\mathbf{x}, \theta, \sigma(\xi))$
2. Collect measurements: $\mathbf{y} = g(\mathbf{s}) + \epsilon$
3. Estimate and fit model to data so as to get the posterior $p(\theta)$ distribution of θ
4. Use the model to optimize the problem:

$$\underset{x}{\text{Max}}\ c(x, \theta), \quad \theta \in p(\theta)$$

3.3 Methods of Optimization

In Stochastic optimization, there is a number of different methods depending on the nature of the problem to be optimized, and also depending on what the researcher intends to optimize from that model.

3.3.1 Mean Criterion

In mean criterion method of stochastic optimization, the decision is based on the expectation of the random objective function. Optimization of the stochastic model based on the mean can be done by direct Monte Carlo sampling. The average of $c(\mathbf{x}, \theta)$ is taken over a large number of the parameters θ and the mean of these parameters is then optimized. This reduces the risk of obtaining $\mathbf{x}$ that gives the optimal value of $c(\mathbf{x}, \theta)$ for a specific value of θ[1]. This method can be defined by the integral

$$C(\mathbf{x}) = \mathbf{E}_{p(\theta|\mathbf{y})}[c(\mathbf{x}, \theta)] = \int c(\mathbf{x}, \theta) p(\theta|\mathbf{y}) d\theta \tag{23}$$

which can be approximated using the generated MCMC samples from $p(\theta|\mathbf{y})$. Good output is obtained by picking a large number of parameters(samples) $\theta_1, \ldots, \theta_N$ from the MCMC generated samples and apply the direct Monte carlo approximation:

$$C(\mathbf{x}) \approx \frac{1}{N} \sum_{i=1}^{N} c(\mathbf{x}, \theta_i). \tag{24}$$

3.3.2 Sample Average Approximation (SAA)

In sample average approximation (SAA) method of stochastic optimization[25], the approximation of the optimal value of the objective function is obtained over a large number of samples as in the mean criterion method above. However, in this method the sample

is first divided into few subsets of samples. The optimization is done separately to each subset separately. The optimal value to the optimization problem is then calculated by averaging the solutions of the subsets.

If for example we consider a stochastic problem

$$\mathrm{Min}_{x\in X}\left\{ f(x) = \mathbb{E}[F(x,\theta)] \right\} \tag{25}$$

where X is non empty closed subset of $\mathbb{R}^n$, $F(x,\theta)$ is the objective function, θ is a random vector of parameters whose posterior probability distributions can be defined and $f(x)$ is a well defined and finite valued expectation function for all $x \in X$.

Suppose that we have a sample $\theta_1, \theta_2, \ldots, \theta^N$ of random vector θ for N observations generated by MCMC techniques. For any $x \in X$, we can estimate the expectation value $f(x)$ by averaging values $F(x,\theta^i), \quad i = 1, 2, \ldots, N$. This then leads to the sample average approximation (SAA) given as

$$\mathrm{Min}_{x\in X}\left\{ \hat{f}_N(x) = \frac{1}{N}\sum_{i=1}^{N} F(x,\theta^i) \right\} \tag{26}$$

for the problem in equation 25.

If the random vector $\theta^i, \quad i = 1, 2, \ldots, N$ is independently identically distributed (iid), then by the Law of Large Numbers (see appendix for more details) under some regularity conditions, $\hat{f}_N(x)$ converges to $f(x)$ as N approaches to infinity ($N \to \infty$).

The sample average function, $\hat{f}_N(x)$ is known as an unbiased estimator of $f(x)$. That is $\mathbb{E}[\hat{f}_N(x)] = f(x)$. It is therefore expected that the optimal value and the optimal solutions of the problem in equation 26 converge to the true solution of the stochastic optimization problem in equation 25

3.3.3 Risk Models

The mean criterion method explained above does not take into account the variability in it. The variability in this method can be taken into account by adding a requirement of small variance to the optimization criterion. This method is commonly used in the risk models[3], in which the decision is not based only on the expected value but also on the variability so as to avoid high risks. The risk measures are therefore considered within the decision process via the loss function that measures the negative effect of a deviation from the expectation. This can be done by penalizing large standard deviations in the parameters.

$$C(\mathbf{x}) = \mathbf{E}_{p(\theta|\mathbf{y})}[c(\mathbf{x},\theta)] - \alpha \mathrm{Std}_{p(\theta|\mathbf{y})}[c(\mathbf{x},\theta)] \tag{27}$$

where α defines the weight given for the variability in the criterion. This method is sometimes called robust mean criterion.

3.3.4 Worst Case Criterion

In this method of stochastic optimization, the optimal value is obtained by maximizing the worst value of $c(\mathbf{x}, \theta)$:

$$C(\mathbf{x}) = \min_{\theta} c(\mathbf{x}, \theta). \tag{28}$$

This can be done by calculating $c(\mathbf{x}, \theta)$ with different possible values for θ and finding the minimum from the calculated samples.

3.3.5 Response Surface Method

Response surface method [14] for stochastic optimization is based on approximations of the objective function. This method uses Mathematical and Statistical techniques to approximate and optimize stochastic functions. Given the objective function, the best local solution is determined using regression analysis based on the number of observation of the objective function.

If we assume the stochastic objective function, we can optimize the expectation of the stochastic output of the minimization problem. Mathematically, this can be written as

$$\min f : D \to \Re, D \subseteq \Re^n \tag{29}$$

where $f(x_1, x_2, \ldots, x_n)$ is equal to $E(G(x_1, x_2, \ldots x_n))$. $G(x_1, x_2, \ldots x_n)$ denotes the stochastic output for given input $\{x_1, x_2, \ldots, x_n\}$, and $E(G(x_1, x_2, \ldots x_n))$ denotes its expectation. The variance in the function values is assumed to be unknown. A simulation aims at finding the model parameters that optimize the problem, in this case minimization of the cost function.

Response surface method comprises of two steps. In the first step, the stochastic objective function is locally approximated by a first-order polynomial. In the second step, the objective function is approximated by a second-order polynomial. The approximation of the stochastic objective is evaluated in the points of an experimental design. The response surface method [3] technique collects the simulated data first, approximates the response function $F(\cdot)$ by some interpolation $\hat{F(\cdot)}$ and optimizes in a second phase this interpolated function. In summary, the response surface method of stochastic optimization follow the following steps:

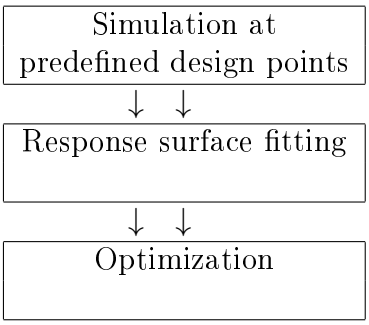

Table 3: Response surface technique

3.3.6 Process Optimization

In process optimization[1], the uncertainties and reliability of the model are taken into account by using Markov Chain Monte Carlo (MCMC) according to the Bayesian paradigm. All the uncertainties are considered as random variables with statistical distributions. The data is equally fitted and the statistical distribution of the unknown parameters is determined basing on the prior distribution. Data is simulated with the assumed true parameter values, based on which MCMC parameter estimation is performed. A process optimization criterion is performed basing on the resulting MCMC chain, both by fixing θ to its maximum a posteriori (MAP) estimate (least squares estimate) and by taking the possible parameter values given by MCMC into account in the proposed way.

4 Numerical Results

4.1 Chemical Reaction

To implement the theories of optimization using MCMC methods as discussed above, a temperature dependency chemical reaction was considered in the first case.

$$A + B \rightarrow C \tag{30}$$

In this reaction, the compounds A and B are fed in a pipe of length L at a certain temperature to produce a new compound C.

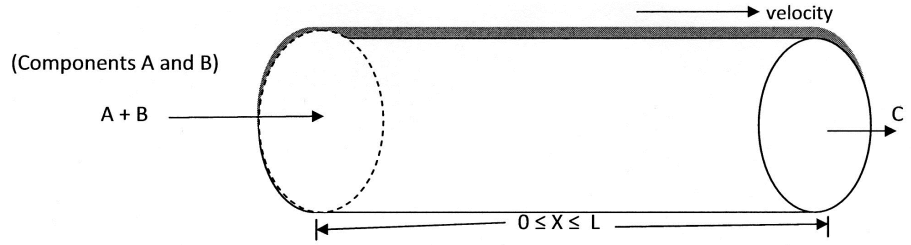

Figure 10: Reactants A and B are fed in the pipe of length L to produce C.

The aim is to optimize (minimize) the length of the pipe, x at which the reaction between A and B will produce the optimal (maximum) product C. The objective cost function for this problem is therefore $f(x, \theta)$, where x is the length of the pipe to be determined and θ is a vector containing the parameters to be estimated.

The chemical reaction equation(30) is modelled as the ordinary differential equation (ODE) system

$$\begin{aligned} \frac{dA}{dt} &= -k(T)AB \\ \frac{dB}{dt} &= -k(T)AB \\ \frac{dC}{dt} &= k(T)AB. \end{aligned} \tag{31}$$

The differential equations above means that C is produced at a rate proportional to the product of the concentration of A and B. The rate of change of A is the same as the rate of change of C, that is per each C that is produced one A is lost. That is similar to B.

The temperature dependency is expressed by the Arrhenius law

$$k(T) = K_{ref} e^{-zE} \tag{32}$$

where T is temperature (in Kelvin), E the activation energy, R is the gas constant ($R = 8.314$ in SI units). $K_{ref} = Ae^{-E/RT_{mean}}$, $z = 1/R(1/T - 1/T_{mean})$, where A is the amplitude and T_{mean} is the mean temperature between the minimum and maximum temperatures used in the experiment. In the parameter estimation part, the parameter $\theta = (K_{ref}, E)$ is estimated by MCMC methods. The initial reactants A_0 and B_0 fo A and B respectively were also included in the estimation to see how they behave in relation to other parameters.

4.1.1 Parameter Estimation

The concentrations of the reactants A and B are assumed to be uniform randomly distributed. The system is then solved at four different temperatures, $T = 20°C$, $T = 40°C$, $T = 50°C$ and $T = 60°C$ by first assuming true parameter values to be $\theta_{true} = (0.7, 1.0 \times 10^5)$ chosen by hand. We use the temperature $T_{ref} = 30°C$ as a reference.

Figure 11 shows the solution of the ODE system at different temperatures with the fitted parameters.

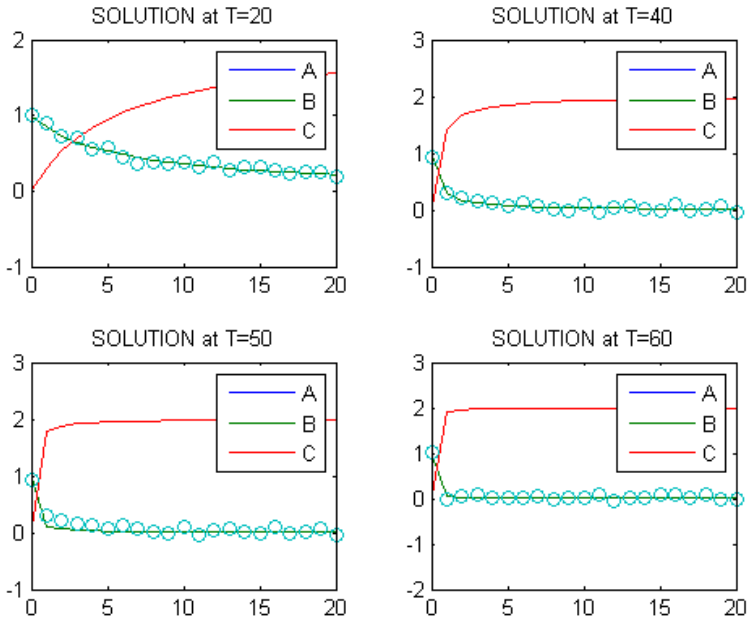

Figure 11: Solution of the ODE system at different Temperatures. The dots in circle form are the measurements.

From Figure 11, it can be seen that the reaction between the compounds A and B increases with the increase of temperature. A is not seen in the figure because it reacts at the same rate as B, so they are coinciding. At high temperatures, the reaction rate is fast and the steady state is attained after a short time.

The parameters K_{ref} and E are fitted by least squares method (see appendix for more details) and a perfect fit is obtained. As an example, the ODE system is again solved at two different temperatures, $T = 20°\text{C}$ and $T = 40°\text{C}$ with the fitted parameter values. From figure 12, it can also be observed that the reaction rate is fast at high temperature $T = 40°\text{C}$ as compared to that at small temperature $T = 20°\text{C}$.

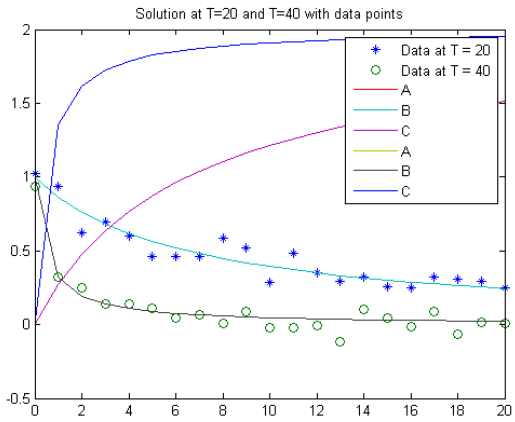

Figure 12: Solution at two temperatures with the generated data

4.1.2 Statistical Analysis

After obtaining the optimal parameter values, the next step is to find their statistical behaviors basing on MCMC samples generated. Using Adaptive Metropolis Algorithm[19], we generate 10000 MCMC samples . Other MCMC methods can also be used for the same purpose. From the MCMC samples generated, we compute the autocorrelation function and determined the trends of the parameters. The relationship between parameters is studied through the scatter plots and their pairwise marginal density estimates. We finally compute the Predictive distributions for the parameters to see how the model can capture the future observations.

Sample Autocorrelation Function

The sample autocorrelation function for the parameters is shown in Figure 13. The plotted function autocorrelation for the parameters is with 60 lags. From the figure, the autocorrelation value of E samples drop to zero faster than K_{ref} samples, which means that the E samples have small variabilities as compared to K_{ref}. How ever, in general the approximation of both parameters is still better as the sample autocorrelation values drop moves around zero with few lags. This shows that the variance between the samples is small.

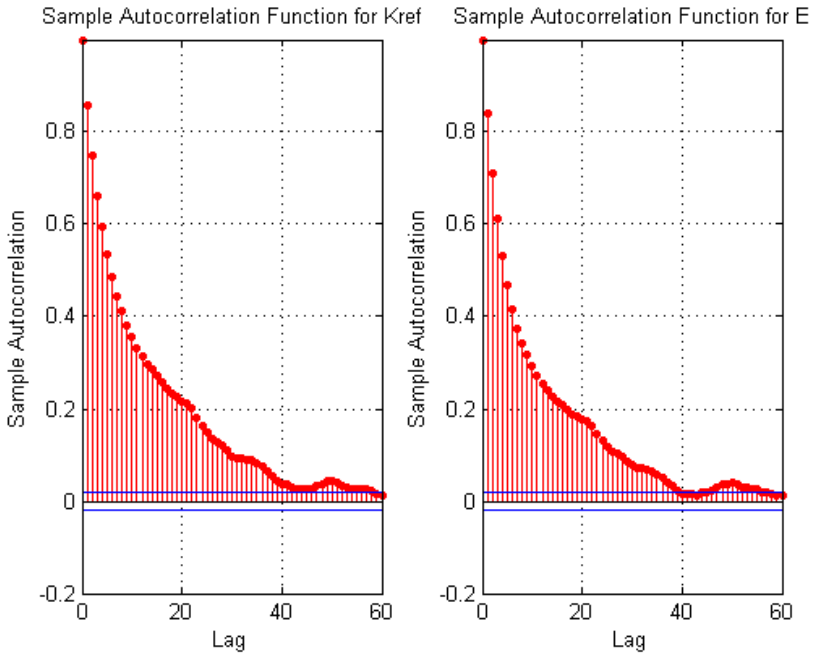

Figure 13: Sample Autocorrelation Function for the Parameters

.

Trace Plots of Parameters

The trend of the parameters is shown in Figure 14. From the figure the trace plots show that the parameters K_{ref} and E have relatively good mixing just from the beginning of the chain .This means that the estimation of these parameters was relatively good.

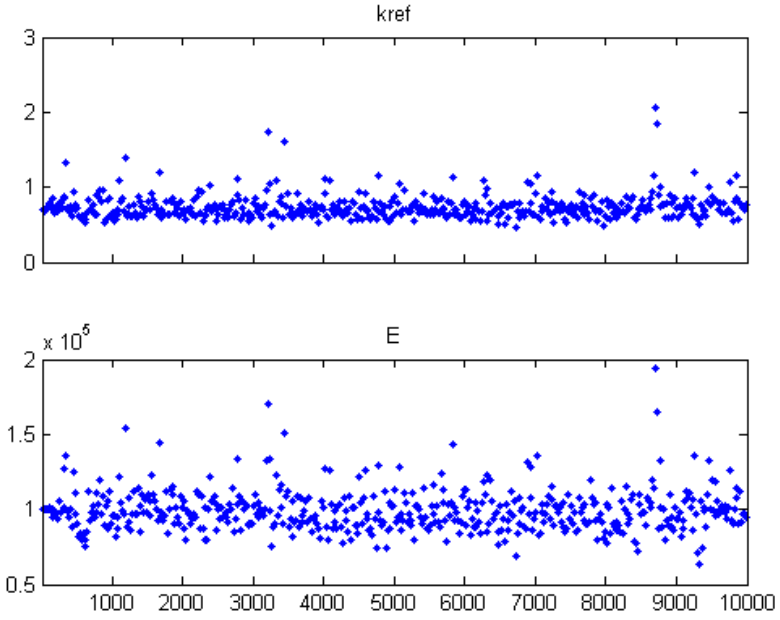

Figure 14: Trace Plots of Parameters: The trace plots for both K_{ref} and E show relatively good mixing as the values are moving like a snake around the mean, though that of E starts to show the mixing after a short time. Generally, we can conclude that the mixing of the parameters was relatively good.

Scatter Plot

The scatter plot for the parameters is shown in Figure 15. The scatter plot shows how the parameters are correlated to each other.

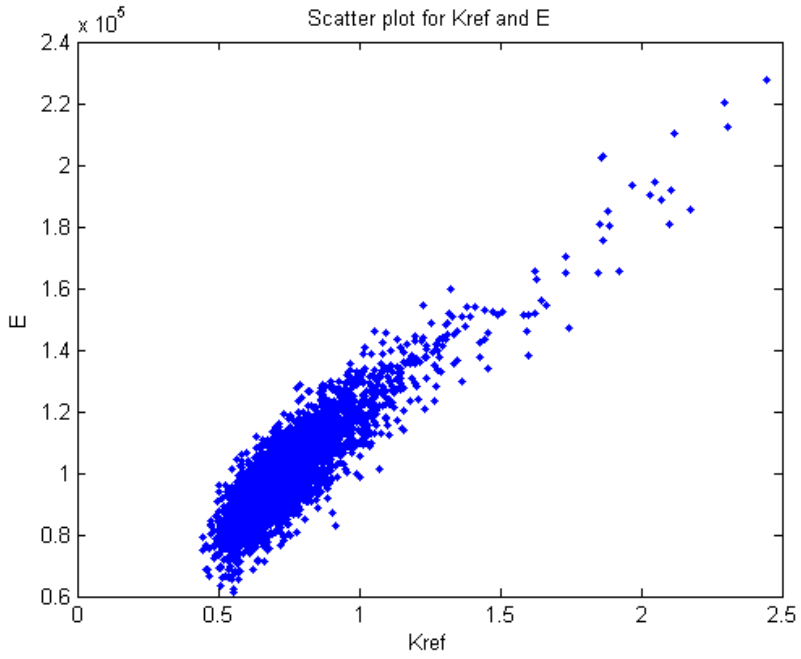

Figure 15: Scatter Plot of the parameters: From this plot, we observe that K_{ref} and E are positively correlated

.

The figure shows an elliptical shape which means there exists a positive correlation between the two parameters. We can also observe from the figure that the original values of the parameters are enclosed in the samples. This also shows that the approximation of the samples is relatively better.

Posterior Marginal Distributions

The posterior marginal distribution of the parameters summarizes the current state of knowledge about all the uncertain quantities (including unobservable parameters and also missing, latent, and unobserved potential data).

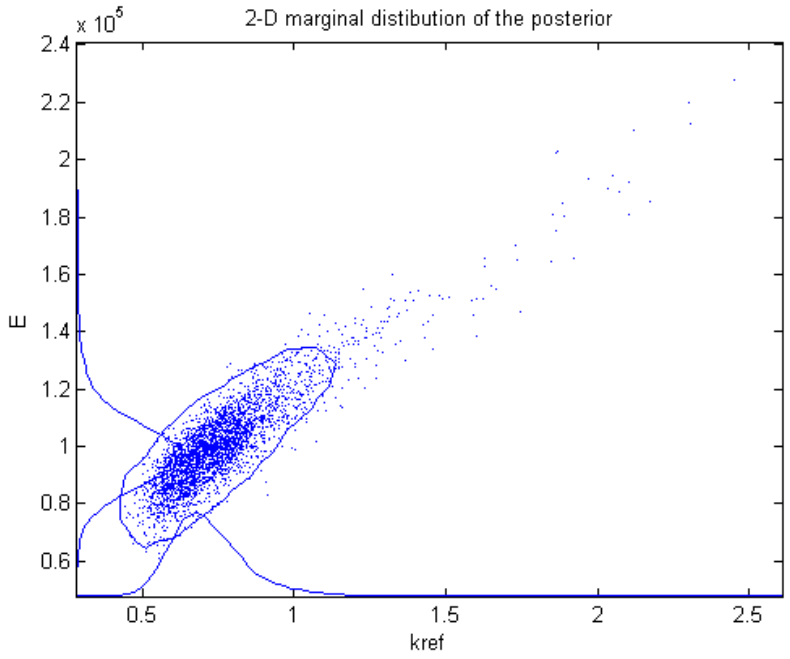

Figure 16: A scatter plot of a parameter pair with confidence regions based on kernel densities.

Figure 16 shows the pairwise marginal density distribution of the parameters based on kernel densities in our model. This is the marginal distribution of interest for our parameters K_{ref} and E.

Standard Error of the Mean Estimate

Figure 17 shows the error standard deviation of the samples. The standard error of the mean is also known as the Monte Carlo standard error (MCSE). The MCSE provides a measurement of the accuracy of the posterior estimates. The mean of the MCMC chain for the parameters is shown in Table 4.

	Kref	E
Original	0.7	100000
Optimized (FMINSEARCH)	0.66538	94187.9507
MCMC Chain Mean	0.685939	106860.1051

Table 4: MCMC Parameter values

The Monte Carlo Standard Error (MCSE) is an indication of how much error is in the estimate due to the fact that MCMC is used. As the number of iterations increases the MCSE approaches to zero. We can from the table above that, the MCMC chain

mean values for the parameters are closer to the true parameter values. This means that MCMC methods perfom better than the FMINSEARCH optimizer.

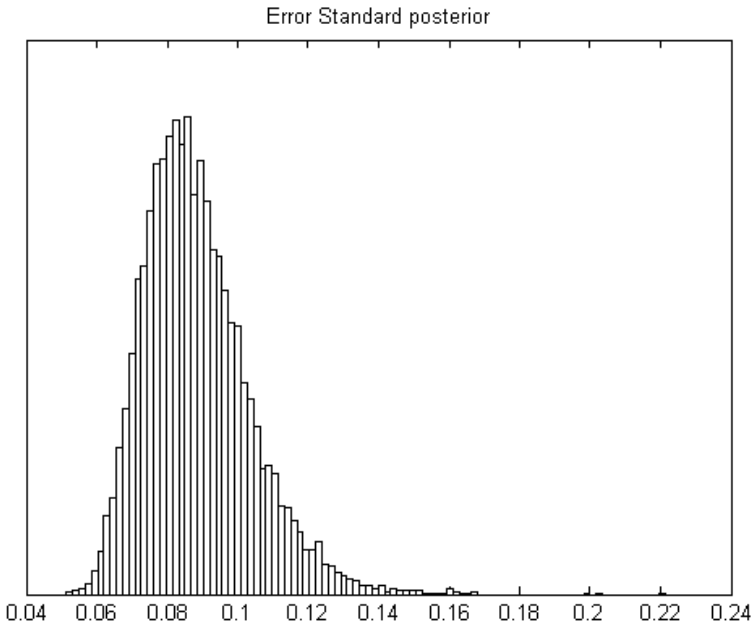

Figure 17: Error Standard Deviation of the Samples

Predictive Posterior Distributions

Figures 17 and 19, show the predictive posterior distributions of the parameters at two different temperatures $T = 20°\text{C}$ and $T = 40°\text{C}$ for the generated MCMC samples with respect to our model. We can observe that the parameters and sampling uncertainty are well captured.

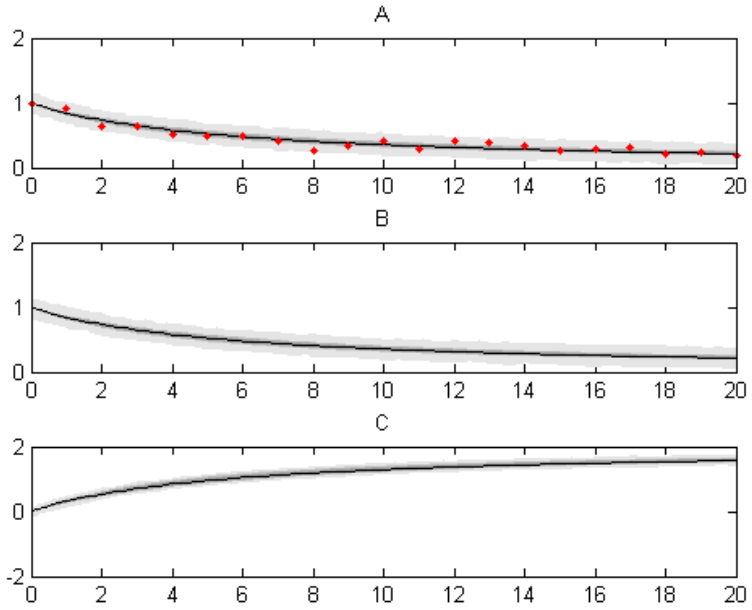

Figure 18: Simulated data and Predictive Distributions calculated from MCMC at $T = 20°C$

As discussed in the theoretical background above, the predictive posterior distribution is the distribution of future observations conditional on the observed data.

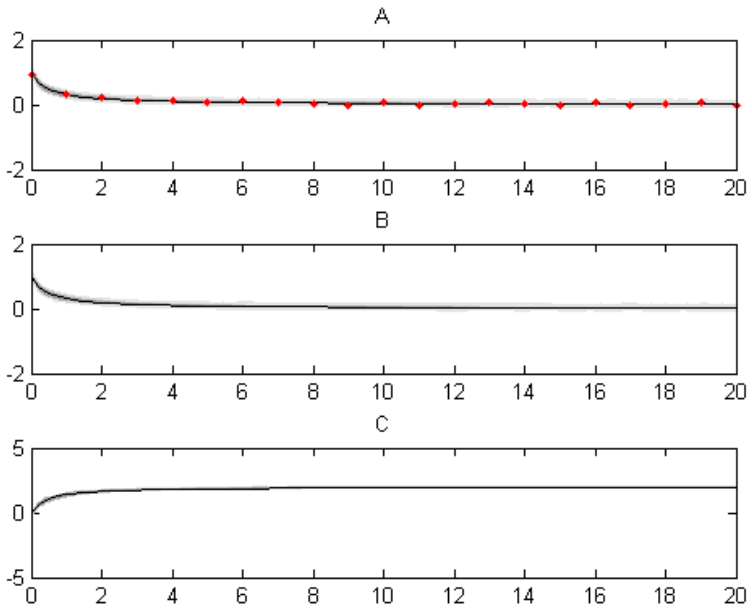

Figure 19: Simulated data and Predictive Distributions calculated from MCMC at $T = 40°C$ The gray colors in the plots correspond to 50%, 80% 95% and 99% confidence envelopes.

4.2 Optimization

Worst Case Criterion with 99% Certainty
As stated above, the aim of this optimization problem is to optimize the length of the pipe at which the production of C will be maximum or optimal. In this first method, we determine the time at which either of the compounds A or B vanishes such that there will be no more production of C. In our example, the compound B is chosen for this purpose. Given the velocity v, at which the compounds A and B are flowing through the pipe of length L, we can easily calculate the length of the pipe beyond which there will be no more production. This helps to determine how long should the pipe be for the reactants to vanish and hence avoiding the cost that would arise by making a very long pipe that will not be used. In our case we assume the reactants to flow at a constant velocity $v = 1$. The temperature is also fixed at 45°C

Using the predictive posterior distributions for the parameters at different temperatures, we can observe a critical(crit.)) concentration for the reactants A or B below which there will be no more production of C with a certain uncertainty. That is the reactant will have finished.

Optimization Task:
We minimize $x = L$, under condition $B(x, \theta) \leq crit$ with a given temperature T.

From the chain generated, we sample the values of B and determine the number of points that are accepted to be above the chosen critical value with 99% certainty. We choose the critical point value of 0.02 and determine the corresponding length at which the concentration of compound B is below this critical value. This length is found to be 23.03 as shown in Figure 20

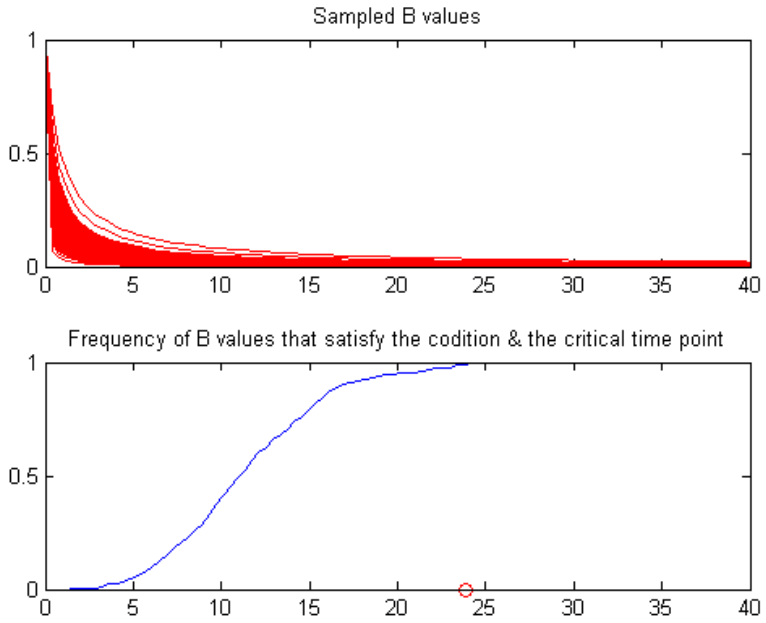

Figure 20: Time (a point in red circle) at which B has decreased under a critical value with 99% certainty at 45°C

At low temperatures, $T = 25$°C for example, the length of the pipe at which the concentration of the compound B goes below the critical point value is found to be bigger than at high temperatures. As we notice in the ODE solution, the reaction rate at high temperatures is high while the rate of reaction at low temperatures is also low, and hence it takes much time for the compound B to vanish. Depending on the temperature at which an experiment is performed, it is therefore possible to determine how should the length of the pipe be to get the maximum production of C at low cost in terms of constructing the pipe. The length of the pipe used in hot environment will be shorter than the pipe used in cold environment.

Sample Average Approximation

In this method[25], we choose few subsets of the generated samples and find the optimal length of the pipe to each subset. The optimal value of the objective function $f(L, \theta)$ where L is the required length of the pipe and θ is a vector of the MCMC generated samples can then be calculated by averaging the optimal solutions of the chosen subsets. According to the Law of Large Numbers, as the number of samples increases (approaches to infinity), the optimal solution to our problem using this method is approximately close to the true solution value of the optimization problem as discussed in the theory part.

In this problem, five subsets are chosen from the MCMC generated samples, each containing 500 samples. We also fix the temperature at 45°C. To each subset of samples

chosen, we choose a critical value of the concentration below which no more production of the component C or the component B vanishes. To demonstrate how this is done, we present one of the subsets chosen.

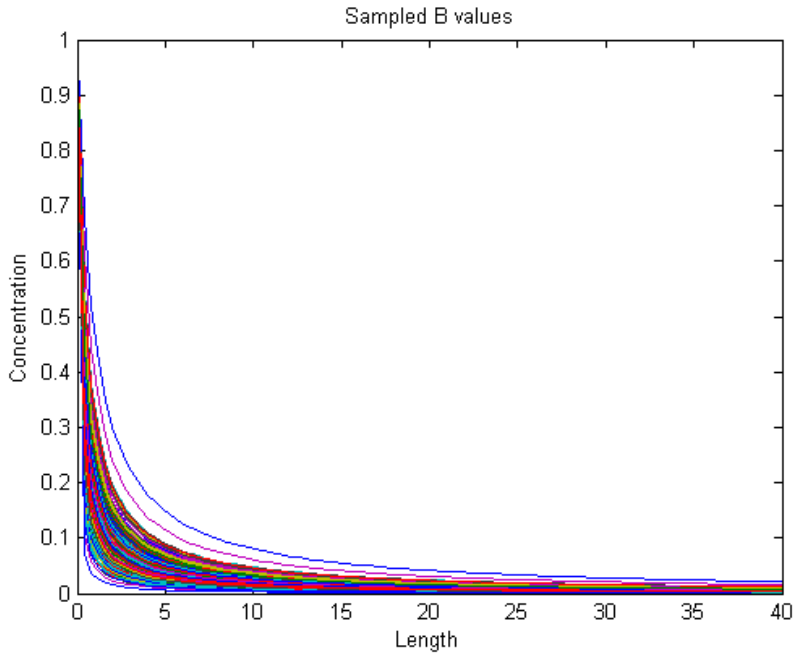

Figure 21: Sampled B values

Figure 21 shows the vectors of the sampled subset. To each vector values, we choose the critical point value below which the concentration of B vanishes with 99% certainty. We also choose this critical value to be 0.02 so as to make comparison with the first method.

In the first method of optimization above (Worst case criterion), we accepted all the points which are below the length corresponding to the critical value as shown in figure 20. In this second method of sample average approximation (SAA), we do the opposite way. That is, we accept the points below the critical value and compute the corresponding length values above the length corresponding to the critical value. After getting the length values, we then compute the minimum length value for each parameters vector and collect them in one vector. We plot the histogram of these minimum length values so as to observe their distribution.

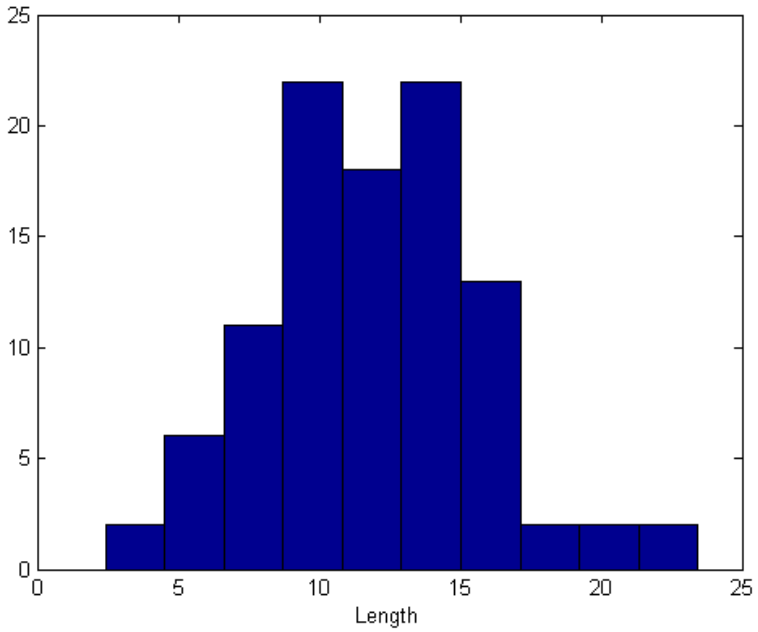

Figure 22: Histogram of the minimum length values

The histogram for the minimum length values for each sampled vector is shown in figure 22. This histogram shows that the minimum lengths are almost normally distributed. From the histogram, we also observe that this second method of optimization gives almost similar results as the first method. In the first method, the minimum length value of the pipe at which the component B vanishes, or no more production of C was calculated to be 23.84. In the second method of optimization, we can observe from the histogram that, the maximum length value at which the component B vanishes is around 23.43.

If we take different subsets of the generated MCMC samples and perform as described above to each subset, then the mean of the length values will approximately be close to the true solution of our problem as the number of samples approaches to infinity.

5 Conclusions

This thesis is about the use of MCMC methods for optimization of stochastic models under uncertainty. However, before learning how to optimize using MCMC methods, we first had a look on MCMC methods. MCMC methods have become very popular methods for sampling from different statistical distributions among statisticians and other researchers in recent years. This is mostly because of the shortcomings of the classical Bayesian inference, especially when sampling from complex or high-dimensional distributions. While a direct Bayesian inference requires the calculation of the normalizing constant in order to get the posterior distribution of the parameters (which is sometimes very tedious or impossible), MCMC only requires the prior information of the parameters and the likelihood evaluations.

With MCMC methods, it is possible to sample from different types of distributions, estimate the parameters involved in the model of interest and study their statistics such as posterior distributions. We can also use MCMC to study the future behavior of the model by plotting the predictive posterior distributions. The samples generated by MCMC methods can also be used for optimization processes. The uncertainties in the model can easily be taken into account because with MCMC it is possible to accept sample values that agree with the conditions set in the model.

When using MCMC methods in sampling however, one can face some challenges and difficulties in using the algorithms especially at the beginning. We have seen for example how important choosing the suitable proposal distribution is when using the Metropolis algorithm. The size of the proposal distribution in this case should neither be too small nor too large as this can lead to poor acceptance rate of the sample points and hence poor sample approximation. Choosing a suitable proposal distribution is not such easy. We need to learn as much as possible about the target distribution before applying MCMC algorithms so as get better convergence to the target distribution. In many cases, adaptive MCMC automatically 'tunes'the optimal proposal.

Some MCMC algorithms utilize much resources in terms of space and time. However due to the rapid increase in the speed of modern computers (for example, parallel computing) MCMC methods have not only become computationally feasible but also produce estimates in a reasonable time for realistic applied problems.

In the optimization of stochastic models under uncertainty, we see how well the MCMC methods perform better as compared to other deterministic methods. For example, in deterministic methods, the stopping criterion for determining whether the model parameters are sufficiently accurate is not linked directly to the accuracy of the performance objective computed by simulated model. Hence the accuracy of the model is assessed

based on guesswork. This is not the case in MCMC optimization methods, in which we can easily assess the accuracy of the model parameters by using some plots such as trace, function autocorrelation and posterior predictive distributions. The plots help to observe the convergence of the generated samples to the targete distribution, and thus we can be able to determine when to stop.

In this work, a single objective function with random variables is considered. We also considered the measurements with added noise. In this case, the MCMC methods have been proved to work well. The major question is still if the methods can still work better if the stochastic models with multiple objective functions are considered. We also need to see what happens if we add a noise to the objective function. We need to observe whether these methods can still work for optimizing such models.

References

[1] Solonen A, Haario H. Model-Based Process Optimization in the Presence of Parameter Uncertainty; Lappeenranta University of Technology.

[2] Paul J. A, The Monte-Carlo Method

[3] Pflug G. C, (1996) Optimization of Stochastic Models, *The Interface Between Simulation and Optimization*, Kluwer Academic Publishers, Boston/Dordrecht/London.

[4] Haario H. (2010) Statistical Analysis in Modelling: *MCMC methods Teaching Notes, Lappenranta University of Technology.*

[5] Cong H, Bradley P. C, (2000) MCMC Methods for Computing bayes Factors: *A comparative Review, University of minnesota.*

[6] Roberts G.O. and Tweedie R.L. (2008). Understanding MCMC [23-50].

[7] McShane E. J, (1974) Stochastic Calculus and Stochastic models, *A series of Monographs and textbooks*, Academic Press Inc. [1—10]

[8] Peter D. H., (2009) A First Course in Bayesian Statistical Methods, Springer [1-62]

[9] Kuo H. H, (2006) Introduction to Stochastic Integration, Springer

[10] Edwin K. P, Stanislaw H. Z, (2001) An Introduction to Optimization, John Wiley & Sons. Inc

[11] Gentle G, Härdle W, Mori Y. Handbook of Computational statistics, Springer Heidelerg

[12] Tomas B, (2009) Arbitrage Theoory in Contimuous Time, Oxford University Press

[13] David L. Ma, Richard B. Braatz Robust identification and control of batch processes

[14] Robin P. N, Rommert D. (2005) Automated Response Surface Methodology for Stochastic Optimization Models with uknown Variance

[15] M. Bernardo and Adrian F. M. Smith(1994) Bayesian Theory, John Wiley & Sons

[16] Solonen A, (2006) Monte Carlo Methods in Parameter Estimation of Nonlinear Models.

[17] Andrew Gelman and John B. Carlin and Hal S. Stern and Donald B. Rubin (2004) Bayesian Data Analysis, 2nd Edition, CRC/Chapman and Hall

[18] Siddhartha Chib; Edward Greenberg (1995) Understanding the Metropolis-Hasting Algorithm, *The American Staticisian Journal*, Vol 49, No. 4.

[19] Heikki H, Eero S and Johanna T, (2001) An adaptive Metropolis algorithm. Bernoulli (Vol. 7(2)). pp. 223-242. Accessed online as a pdf at: http://citeseer.ist.psu.edu/haario98adaptive.html

[20] Laine, M. (2008). Adaptive MCMC methods with applications in environmental and geophysical models. PhD thesis. Lappeenranta University of Technology.

[21] Heikki H., Laine M., Mira A.,and Saksman E. (2005) DRAM - efficient adaptive MCMC

[22] Scott, L. (2007). Introduction to Applied Bayesian Statistics and Estimation for Social Scientists, Springer Berlin Heidelberg NewYork [107-163]

[23] Brooks S.P. and Roberts G.O. (1999). Assessing convergence of Markov chain Monte Carlo algorithms,*Statistics and Computing*

[24] http:www.wikipedia.org

[25] Shapiro A.,Dentcheva D., Andrzej R. (2009) Lectures on Stochastic Programming, Modelling and Theory.

[26] Hui S. (2011) Posterior simulation, *Chapter* 11 *of Bayesian data analysis by Andrew Gelman*, Department of Mathematics and Statistics, Queen's University, Kingston

[27] Bayesian Statistics for Engineers, *Bayesian Inference, Bayesian Computation, Applications.* Accessed online as pdf at http://www2.isye.gatech.edu/ brani/isyebayes/bank/handout10.pdf

[28] Mark S. (2011) Computational Statistics with Matlab. Accessed online as pdf at http://psiexp.ss.uci.edu/research/teachingP205C/205C.pdf

[29] Guy L. (2010)The Law of Large Numbers and the Central Limit Theorem. Accessed online as pdf at http://www.cc.gatech.edu/ lebanon/notes/wllnAndClt.pdf

APPENDIX
Some Mathematical Definitions

Mean, Variance, Covariance and Correlation

The mean or expectation (μ) of a random variable X describes the most probable value of all possible values. If the random variable X is discrete, then the expectation is defined as

$$\mu = E(X) = \sum_i x_i p(x_i)$$

For the case of continuous random variable, the mean is defined as

$$\mu = E(X) = \int_{-\infty}^{\infty} dx$$

where p is the probability density function.

The variance of a random variable describes the variation and diffusion of the variable around its expectation. The variance for a discrete random variable with probability density function p is defined as

$$\sigma^2 = Var(X) = E(X - \mu^2) = \sum_i (x_i - \mu)^2 p(x_i)$$

For the continuous random variable, the variance is defined as

$$\sigma^2 = Var(X) = E(X - \mu^2) = \int_{-\infty}^{\infty} (x - \mu)^2 p(x) dx$$

If $x_n = (x_{1k}, \cdots, x_{nk})$ denotes the column vector in a design matrix (n observations for a variable), the sample variance can be calculated using

$$\sigma_{xk}^2 = Var(x_k) = \frac{1}{n-1} \sum_{i=1}^{n} (x_{ik} - \bar{x_k})^2$$

The square root of the variance is the standard deviation of a random variable

$$Std(x_k) = \sigma_{xk} = \sqrt{Var(x_k)}$$

If X_1 and X_2 are two random variables, $\mu_i = E(X_i)$ and $\sigma_i^2 = Var(X_i)$ are the mean and variance of X_i respectively. The *covariance* and *correlation* are measures of the

linear dependence between X_1 and X_2. In other words, the covariance and correlation indicate how well the relationship between X_1 and $X-1$ is described by the model

$$(X_1 - \mu_1) = \beta(X_2 - \mu_2) + \epsilon$$

where ϵ is a random variable with mean 0 that is independent of X_2. If, in fact, $(X_1 - \mu_1) = \beta(X_2 - \mu_2)$, then the model is perfect. On the other hand, if X_1 and X_2 are statistically independent, then $\beta = 0$ and the model is of no value. In general, a positive value of β indicates that $X-1$ and X_2 tend to be above or below their means together, while a negative value β indicates that they tend to be opposite sides of their means.

The covariance between X_1 and X_2 is defined to be

$$Cov(X_1, X_2) = E[(X_1 - \mu_1)(X_2 - \mu_2)] = E(X_1X_2) - \mu_1\mu_2$$

A value of $Cov(X_1, X_2) = 0$ implies $\beta = 0$ in our model of dependence, while $Cov(X_1, X_2) < 0 (> 0)$ implies $\beta < 0 (> 0)$. The covariance can take any value between $-\infty$ and ∞. The correlation standardizes the covariance to between -1 and 1:

$$\rho = corr(X_1, X_2) = \frac{Cov(X_1, X_2)}{\sigma_1\sigma_2}$$

Also, a value of $corr(X_1, X_2) = 0$ implies $\beta = 0$ in our model, while $corr(X_1, X_2) < 0 (> 0)$ inplies $\beta < 0 (> 0)$. The closer ρ is to -1 or 1 the stronger the linear relationship is between X_1 and X_2.

If for example we consider the linear model[16] with respect to the unknown parameter θ, it can be written in the form $y = X\theta + \epsilon$. The estimate $\hat{\theta}$ that minimizes the least square (LSQ) function, is the solution of the normal equation $X^TX\hat{\theta} = X^Ty$, which leads to

$$\hat{\theta} = (X^TX)^{-1}X^Ty.$$

Proof: If we assume n-dimensional vector space, y is the observation vector from the original to the observations. $X\hat{\theta}$ is the plane of all possible model values, thus the range of $X : R(X) = \{X\theta, \theta \in \mathbb{R}^n\}$.

The shortest distance from the model to the observations is perpendicular from y to $X\theta$. This perpendicular is defined as $y - X\hat{\theta}$, where $\hat{\theta}$ is the vector from the origin to the perpendicular base, $X\hat{\theta}$ belongs to the $X\theta$ plane.

Therefore $X\hat{\theta}$ and $(y - X\hat{\theta})$ are orthogonal to one another, that is $(y - X\hat{\theta}) \perp X\hat{\theta}$ and hence $(y - X\hat{\theta})(X\hat{\theta}) = 0$.

If we take the transpose on both sides, we get

$$\Big((y - X\hat{\theta})(X\hat{\theta})\Big)^T = \hat{\theta}^T X^T (y - X\hat{\theta}) = 0$$

$\hat{\theta}$ is a non-zero value in general case, thus
$X^T(y - X\hat{\theta}) = 0$, $X^T y = X^T X\hat{\theta}$, and therefore

$$\hat{\theta} = (X^T X)^{-1} X^T y.$$

For a random variable vector y with n components, and a fixed matrix A, it can be shown that

$$Cov(Ay) = ACov(y)A^T,$$

where A^T denotes the transpose of A.

Proof: By definition, $Cov(y) = E[(y - Ey)(y - Ey)^T]$. For Ay we have,

$$Cov(Ay) = E[(Ay - E(Ay))(Ay - E(Ay))^T]$$

Since matrix A is fixed, $EA = A$ and $E(Ay) = AEy$. This gives

$$\begin{aligned} Cov(Ay) &= E[(Ay - AEy)(Ay - AEy)^T] \\ &= E[(A(y - Ey))(A(y - Ey))^T] \\ &= E[A(y - Ey)(y - Ey)(y - Ey)^T A^T] \\ &= AE[(y - Ey)(y - Ey)^T]A^T \\ &= ACov(y)A^T \end{aligned}$$

Thus $Cov(Ay) = ACov(y)A^T$

If the measurement error $\varepsilon \sim N(0, \sigma^2 I)$ (independent and identically distributed (i.i.d.) components), where I is an identity matrix, we get $Cov(\hat{\theta}) = \sigma^2(X^T X)^{-1}$.

Proof[16]: Let $\hat{\theta}$ be the LSQ solution to the linear problem $y = X\theta + \epsilon$. That is, $\hat{\theta} = (X^T X)^{-1} X^T y$ (normal equation). Now $Cov(\hat{\theta}) = Cov((X^T X)^{-1} X^T y)$. Using the assumption that $Cov(\epsilon) = Cov(y) = \sigma^2 I$ and the fact that $Cov(Ay) = ACov(y)A^T$, $(AB)^T = B^T A^T$ and $(A^T)^{-1} = (A^{-1})^T$ ([8]) we get

$$\begin{aligned} Cov(\hat{\theta}) &= (X^T X)^{-1} X^T \sigma^2 ((X^T X)^{-1} X^T)^T \\ &= \sigma^2 (X^T X)^{-1} (X^T X)(X^T X)^{-1} \\ &= \sigma^2 (X^T X)^{-1}. \end{aligned}$$

Thus, $Cov(\hat{\theta}) = \sigma^2(X^T X)^{-1}$

Jacobian Matrix

The Jacobian matrix contains the partial derivatives of first order with respect to every function and every variable. If $y = y(x)$ is defined as a set of functions

$$\begin{aligned} y_1 &= y_1(x_1, x_2, \cdots, x_k) \\ y_2 &= y_2(x_1, x_2, \cdots, x_k) \\ \vdots &= \quad\quad \vdots \\ y_n &= y_n(x_1, x_2, \cdots, x_k) \end{aligned}$$

then the Jacobian matrix is defined as

$$\begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_k} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \frac{\partial y_n}{\partial x_2} & \cdots & \frac{\partial y_n}{\partial x_k} \end{pmatrix}$$

Confidence Intervals

The interval to which the true value of the estimated parameter θ belongs with probability $1-\alpha$ is the $(1-\alpha)\times 100\%$ *Confidence Interval*[16]. In order to form the confidence interval of the samples parameter θ, it assumed that the parameter follows some distribution D_θ. From D_θ we can assign the limits a and b so that

$$P(a \leq D_\theta \leq b) = 1 - \alpha$$

From the two inequalities $a \leq D_\theta \leq b$ we can compute the limits for the parameter θ:

$$L \leq \theta \leq U$$

where L and U are the lower and upper limits respectively.

For example if the random variable X is normally distributed ($X \sim N(\mu, \sigma^2)$), it is known that $(\bar{X}) \sim N(\mu, \frac{\sigma^2}{n})$. Then

$$Z = \frac{\bar{X} - \mu}{\sigma^2/\sqrt{n}} \sim N(0, 1).$$

The $1-\alpha$ confidence interval for μ, for example, can be calculated from the equation

$$-z_{1-\alpha/2} \leq Z \leq z_{1-\alpha/2}$$

where z_x represents the point of the cumulative density function (CDF) of $N(0,1)$ at which $P(X \leq z_x) = x$.

The Weak Law of Large Numbers

The weak law of large numbers[29] states that, if $X^{(1)}, X^{(2)}, \cdots$ is a sequence of $d-$dimensional i.i.d random vectors with finite expectation vector μ and covariance matrix $\sum$, then the sequence $Y^{(n)} = \bar{X} = \frac{1}{n}\sum_{i=1}^{n} X^{(i)}$ converges in probability to μ, that is

$$\lim_{n\to\infty} P\left(\left\|\frac{1}{n}\sum_{i=1}^{n} X^{(i)} - \mu\right\| \geq \epsilon\right) = 0, \quad \forall \epsilon > 0.$$

Proof: We apply the Chebyshev Inequality[29] which states that, for a scalar random variable X (with finite expectation and variance)

$$P\left(|X - E(X)| \geq a\right) \leq \frac{Var(X)}{a^2}, \quad \forall a > 0$$

Thus, for scalar random variables, $X^{(i)}(d = 1)$ with expectation μ and variance σ^2, the proof follows by noting that

$$\begin{aligned} E(\bar{X}) &= \frac{1}{n}\sum_{i=1}^{n} EX^{(i)} \\ &= (\frac{n}{n})\mu \\ &= \mu \end{aligned}$$

$$\begin{aligned} \text{Var}(\bar{X}) &= \frac{1}{n^2}\text{Var}\sum_{i=1}^{n} X^{(i)} \\ &= \frac{1}{n^2}\sum_{i=1}^{n} \text{Var}X^{(i)} \\ &= \frac{1}{n}\sigma^2 \quad (\text{since } X^{(i)} \text{ are i.i.d}). \end{aligned}$$

Applying Chebyshev Inequality to $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X^{(i)}$, we have

$$P\left(|\bar{X} - E\bar{X}| \geq \epsilon\right) = P(|\bar{X} - \mu| \geq \epsilon) \leq \frac{\sigma^2}{n\epsilon^2} \to_{n\to\infty} 0.$$

For $d > 1$, we apply Boole's Inequality[24] and the one dimiensional result above.

$$\begin{aligned} P(\|\bar{X} - E\bar{X}\| \geq \epsilon) &= P\Big(\sum_{j=1}^{d} |\bar{X}_j - \mu_j|^2 \geq \epsilon^2\Big) \\ &\leq \sum_{j=1}^{d} P\Big(|\bar{X}_j - \mu_j|^2 \geq \frac{\epsilon^2}{d}\Big) \\ &= \sum_{j=1}^{d} P\Big(|\bar{X}_j - \mu_j| \geq \frac{\epsilon}{\sqrt{d}}\Big) \\ &\leq \frac{d}{n\epsilon^2}\text{trace}(\text{Var}(X)) \to_{n\to\infty} 0 \end{aligned}$$