Technische Universität Berlin
Fakultät IV Elektrotechnik und Informatik
Institut für Softwaretechnik und Theoretische Informatik
Fachgebiet Methoden der Künstlichen Intelligenz

Diplomarbeit

Comparing Markov Chain Monte Carlo Proposal Densities for Diffusion Processes

Florian Stimberg
Matrikelnummer: 301138

Berlin, den 1. März 2010

Betreuer: Prof. Dr. Manfred Opper
Dr. Andreas Ruttor

Gutachter: Prof. Dr. Manfred Opper
Dr. Marc Toussaint

Eidesstattliche Erklärung

Die selbständige und eigenhändige Anfertigung versichert an Eides statt

Berlin, den 1. März 2010

Unterschrift

Zusammenfassung

Inferenz für diskret beobachtete Diffusionsprozesse ist seit einiger Zeit ein Forschungsschwerpunkt. Analytische Lösungen sind nur für einige Spezialfälle möglich, weshalb Samplingmethoden in den letzten Jahren sehr verbreitet für diese Art von Modellen sind.
In dieser Diplomarbeit wird ein Markov-Chain-Monte-Carlo-Ansatz auf Reaktionssysteme, die durch einen Diffusionsprozess approximiert werden, angewandt. Die Qualität von Markov-Chain-Monte-Carlo-Verfahren hängt stark von der Wahl der Vorschlagsdichte ab, deshalb konzentriert sich diese Diplomarbeit auf deren Einfluss. Neben einer modifizierten Diffusionsbrücke wird ein Variationsansatz benutzt um Vorschläge für den Pfad zu erzeugen. Simulationen von Modellen aus der Systembiologie werden zum Vergleich dieser Methoden benutzt und um zu bestimmen wie sie auf verschiedene Diskretisierungsgrade, die Stärke der Messabweichungen und weitere Simulationsparameter reagieren. Obwohl sie nicht allgemein anwendbar ist, zeigt die Variationsmethode gute Resultate für einige Modelle.
Zusätzlich wird eine Gauß'sche Zufallsbewegung und eine Maximum-Likelihood-Methode für Parametervorschläge benutzt und auf die Modelle angewandt um deren Effizienz zu bestimmen. Es stellt sich heraus, dass der Maximum-Likelihood-Ansatz für sich allein nicht als Vorschlagsdichte geeignet ist, deshalb werden in dieser Arbeit mögliche Verbesserungen diskutiert.

Abstract

Inference for discretely observed diffusion processes has been of interest for some time. Analytical solutions are only possible for special types of models and therefore sampling strategies haven recently been increasingly popular for models of this type.
In this thesis a Markov chain Monte Carlo approach is applied to reaction systems approximated by a diffusion process. The quality of Markov chain Monte Carlo methods depends heavily on the employed proposal distributions, thus their influence is the focus of this thesis. Besides a modified diffusion bridge, a variational approach to the inference problem is adapted as a path proposal. Simulations of system biology models are used to compare these methods and determine how they react to different values of observation noise and discretization among other simulation parameters. Although not generally applicable, the variational proposals proved to give good results for a number of models.
Additionally, a Gaussian random walk and a maximum likelihood approach are used for parameter proposals and applied on the models to test their performance. The maximum likelihood method is found to be not practical to be used on its own, therefore possible enhancements are discussed.

Contents

List of Figures

List of Tables

1. Introduction

1.1. General Introduction

Diffusion processes have been a focus of research in the past years, because they present an elegant way of describing many realistic systems coping with randomness. Their applications are manifold and range from financial analysis (see for example Merton, 1970; Jiang and Knight, 1997; Duffie et al., 2000; Eraker, 2001) to cell biology (Slepchenko et al., 2002; Chen et al., 2005; Manninen et al., 2006). Most of the models presented in this thesis originate from the latter field of research and involve the interaction of different chemical species. In the past these *reaction systems* have often been treated as deterministic, but this approach does not reflect the inherent noise of chemical interactions on a cellular level (McAdams and Arkin, 1999; Elowitz et al., 2002; Bar-Even et al., 2006). An algorithm for stochastic simulations of chemical reactions was presented by Gillespie (1977), but for the purpose of inference Golightly and Wilkinson (2005) proposed to approximate these reaction systems by continuous diffusion processes.

Under real conditions the state of a reaction system will only be observed at discrete times and these observations are generally not exact due to measurement errors. Inference for this kind of situation has two main goals: Inferring what the true state of the system is at and in-between the measurements and determining the parameters governing the reactions. Since analytical solutions of the stochastic differential equations (SDE) governing the diffusion process are seldom possible, other methods have been pursued[1]. These include importance sampling (Fearnhead, 2008), exact simulation (Beskos et al., 2006), particle filtering (Doucet et al., 2000), Hybrid Monte Carlo (Alexander et al., 2005) and variational inference (Archambeau et al., 2008; Ruttor et al., 2009). Markov Chain Monte Carlo (MCMC) methods together with data augmentation, based on the work of Tanner and Wong (1987), have been applied to discretely observed diffusion processes by Jones (1998) Eraker (2001) and Elerain et al. (2001) simultaneously. They use Gibb's sampling to draw samples from the posterior of the data and the reaction parameters and insert latent data in between the observations in order to have minimal approximation errors, when applying an Euler discretization to the SDEs.

The MCMC approach has the advantage over approximative methods that it

[1] For a good summary of the different approaches on diffusion processes with discrete-time observations see Sørensen (2004).

draws samples from the exact posterior. On the downside computational costs of MCMC methods have proven to be significantly higher in comparison to approximations. Additionally Roberts and Stramer (2001) highlighted that simple Gibb's sampling will generate increasingly correlated samples if the discretization becomes higher, so that the algorithm will effectively break down. To bypass this problem Golightly and Wilkinson (2006) use a Bayesian sequential filter, called simulation filter, which does not suffer from these problems while Eraker (2001) starts with low discretization and gradually raises it to counter the discretization bias.

An alternative to these approaches is the innovation scheme, first suggested by Chib et al. (2004) and further developed in Golightly and Wilkinson (2008). Instead of alternately sampling the parameters conditioned on the augmented data, and vice versa, the algorithm swaps between sampling the Brownian motion, which drives the diffusion, and the parameters. This helps overcome the dependence between the augmented data and parameter samples, which cause the algorithm to converge the slower the higher the discretization is chosen. Increasing the discretization actually enhances the algorithm's performance.

While theoretically the Metropolis-Hastings algorithm will generate samples from the target density with any proposal density, which can reach every state of the system from any other state, it is crucial to adapt the proposal density to the problem (Andrieu et al., 2003). Otherwise the proposals will be seldom accepted, or the samples will be highly correlated, leading to slow convergence of the Markov chain. Although often the main interest of inference is the posterior distribution of the parameters governing drift and diffusion, most works focus on improving the proposal of the time series data, while a simple Gaussian random walk is used as a parameter proposal (see e.g. Golightly and Wilkinson, 2009). Especially for models with many parameters, this can lead to high convergence times, because the proposals are not directed towards regions of high probability.

1.2. Outline

The goal of this thesis is to study the effect of different proposal densities on Markov Chain Monte Carlo methods used for Bayesian inference of reaction systems. Following Golightly and Wilkinson (2005) the reaction rate equations are approximated by a diffusion process governed by a set of coupled SDEs. For the parameter update the proposal densities examined were the Gaussian random walk on the logarithm used in Golightly and Wilkinson (2005) and a maximum likelihood approach of Ruttor et al. (2009). As path proposals the so-called modified diffusion bridge of Golightly and Wilkinson (2008) was employed as well as variational methods based on Ruttor et al. (2009). To circumvent correlation problems for high discretization, the innovation scheme of Golightly and Wilkinson (2009) is applied. For the simulations two auto-regulatory

gene networks were employed, a simple version with only two species taken from Opper and Sanguinetti (2008) and a more complex version of a prokaryotic gene network from Golightly and Wilkinson (2005). Additionally the Bicoid-model of Wu et al. (2007) and the double-well model[2] were simulated, but most comparisons were done on the Lotka-Volterra model, because it combines relatively low computational costs, with non-trivial behaviour.

Previous studies showed that the innovation scheme improves with higher discretization (Golightly and Wilkinson, 2009). To examine if the path proposal densities react differently to this, their performance was compared for different values of discretization. The amount of information the algorithm gets is measured by the mean number of reactions happening per observations and simulations were conducted to show how this effects the algorithms. In addition the strength of the measurement noise is another important factor determining the effectiveness of the inference, and therefore comparative studies have been made on this aspect. After results for the higher-dimensional models are presented, the computational costs of the path proposal distributions are compared.

For all models besides the double well model parameter inference was conducted, showing that maximum likelihood proposal does not approximate the true parameters well enough to be used as a pure proposal distribution. Possible enhancements to use this method for MCMC inference are discussed in the last chapter of this thesis.

The remainder of this thesis is structured as following. Chapter 2 explains basic concepts and methods applied in this thesis and in section 2.3 the models used for simulation are described. The algorithm used for MCMC inference is derived in chapter 3 followed by the presentation of proposal densities to update the system's state. First, in section 3.4 the proposal of Golightly and Wilkinson (2009) is described then in 3.5 different proposals based on the variational approach of Ruttor et al. (2009) are developed. Analogue to this, the following chapter 4 explains proposal densities for the system's parameters and addresses the problem of the inference algorithm degenerating for fine discretization in section 4.3. The results of the simulations are presented in chapter 5 and afterwards conclusions are drawn and the results discussed in chapter 6. Finally, the appendix goes into further details of the implementation and parameters used for the simulations.

[2] Unlike the other models presented here, the double-well model is not a reaction system, but a diffusion process and thus the diffusion approximation is unnecessary.

2. Background

This chapter explains methods, models and technical terms used in this thesis and is intended to give an ample overview over the topics in order to understand the following chapters. References for a more detailed description are given in the respective sections, but Bishop (2007); MacKay (2003) and Neal (1996) are especially recommended for a general introduction to Bayesian inference and the sampling methods presented in section 2.1. After this, reaction systems are introduced in section 2.2 and it is explained how Gillespie (1977) incorporated non-determinism into them. The chapter is completed with a description of the models used for simulation in this thesis.

2.1. Markov Chain Monte Carlo

While playing solitaire, Stan Ulam thought about the chances to get a successful card arrangement (Eckhardt, 1987). Analytical calculation turned out to be very tedious. Therefore Ulam tried to lay out the cards for several times and counted how many successful plays resulted. Metropolis and Ulam (1949) introduced this approach and called it the *Monte Carlo Method*.

The later called *Metropolis*-algorithm was proposed in Metropolis et al. (1953) and established the Markov Chain Monte Carlo method. While importance and rejection sampling both are impractical in high dimensions, MCMC is the only generally applicable algorithm, which is time efficient for complex problems, because the computation time is polynomial dependent on the number of dimensions (Andrieu et al., 2003). In many areas of application, multi-dimensional integrals over probability distributions need to be computed (e.g. expectations or normalisation constants in Bayesian statistics). MCMC methods create a Markov chain, which has the target distribution as its equilibrium distribution. Although in theory the Markov chain will generate samples from the exact distribution, when reaching equilibrium, it is difficult to determine if the chain has converged already (MacKay, 2003, p. 366).

2.1.1. Metropolis-Hastings algorithm

The idea of the Metropolis-Hastings algorithm is to have a proposal density $Q(\mathbf{x_a}|\mathbf{x_b})$, which is easy to compute. After generating an initial state $\mathbf{x}(0)$, we use this density in every step to draw a proposal state $\mathbf{x}^*$ conditioned on the last

state $\mathbf{x}(t_i)$. This proposal is accepted as the new state $\mathbf{x}(t_{i+1})$ with a probability α. If it is rejected, the state does not change so that $\mathbf{x}(t_{i+1}) = \mathbf{x}(t_i)$.

The algorithm of Metropolis et al. (1953) uses an symmetric distribution as proposal density. This means $Q(\mathbf{x_a}|\mathbf{x_b}) = Q(\mathbf{x_b}|\mathbf{x_a})$ for all pairs of $\mathbf{x_a}$ and $\mathbf{x_b}$. The proposal is accepted with probability

$$\alpha = \min\left(1, \frac{P(\mathbf{x}^*)}{P(\mathbf{x}(t))}\right). \tag{2.1}$$

It is important to highlight that in order to calculate α only rates of the target density $P(\mathbf{x})$ have to be evaluated. This means that it is unnecessary to know the normalization constant of $P(\mathbf{x})$.

Hastings (1970) generalized Metropolis' algorithm by dropping the necessity of a symmetric proposal density. With this change a new state $\mathbf{x}^*$ is accepted with probability

$$\alpha = \min\left(1, \frac{P(\mathbf{x}^*)Q(\mathbf{x}(t)|\mathbf{x}^*)}{P(\mathbf{x}(t))Q(\mathbf{x}^*|\mathbf{x}(t))}\right). \tag{2.2}$$

The goal of the algorithm is to generate a Markov chain which has the target density as its equilibrium distribution. Equilibrium distribution in this case means the chain leaves the density invariant and that, independent of the chosen initial state of the Markov chain, the distribution converges to the desired distribution $P(\mathbf{x})$ (Bishop, 2007, p. 540). The last criteria is called ergodicity. A distribution is invariant to a Markov chain if the transition probability does not change the distribution. If $T(\mathbf{x}^*|\mathbf{x})$ is the transition probability of the chain, then an invariant distribution $P(\mathbf{x})$ satisfies

$$P(\mathbf{x}) = \sum_{\mathbf{x}^*} T(\mathbf{x}^*|\mathbf{x})P(\mathbf{x})$$

This is particularly fulfilled if the distribution suffices detailed balance:

$$P(\mathbf{x})T(\mathbf{x}|\mathbf{x}^*) = P(\mathbf{x}^*)T(\mathbf{x}^*|\mathbf{x})$$

It can be shown (see e.g. Bishop, 2007, p. 541) that if we use the product of the proposal density $Q(\mathbf{x_a}|\mathbf{x_b})$ and the acceptance probability from equation (2.2) as transition probability of our Markov chain, the target density $P(\mathbf{x})$ fulfils detailed balance and thus is an invariant distribution of the chain. Note that it is still necessary that the proposal density satisfies ergodicity. It is easy to see that this is fulfilled if the distribution is greater than zero for every pair of states $\mathbf{x_a}, \mathbf{x_b}$, but this is not an essential requirement, since it would be sufficient if every state could be reached from any other other state indirectly.

2.1.2. Gibbs Sampling

Most probability distributions used in realistic models are multidimensional and it is often not practical to sample from the full distribution over all variables $P(\mathbf{x})$.

Gibbs sampling methods assume that sampling from the conditional distributions $P(x_i|\{x_j\}_{i\neq j})$ is mathematically tractable. To draw samples from $P(\mathbf{x})$ Gibbs sampling algorithms first initialize $\mathbf{x}$ and then iteratively samples each variable x_i from the conditional distribution. (MacKay, 2003, p. 371-372)

To understand that iteratively drawing samples from the conditionals generates samples from the complete distribution, it is best to see Gibbs sampling as an implementation of the Metropolis-Hastings method. For this let's consider drawing a sample for x_i as a Metropolis-Hastings update with $P(x_i|\{x_j\}_{i\neq j})$ as proposal density $Q(\mathbf{x}^*|\mathbf{x})$. This leads to the proposal being acceptance with probability (Bishop, 2007, p. 544)

$$\frac{P(\mathbf{x}^*)Q(\mathbf{x}|\mathbf{x}^*)}{P(\mathbf{x})Q(\mathbf{x}^*|\mathbf{x})} = \frac{P(x_i^*|\{x_j^*\}_{i\neq j})P(\{x_j^*\}_{i\neq j})P(x_i|\{x_j^*\}_{i\neq j})}{P(x_i|\{x_j\}_{i\neq j})P(\{x_j\}_{i\neq j})P(x_i^*|\{x_j\}_{i\neq j})} = 1$$

In the last step $\{x_j\}_{i\neq j} = \{x_j^*\}_{i\neq j}$ was used, because the proposal differs only in x_i from the former state. Thus, the proposal is always accepted and the Markov chain created by the Gibbs sampling algorithm has $P(x)$ as its equilibrium distribution.

Of course the conditionals have to fulfil the usual preconditions of a Metropolis-Hastings proposal density (see Section 2.1.1) to be able to draw samples from $P(\mathbf{x})$ using Gibbs sampling.

2.1.3. Hybrid Monte Carlo

Standard MCMC algorithms tend to display random walk behaviour (MacKay, 2003, p. 367), which can be problematic because it takes many iterations to explore the whole state space. Hybrid Monte Carlo (sometimes called Hamiltonian Monte Carlo) tries to avoid random walk behaviour and takes significantly less steps to reach a roughly independent state (see Bishop, 2007, p. 553ff). The algorithm was proposed by Duane et al. (1987) and combines the Metropolis-Hastings algorithm with stochastic dynamics.

If we wish to sample from a distribution $P(\mathbf{x})$ in the context of Hybrid Monte Carlo, the variables x_i represent the coordinates of particles. For every variable x_i a corresponding momentum variable p_i is introduced. The space of all possible values for $\mathbf{x}$ and $\mathbf{p}$ is called *phase space* and the canonical distribution over it is defined as

$$P(\mathbf{x}, \mathbf{p}) \propto \exp(-H(\mathbf{x}, \mathbf{p})). \tag{2.3}$$

$H(\mathbf{x}, \mathbf{p}) = E(\mathbf{x}) + K(\mathbf{p})$ is called the Hamiltonian function and in the physical interpretation it describes the total energy of the system consisting of the kinetic energy $K(\mathbf{p})$ and the potential energy $E(\mathbf{x})$. For our purpose it is important that $\mathbf{x}$ and $\mathbf{p}$ are independent in (2.3) and that any non-zero distribution $P(\mathbf{x})$ can be transformed into the marginal canonical distribution

$$P(\mathbf{x}) \propto \exp(-E(\mathbf{x})) \tag{2.4}$$

simply by defining the potential energy accordingly (Bishop, 2007, p. 549). Thus instead of sampling from $P(\mathbf{x})$ directly we can instead sample from $P(\mathbf{x}, \mathbf{p})$ and omit the values for $\mathbf{p}$.

Hybrid Monte Carlo alternates between a stochastic and a dynamical update. In the stochastic step, new values for the momentum variables $\mathbf{p}$ are sampled, while in the dynamical step $\mathbf{x}$ and $\mathbf{p}$ are updated according to Hamilton dynamics. Exact simulations of Hamilton dynamics would leave the Hamiltonian constant but for computational purposes an approximation with discrete time steps is used, which results in small changes in the total energy of the system (MacKay, 2003, p. 389). The bias introduced by this can be countered by rejecting new states with a probability depending on the change of the Hamiltonian.

It is easy to see that the dynamical updates alone would not be able to explore the entity of the phase space, thus not being ergodic. This is why the stochastic step is necessary, to explore regions in phase space, which have a different total energy.

Neal (1996) gives an excellent introduction to the Hybrid Monte Carlo algorithm.

2.2. Reaction Systems

To describe a chemical reaction system, which includes a set of chemical species and reactions, a set of ordinary differential equations can be used. These so-called reaction-rate equations describe the system as continuous and deterministic (Connors, 1990, p.114). On a cellular level, however, only comparatively small numbers of molecules are involved (McAdams and Arkin, 1999) and a continuous approach can only be seen as an approximation. Furthermore to be able to exactly predict the reactions (under the assumption of classical mechanics), the position and velocity of each molecule must be known. It is clear to see, as Gillespie (1977) pointed out that this is an unrealistic scenario and does not even take into account the fundamental indeterminism of quantum mechanics. It is furthermore assumed that the innate noise of chemical kinetics inside a cell play an important role in ensuring heterogeneity (Elowitz et al., 2002). These problems have led to a stochastic approach to chemical kinetics.

2.2.1. Stochastic Kinetics

While molecules move around randomly through Brownian motion a reaction $R : x_1 + x_2 \rightarrow x_3$ occurs if a molecule of type x_1 hits a molecule of type x_2. It is assumed that our system is kept well stirred, in a constant volume and is in thermal equilibrium. Gillespie (1992) showed that under these circumstances the chance of a reaction happening is constant. It is furthermore assumed that the law of mass action applies which means that the chance of a reaction of type R is

proportional to $x_1 x_2$, while here x_i represents the number of molecules of species x_i. Note, that this ambiguous notation is used throughout the rest of this thesis.

If we have a system with k species and r reactions we can represent the reactions in two $r \times k$ matrices U and V, where u_{ij} is the number of molecules of species j needed for reaction R_i to happen and v_{ij} represents how much molecules of species j are products of reaction R_i (Golightly and Wilkinson, 2005).

The hazard of reaction R_i is denoted by $h_i(X, c_i)$, where $X = (x_1, ..., x_k)'$ is the current number of molecules for each species and c_i is called the rate constant of the reaction. We follow Golightly and Wilkinson (2005) and only store one matrix $A = V - U$, called the *net effect reaction matrix*. Together with the hazard function this is sufficient to describe the reactions, because if there are not enough reactants for a reaction to happen, the hazard will be zero. For our purpose we try to solve the *chemical master equation* which describes the time evolution of the probability distribution over the system's states

$$\frac{\partial}{\partial t} P(X;t) = \sum_{i=1}^{r} \left(h_i(X - A_i, c_i) P(X - A_i; t) - h_i(X, c_i) P(X;t) \right) \tag{2.5}$$

where $P(X;t)$ is the probability that the system is in state $X = (x_1, x_2, \ldots, x_k)$ at time t. For a detailed derivation of the master equation see Gillespie (1992).

Gillespie algorithm

Gillespie (1977) introduced an algorithm which allows to sample from the master equation, which needs the hazard of *any* reaction happening. It is straightforward to see that this is just the sum of the individual hazard

$$h_0(X, \Theta) = \sum_{i=1}^{r} h_i(X, c_i) \tag{2.6}$$

where $\Theta = (c_1, \ldots, c_r)$. This means that the time to the next reaction is exponentially distributed with the parameter

$$\lambda = \exp(h_0(X, \Theta)) \tag{2.7}$$

The Gillespie algorithm first samples from (2.7) to decide when the next reaction happens and then it determines what type of reaction occurs. By using (2.6), it is easy to see that the probability of reaction i is

$$\frac{h_i(X, c_i)}{\sum_{i=1}^{r} h_i(X, c_i)} \tag{2.8}$$

It is notable that this algorithm truly generates continuous time samples and therefore gives exact samples from the master equation.

2.2.2. Markov Jump Process

A stochastic process in continuous time is called a *Markov jump process* (MJP) if it fulfils the Markov property and the state changes are discontinuous[1]. In the context of continuous time the Markov property means that the probability of the system being in state X at time t, conditional on all the previous states, depends only on the last state.

Process rates are used to describe Markov jump processes. If $f(X'|X)$ is the process rate of a MJP, a transition from state X to state X' happens with probability $\Delta t f(X'|X)$ during the infinitesimal time interval Δt. This means

$$p(Y'|Y) \approx \delta_{Y'Y} + \Delta t f(Y'|Y) \tag{2.9}$$

with $\delta_{Y'Y}$ being the Kronecker delta (Ruttor et al., 2009). The approximation is exact for $\Delta t \to 0$.

Like we did for reaction system (equation (2.5)) we can set up a master equation for the marginal probabilities:

$$\frac{\partial}{\partial t} P(X;t) = \sum_{X' \neq X} \left(P(X';t) f(X|X') - P(X;t) f(X'|X) \right). \tag{2.10}$$

As shown in Ruttor et al. (2009) every reaction system, as described in 2.2.1, is a special incarnation of a Markov jump process. To see this we use the net effect matrix A and the hazard vector $h(X)$ to define the process rates of the MJP:

$$f(X'|X) = \sum_{i=1}^{n} \delta_{X'-X, Ae_i} h_i(X) \tag{2.11}$$

with e_i being a vector of length n with one as the ith element and zero otherwise.

2.2.3. Diffusion Approximation

A solution to a stochastic differential equation of the form

$$\frac{\mathrm{d}X_t}{\mathrm{d}t} = b(t, X_t) + \sigma(t, X_t) W_t \tag{2.12}$$

with W_t being Brownian motion, is called a *diffusion process* (Øksendal, 1992). In the notation of the Itô calculus the SDE takes the form

$$\mathrm{d}X_t = b(t, X_t)\mathrm{d}t + \sigma(t, X_t)\mathrm{d}W_t \tag{2.13}$$

and $b(t, X_t)$ is referred to as the drift and $\sigma(t, X_t)$ is called the diffusion. In contrast to Markov jump processes, the changes of a diffusion process are continuous,

[1] For a formal definition see e.g. Ethier and Kurtz (1986, p. 162).

but like MJPs they satisfy the Markov property and therefore are often used as an approximation for MJPs.

For most cases, finding an exact solution to the master equation 2.5 is not tractable. As in Golightly and Wilkinson (2005) we use the Fokker-Planck equation as an continuous approximation of the chemical master equation. van Kampen (2007, p. 197 ff.) shows this can be done by a second-order Taylor expansion of the master equation, if we assume that $P(X,;t)$ changes slowly with X and the Markov process' jumps are small compared to the number of molecules.

For a process $X(t) = (X_1, \ldots, X_k)$ the Fokker-Planck equation is (Gardiner, 2009, p. 291)

$$\frac{\partial}{\partial t}P(X;t) = -\sum_{i=1}^{k}\frac{\partial}{\partial X_i}\left(\mu_i(X)P(X;t)\right) + \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k}\frac{\partial^2}{\partial X_i \partial X_j}\left(\beta_{ij}(X)P(X;t)\right) \tag{2.14}$$

with

$$\mu_i(X) = \lim_{\Delta t \to 0}\frac{1}{\Delta t}E\left[(X_i(t+\Delta t) - X_i(t))|X(t) = X\right] \tag{2.15}$$

and

$$\beta_{ij}(X) = \lim_{\Delta t \to 0}\frac{1}{\Delta t}Cov\left[(X_i(t+\Delta t) - X_i(t)), (X_j(t+\Delta t) - X_j(t))|X(t) = X\right] \tag{2.16}$$

for $i, j = 1, \ldots, k$.

Thus, as shown in (Gardiner, 2009, p. 138), the Itô diffusion for (2.14) is

$$\mathrm{d}X(t) = \mu(X)\mathrm{d}t + \beta^{\frac{1}{2}}(X)\mathrm{d}W(t), \tag{2.17}$$

where $\beta^{\frac{1}{2}}(X)$ is a matrix, which fulfils

$$\beta^{\frac{1}{2}}(X)(\beta^{\frac{1}{2}}(X))' = \beta(X). \tag{2.18}$$

Golightly and Wilkinson (2005) showed that for the case of reaction systems this leads to the stochastic differential equation

$$dY(t) = \mu(X,\Theta)dt + \beta^{\frac{1}{2}}(X,\Theta)dW(t) \tag{2.19}$$

where the drift and diffusion are

$$\mu(X,\Theta) = A'h(X,\Theta) \tag{2.20}$$

$$\beta(X,\Theta) = A'diag(h(X,\Theta))A \tag{2.21}$$

with A being the net effect reaction matrix, $h(X,\Theta) = (h_1(X,c_1), \ldots, h_r(X,c_r))'$ is the hazard and $\Theta = (c_1, \ldots, c_r)$ the parameter vector. We assume that (2.19) has a unique, non-exploding solution (Øksendal, 1992, p. 48ff.), which is reasonable assumption for chemical reaction systems.

For an example, see section 2.3.1, where the drift and diffusion of the Lotka-Volterra model are stated explicitly.

2.3. Models

The five models introduced in the following were used in this thesis to test and compare the performance of the different proposal distribution. All except the double well model, are reaction systems, on which the diffusion approximation, described in the last section, needs to be applied in order to use the MCMC algorithms presented in section 3. The double well model is defined by a stochastic differential equation and thus no further approximation is needed.

2.3.1. Lotka-Volterra Model

The Lotka-Volterra model is a simple but non-trivial system of non-linear differential equations. It was first described by Lotka (1924) and Volterra (1926) independently. In this thesis a slightly modified version of the Lotka-Volterra model described as a reaction system by Boys et al. (2008) is used. It consists of two species X_1,X_2 called prey and predator and the following four (instead of three in Boys et al., 2008) reactions:

$$R_1 : X_1 \xrightarrow{c_1} 2X_1 \tag{2.22}$$
$$R_2 : X_1 \xrightarrow{c_2} \emptyset \tag{2.23}$$
$$R_3 : X_2 \xrightarrow{c_3} 2X_2 \tag{2.24}$$
$$R_4 : X_2 \xrightarrow{c_4} \emptyset \tag{2.25}$$

representing reproduction and death of the species. It is important to know that the probability of (2.23) and (2.24) depends on X_1 and X_2. This is represented in the hazard-function of the model:

$$h_1(X, c_1) = c_1 X_1 \tag{2.26}$$
$$h_2(X, c_2) = c_2 X_1 X_2 \tag{2.27}$$
$$h_3(X, c_3) = c_3 X_1 X_2 \tag{2.28}$$
$$h_4(X, c_4) = c_4 X_2 \tag{2.29}$$

Accordingly the net effect reaction matrix for this model is

$$A^T = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} \tag{2.30}$$

Figure 2.1 shows an example of a simulation of the Lotka-Volterra model using the Gillespie algorithm.

If we apply the diffusion approximation described in section 2.2.3 to the Lotka-Volterra model, we can insert the net effect matrix from (2.30) and the hazards

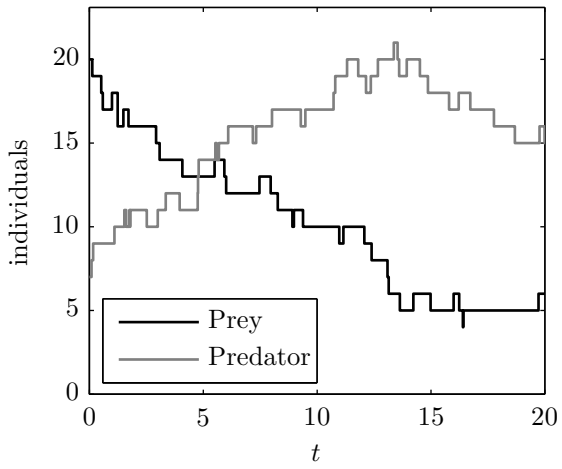

Figure 2.1.: Population dynamics for a Lotka-Volterra model simulated by the Gillespie algorithm. The rate parameters for the simulation were $c_1 = c_4 = 0.05$ and $c_2 = c_3 = 0.01$ and the simulation started with 20 prey and 7 predator individuals. The black line represents the prey population and the grey line the predator. Notice that in the beginning the prey population diminishes and the predator's grows until, at around $t = 14$, the trend is reversed.

from (2.26) to (2.29) into equation (2.20) and (2.21) to get the drift and diffusion for the Itô diffusion approximating the model

$$\mu(Y, \Theta) = \begin{pmatrix} c_1 X_1 - c_2 X_1 X_2 \\ c_3 X_1 X_2 - c_4 X_2 \end{pmatrix} \tag{2.31}$$

$$\beta(Y, \Theta) = \begin{pmatrix} c_1 X_1 + c_2 X_1 X_2 & 0 \\ 0 & c_3 X_1 X_2 + c_4 X_2 \end{pmatrix}. \tag{2.32}$$

These terms then will be used for the inference algorithm in section 3.

2.3.2. Auto-regulatory Gene Network Models

Two models of auto-regulatory gene networks are presented in this section. The first one specifies the production and downregulation of a protein through mRNA.

Protein Downregulation

In Ruttor et al. (2009) this simple model with only two chemical species is described. The first species is the mRNA, the second is a protein P. It includes four

reactions:

$$R_1 : \emptyset \xrightarrow{c_1} mRNA \tag{2.33}$$

$$R_2 : mRNA \xrightarrow{c_2} \emptyset \tag{2.34}$$

$$R_3 : \emptyset \xrightarrow{c_3} P \tag{2.35}$$

$$R_4 : P \xrightarrow{c_4} \emptyset. \tag{2.36}$$

Both species decay exponentially, i.e. the hazards of (2.34) and (2.36) are proportional to the number of mRNA and protein molecules respectively. New protein molecules are created my mRNA translation thus (2.35) has a hazard proportional to the number of mRNA molecules. The hazard of (2.33) is defined as

$$h_1(X, c_1) = c_1(1 - 0.99 \times H(P - P_c)) \tag{2.37}$$

with H representing the Heaviside step function[2]. This means mRNA production is constant until the number of protein molecules reaches a threshold P_c, then it is diminished drastically. P_c is provided to the algorithm, leaving only the parameters c_1, c_2, c_3 and c_4 to be learned.

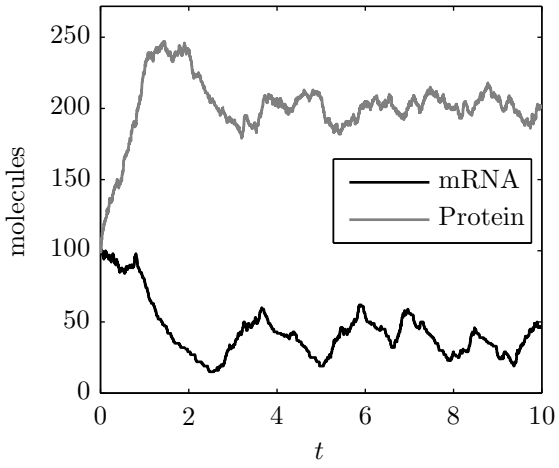

Figure 2.2.: Simulation of the protein downregulation model with rate constants $c_1 = 90, c_2 = 1, c_3 = 2.5, c_4 = 0.5$ and $P_c = 200$. The molecular dynamics were simulated with Gillespie's algorithm. Just before $t = 1$ the protein (grey line) surpasses the critical value of 200 molecules and the mRNA (black line) population drops immediately. After this the number of protein molecules oscillates around the threshold.

[2] $H(x) = 1$ if $x > 0$ and $H(x) = 0$ otherwise.

Prokaryotic Auto-regulatory Gene Network Model

A more complex model for a gene network in prokaryotic cells was introduced by Golightly and Wilkinson (2005). This model is based on a transcription factor P and his dimerized form P_2. The dimer inhibits the binding of the enzyme RNA-polymerase to the gene, which codes the corresponding monomer. When the enzyme binds to the beginning of the gene it travels along the DNA and synthesizes mRNA.

The ability of the transcription factor to transform into the repressing dimer is ensuring that the protein is not overproduced thus the network is auto-regulatory.

This gene network is characterized by eight reactions, which in most cases represent the overall effect of a process and should not be comprehended as the actual chemical reactions happening. The binding and unbinding of the dimerized transcription factor to the DNA is represented by (2.38) and (2.39), while the generation of RNA through the RNA-polymerase is summarized in (2.40). The complex process of protein synthesis is combined in the simple reaction (2.41). Reaction (2.42) and (2.43) describe the dimerization of the protein and the reverse reaction, while the last two reactions represent the process of degradation of RNA and the protein P.

$$R_1 : \mathrm{DNA} + \mathrm{P}_2 \xrightarrow{c_1} \mathrm{DNA} \cdot \mathrm{P}_2 \tag{2.38}$$

$$R_2 : \mathrm{DNA} \cdot \mathrm{P}_2 \xrightarrow{c_2} \mathrm{DNA} + \mathrm{P}_2 \tag{2.39}$$

$$R_3 : \mathrm{DNA} \xrightarrow{c_3} \mathrm{DNA} + \mathrm{RNA} \tag{2.40}$$

$$R_4 : \mathrm{RNA} \xrightarrow{c_4} \mathrm{RNA} + \mathrm{P} \tag{2.41}$$

$$R_5 : 2\mathrm{P} \xrightarrow{c_5} \mathrm{P}_2 \tag{2.42}$$

$$R_6 : \mathrm{P}_2 \xrightarrow{c_6} 2\mathrm{P} \tag{2.43}$$

$$R_7 : \mathrm{RNA} \xrightarrow{c_7} \emptyset \tag{2.44}$$

$$R_8 : \mathrm{P} \xrightarrow{c_8} \emptyset \tag{2.45}$$

The hazards of the second-order reactions (2.38) and (2.42) are $h_1(X, c_1) = c_1 \mathrm{DNAP}_2$ and $h_5(X, c_5) = c_5 \mathrm{P}(\mathrm{P} - 1)$ respectively, while the other reactions are all of first order and therefore straightforward.

Since the sum of DNA and DNA·P molecules remains constant throughout the whole process a conversation constant K is introduced such that

$$K = \mathrm{DNA} + \mathrm{DNA} \cdot \mathrm{P}. \tag{2.46}$$

which can be used to substitute $K - \mathrm{DNA}$ for $\mathrm{DNA} \cdot P$, leaving only 4 types of molecules, which need to be observed[3]. K is known by the algorithm for the sake of simplicity, but it could be learned without problems like the other parameters.

[3] This is necessary because otherwise the net effect reaction matrix A would not be of full rank, causing problems for the inference algorithm. See Golightly and Wilkinson (2005) for details.

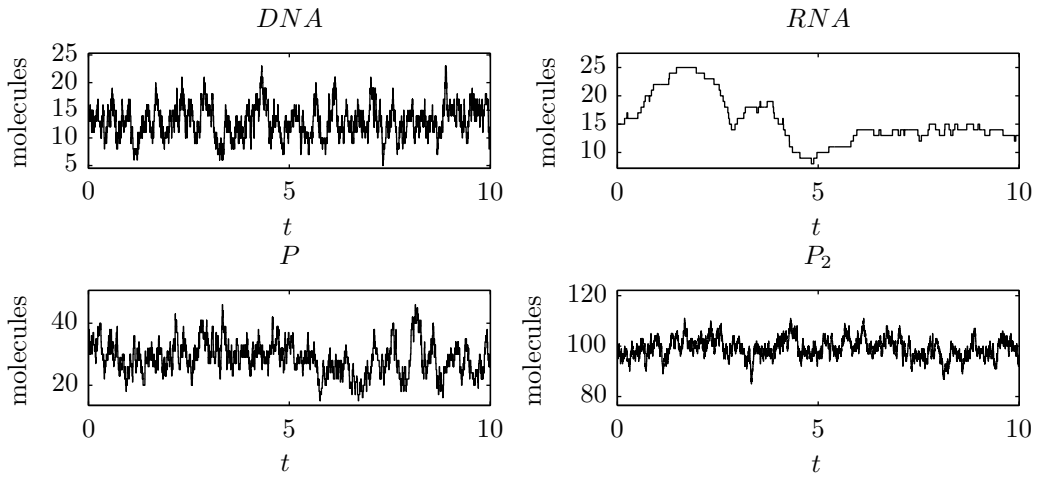

Figure 2.3.: The plot shows the time-series of the number of molecules for a simulation of the prokaryotic auto-regulatory gene network model with parameters $c_1 = 0.1, c_2 = 0.7, c_3 = 0.35, c_4 = 0.2, c_5 = 0.1, c_6 = 0.9, c_7 = 0.3, c_8 = 0.1$ and $K = 200$. Gillespie's algorithm was used to simulate the dynamics. The DNA·P population is not drawn because it depends deterministically on the number of DNA molecules. Notice that the fluctuation of the RNA population is very low in comparison to the other species, because its only interaction with the protein is reaction R_4, which leaves the number of RNA molecules invariant.

2.3.3. Bicoid Dynamics Model

Unlike the Lotka-Volterra and both auto-regulatory gene network models, in the bicoid dynamics model only one chemical species is involved. Instead of the change in number of molecules per species, we're interested in the spatial distribution of the molecules. The derivation of the chemical master equation in section 2.2.1 assumes the system to be well stirred, which means all reactants are equally distributed in space. Naturally this can't be expected if we focus on the spatial distribution of a single chemical species. A method to bypass this problem is described by Malek-Mansour and Houard (1979). The general idea is to divide the space into several spatial bins, such that we can assume the content of each bin to be approximately well stirred. The molecules in each of these bins are treated as a chemical species of their own and diffusion between adjacent bins is modelled as a chemical reaction in which a molecule of the source bin reacts to become one in the target bin.

The *Bicoid* protein is produced in the anterior region of *Drosophila melanogaster* embryos and is responsible for the development of head and thorax of the insect (Driever and Nüsslein-Volhard, 1988). This thesis uses the one-dimensional model introduced by Wu et al. (2007) in which the protein is produced in the first bin and diffuses through the embryo and decays at a constant rate on the

way. This means for k bins there will be $3k$ reactions:

$$R_1 : \emptyset \xrightarrow{k_1} x_1 \tag{2.47}$$

$$R_{i+1} : x_i \xrightarrow{d} x_{i+1} \; \forall i = 1, \ldots, k \tag{2.48}$$

$$R_{k+i} : x_{i+1} \xrightarrow{d} x_i \; \forall i = 1, \ldots, k-1 \tag{2.49}$$

$$R_{2k-1+j} : x_i \xrightarrow{k_2} \emptyset \; \forall j = 1, \ldots, k, \tag{2.50}$$

where x_i represents the number of bicoid molecules in the ith bin. Reaction (2.47) models the production of the bicoid protein in the most anterior bin. The $2k-2$ reactions (2.48) and (2.49) describe the diffusion to a neighbouring bin, while (2.50) is a decay reaction for each bin.

It is important to note that the number of reaction parameters is independent of the number of bins, if these are evenly spaced. In this case the diffusion reactions ((2.48) and (2.49)) all happen at a rate $d = \frac{D}{h}$, with D being the Diffusion rate and h the width of a bin. As stated above the decay rate also remains constant, thus there are only three reaction constants involved in this model: k_1, k_2 and d.

For this thesis, as in Dewar et al. (2009), 8 bins were used. An example simulation can be seen in figure 2.4.

2.3.4. Double Well Model

An example of a one-dimensional diffusion process is the double well model. It is defined by the following stochastic differential equation:

$$\mathrm{d}x(t) = \mu_\theta(x(t))\mathrm{d}t + \beta \mathrm{d}W_t \tag{2.51}$$

with the drift function

$$\mu_\theta(x) = 4x(\theta - x^2), \theta > 0. \tag{2.52}$$

The system has two stable states (called wells) at $x = -\theta$ and $x = \theta$ and an unstable equilibrium at $x = 0$ (Miller et al., 1994). The system noise allows the dynamical system to change between the wells, making its stationary distribution multi-modal. For the simulation in figure 2.5 θ was set to 1 and β to 0.5, which makes the process switch between the wells roughly every 2000 time steps on average (Miller et al., 1994), rendering the transition in the time-frame of the simulation very unlikely.

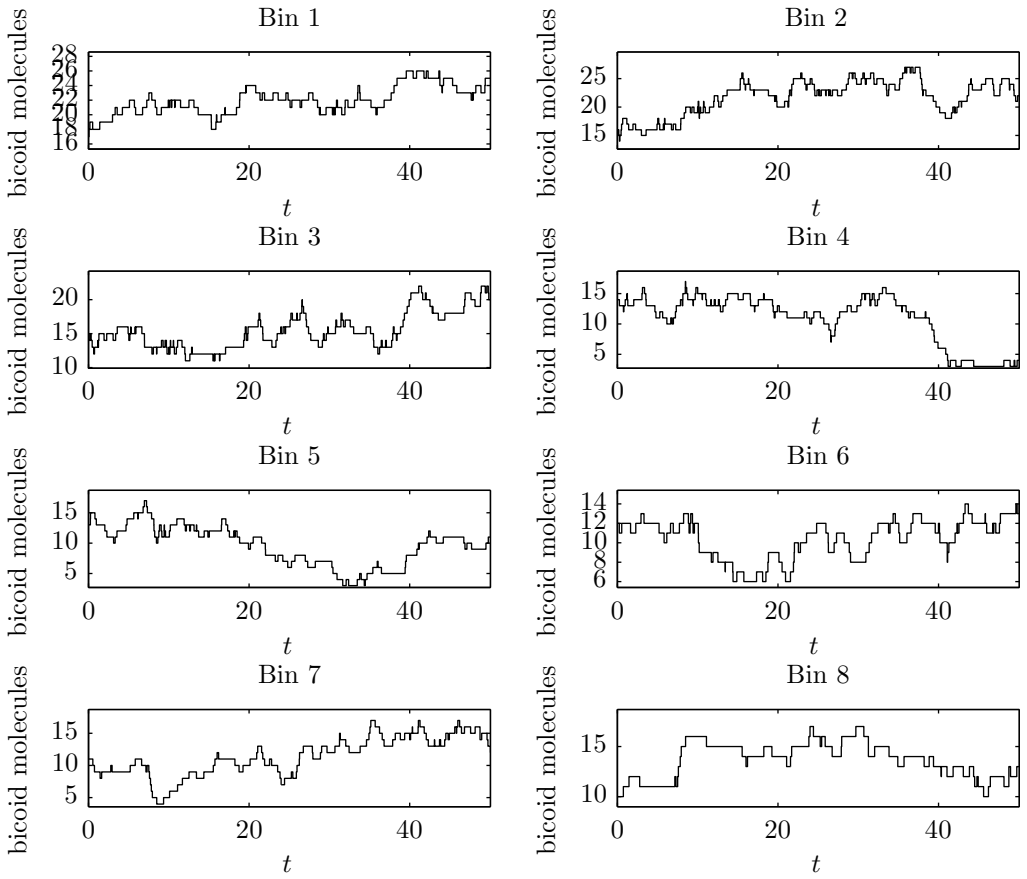

Figure 2.4.: Spatial distribution of molecules in a simulation of Bicoid dynamics with parameters $k_1 = 0.001, k_2 = 0.4$ and $d = 0.05$, simulated with the Gillespie algorithm. From bin 1 to bin 8 the starting values were 17,16,15,14,13,12,11 and 10. Bin 1 is the most anterior bin, where the bicoid protein is produced. At around $t = 40$ the population in bin 4 drops sharply, possibly because of a stochastical high diffusion to bin 3 and bin 5.

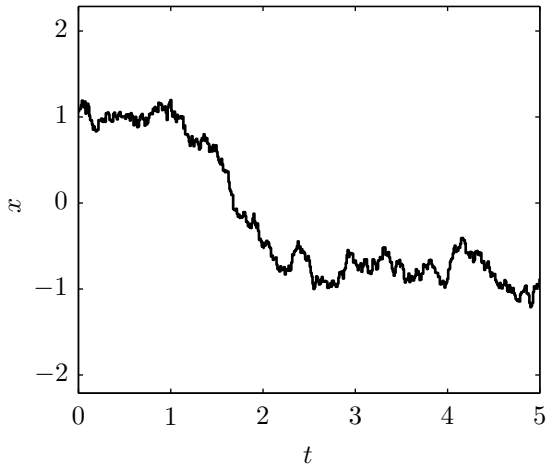

Figure 2.5.: Time series of a double well simulation with parameters $\theta = 1$ and a diffusion constant $\beta = 0.5$. The two wells at $x = \theta = 1$ and $x = -\theta = -1$ are well observable. Unlike for the reactions systems, an exact simulation is not possible. Instead the simulation was done by using an Euler discretization of the SDE (2.51) with a step-size of $\Delta t = 0.01$. The simulation ran until a swap into the second well was observed and only this section of the time-series is shown, since the average time for a well change to happen is roughly 2000 time steps.

3. Inference and Path Proposals

Given a set of observations we want to conduct inference on the path and the parameters Θ. This is done by approximating a solution for (2.19). Following the approach of Golightly and Wilkinson (2005) the first step is discretizing (2.19) by the Euler approximation

$$\Delta X(t) = \mu(X(t), \Theta)\Delta t + \beta^{\frac{1}{2}}(X(t), \Theta))\Delta W(t) \tag{3.1}$$

with $\Delta W(t) \sim N(0, I\Delta t)$. Naturally this introduces a discretization error, which can be reduced by introducing latent data points in-between observations. For the sake of simplicity we assume the observations are at evenly spaced time points and insert $m-1$ latent data points between two neighbouring observations. This method was proposed by Pedersen (1995) and allows to make the approximation arbitrary precise through an increase of m. If we have l observations this leads to $k(l-1)(m-1)$ missing values $Y_i(t_j)$, which need to be simulated. The simulated data together with the observations is called the *augmented data* Y as in Eraker (2001) and can be merged into a matrix

$$Y = \begin{pmatrix} X_1(t_0) & Y_1(t_1) & \dots & X_1(t_m) & Y_1(t_{m+1}) & \dots & X_1(t_n) \\ X_2(t_0) & Y_2(t_1) & \dots & X_2(t_m) & Y_2(t_{m+1}) & \dots & X_2(t_n) \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ X_k(t_0) & Y_k(t_1) & \dots & X_k(t_m) & Y_k(t_{m+1}) & \dots & X_k(t_n) \end{pmatrix} \tag{3.2}$$

where $n = lm$ and observations of the process appear at t_i with i being a multiple of m. Through the rest of this thesis, Y^i will denote the ith column of Y.

3.1. Exact Measurements

Even though this thesis does not cover any simulations without observation noise, we start by deriving the inference algorithm for this case as it was done by Golightly and Wilkinson (2005). Subsequent to this, the approach is extended to observations obscured by Gaussian noise in section 3.2.

If the observations are not altered by noise, they will remain unchanged during the whole simulation, leaving only the latent data in between the measurements to be sampled. In this case the joint posterior of the augmented data and the parameters is

$$\pi(Y, \Theta|D) \propto \pi(\Theta) \left[\prod_{i=0}^{n-1} \pi(Y^{i+1}|Y^i, \Theta) \right], \tag{3.3}$$

where Y^i denotes the ith column of Y, $\pi(\Theta)$ is the prior density over the parameter values and

$$\pi(Y^{i+1}|Y^i,\Theta) = |\beta_i^{-1}|^{\frac{1}{2}} \exp\left\{-\frac{1}{2}(\Delta Y^i - \mu_i\Delta t)'(\Delta t\beta_i)^{-1}(\Delta Y^i - \mu_i\Delta t)\right\} \quad (3.4)$$

with $\Delta Y^i = Y^{i+1} - Y^i$, $\mu_i = \mu(Y^i,\Theta)$, $\beta_i = \beta(Y^i,\Theta)$ (Golightly and Wilkinson, 2005). Note that a prior distribution over the starting values of the simulated path Y^0 is not necessary, because we have a noisefree observation at time t_0.

In most cases the posterior (3.3) will be high-dimensional and therefore naive sampling strategies will not be successful. For cases like this Tanner and Wong (1987) suggested a Gibbs-Sampling algorithm, which alternately draws samples from the latent data conditional given the parameters and observations and the parameter conditional given the augmented data. Since our diffusion models are highly non-linear, we cannot sample directly from the conditionals (Golightly and Wilkinson, 2008) and therefore an Metropolis-Hastings step is used to sample from both conditionals.

The conditionals to sample from are:

$$\pi(Y^i|Y^{i-1},Y^{i+1},\Theta,D) \propto \pi(Y^i|Y^{i-1},\Theta)\pi(Y^{i+1}|Y^i,\Theta) \quad (3.5)$$

for $i = 1,\ldots,n-1$

$$\pi(\Theta|Y,D) \propto \pi(\Theta)\left[\prod_{i=0}^{n-1}\pi(Y^{i+1}|Y^i,\Theta)\right] \quad (3.6)$$

Golightly and Wilkinson (2005) proposes the following algorithm for this case:

1. Set starting values for the parameters and initialize the latent data by linear interpolating between the adjacent observations.
2. For $i = 1,\ldots,n-1$ with i not being a multiple of m use a Metropolis-Hastings step to draw Y^i using a Gaussian density with mean $(Y^{i-1} + Y^{i+1})/2$ and variance $\frac{1}{2}\Delta t\beta(Y^{i-1},\Theta)$ as proposal.
3. Use a Metropolis-Hastings step to draw a new Θ with a Gaussian random walk on $\log(\Theta)$ as proposal. This is done to ensure that all proposals are positive, which is necessary for the rate constants.
4. If enough cycles have been performed stop, otherwise return to step 2.

3.2. Noisy Measurements

To account for measurement errors, now the real process $X(t)$ is not observed directly, instead we observe $D(t) \sim X(t) + N(0,\Sigma)$, with $\Sigma = diag(\sigma_i^2)$. It is

noteworthy that the algorithm could easily learn the observation noise covariance from the data, together with the rate constants, but this would require more iterations and hence we abstain from doing so in the scope of this thesis. Additionally the noise covariance was always chosen to be a diagonal matrix, although this is not a constraint of the inference method.

With the additional noise term the posterior over the path and parameters becomes (Golightly and Wilkinson, 2005):

$$\pi(Y, \Theta|D) \propto \pi(\Theta)\pi(Y^0) \left[\prod_{i=0}^{n-1} \pi(Y^{i+1}|Y^i, \Theta)\right] \left[\prod_{i\in\{0,m,\ldots,n\}} \pi(D^i|Y^i, \Theta)\right] \tag{3.7}$$

where $D = \{D^1, D^m, \ldots, D^n\}$ are the noisy measurements and $\pi(Y^0)$ is the prior over the starting values of the path.

For the parameter values we sample from the following conditional

$$\pi(\Theta|Y, D) \propto \pi(\Theta) \left[\prod_{i=0}^{n-1} \pi(Y^{i+1}|Y^i, \Theta)\right] \left[\prod_{i\in\{0,m,\ldots,n\}} \pi(D^i|Y^i, \Theta)\right]. \tag{3.8}$$

When sampling the latent process conditional on the current parameter values it is now also needed to sample on times where observations were made. Therefore, we have to distinguish four different cases:

1. for Y^i, with i not being a multiple of m the full conditional is

$$\pi(Y^i|Y^{i-1}, Y^{i+1}, \Theta, D) \propto \pi(Y^i|Y^{i-1}, \Theta)\pi(Y^{i+1}|Y^i, \Theta) \tag{3.9}$$

2. for Y^i, with i being a multiple of m, but not 0 or n

$$\pi(Y^i|Y^{i-1}, Y^{i+1}, \Theta, D) \propto \pi(Y^i|Y^{i-1}, \Theta)\pi(Y^{i+1}|Y^i, \Theta)\pi(D^i|Y^i, \Theta) \tag{3.10}$$

3. for Y^0

$$\pi(Y^0|Y^1, \Theta, D) \propto \pi(Y^0)\pi(Y^1|Y^0, \Theta)\pi(D^0|Y^0, \Theta) \tag{3.11}$$

4. for Y^n

$$\pi(Y^n|Y^{n-1}, \Theta, D) \propto \pi(Y^n|Y^{n-1}, \Theta)\pi(D^n|Y^n, \Theta) \tag{3.12}$$

These cases are employed in step 2 of the algorithm in section 3.1 to update the latent data. Again the proposal is drawn from a simple Gaussian density centred on the mean of both adjacent data points. For the first and last measurement a Gaussian random walk on the current values is chosen as proposal distribution.

3.3. Updating the Path

The preceding section introduced a simple method to update the latent process, usable for the noise-free (section 3.1) as well as noisy observations (section 3.2). Updating only one column of Y at a time is called single-site updating and can be sufficient for simple models. The problem of this algorithm (highlighted by Elerian et al., 1998) is the rather high correlation between the old and new states. Because of this the state space is explored slowly and many samples need to be discarded to receive independent samples. An alternative to this method can be to update the whole state in one step (see section 3.5.3) but this often leads to high rejection rates, because of the high-dimensionality (Elerain et al., 2001). As a compromise Golightly and Wilkinson (2009) use block updating. The latent space is updated in overlapping blocks of size $2m-1$ with an observation at the start, end and in the middle. The values of the first and last point of the block are held fixed, while for the rest a new proposal is computed. This proposal is accepted or rejected according to the Metropolis-Hastings algorithm (section 2.1.1).

For updating the block around the measurement at position M the posterior to sample from is

$$\pi(Y^{M^-+1},\dots,Y^{M^+-1}|Y^{M^-},D^M,Y^{M^+},\Theta) \propto \pi(Y^M|D^M,\Theta)\prod_{i=M^-}^{M^+-1}\pi(Y^{i+1}|Y^i,\Theta). \tag{3.13}$$

with $M^- = M-m$ and $M^+ = M+m$.

Figure 3.1 shows an example of a block update.

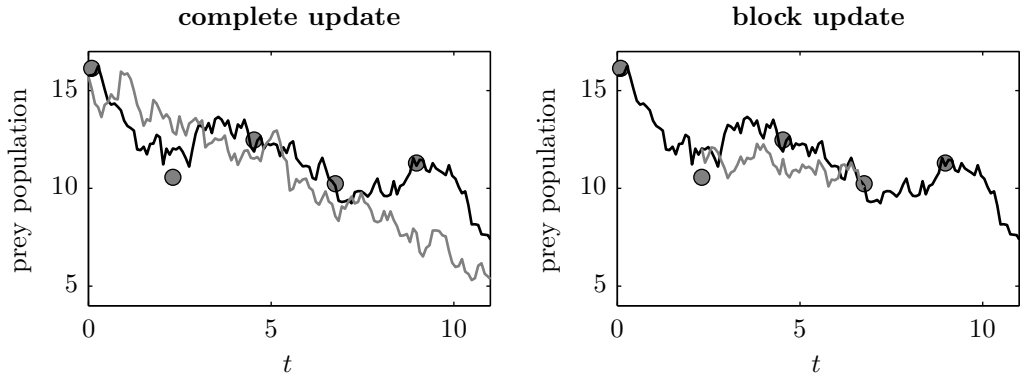

Figure 3.1.: An example illustrating the methods used to update the path. The black lines show the current path, grey lines represent the current proposal and the observations are drawn as circles. The left figure shows a complete update, only conditioned on the observations. The block proposal in the right figure spans only three measurements, starts like the current path and is conditioned to join the former path at the last measurement.

3.4. Modified Diffusion Bridge

For updating the latent process Golightly and Wilkinson (2009) use a two-part proposal. The first part is a Gaussian approximation, which is only conditioned on the observation in the middle of the block D^M. The second part is drawn from a modified diffusion bridge, which is based on the modified Brownian bridge by Durham and Gallant (2001). The first part proposal, is derived from the joint posterior of Y^i+1 and the observation D^M:

$$\begin{pmatrix} Y^{i+1} \\ D^M \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} Y^i + \mu_i \Delta t \\ Y^i + \mu_i \Delta^- \end{pmatrix}, \begin{pmatrix} \beta_i \Delta t & \beta_i \Delta t \\ \beta_i \Delta t & \beta_i \Delta^- + \Sigma \end{pmatrix} \right) \tag{3.14}$$

with $\Delta^- = t_M - t_i$.

If we condition this on the measurement D^M we get (Golightly and Wilkinson, 2009)

$$\widetilde{p}(Y^{i+1}|Y^i, D^M, \Theta) = \mathcal{N}(Y^{i+1}; Y^i + \widetilde{\mu}(Y^i, \Theta)\Delta t, \widetilde{\beta}(Y^i, \Theta)\Delta t) \tag{3.15}$$

with

$$\widetilde{\mu}(Y^i, \Theta) = \mu_i + \beta_i(\beta_i \Delta^- + \Sigma)^{-1}(D^M - (Y^i + \mu_i \Delta^-)) \tag{3.16}$$

$$\widetilde{\beta}(Y^i, \Theta) = \beta_i - \beta_i(\beta_i \Delta^- + \Sigma)^{-1}\beta_i \Delta t \tag{3.17}$$

The modified diffusion bridge of Golightly and Wilkinson (2009) is simpler as its drift is just a straight line to the fixed end point updated after drawing each point. The diffusion is just (2.21) multiplied with a factor which declines linear with the time to the end point. This means we sample the second part of the block by drawing points from the following distribution:

$$p^*(Y^{i+1}|Y^i, Y^{M^+}, \Theta) = \mathcal{N}(Y^{i+1}; Y^i + \mu^*(Y^i)\Delta t, \beta^*(Y^i, \Theta)\Delta t) \tag{3.18}$$

where

$$\mu^*(Y^i) = \frac{Y^{M^+} - Y^i}{t_{M^+} - t_i} \tag{3.19}$$

$$\beta^*(Y^i, \Theta) = \frac{t_{M^+} - t_{i+1}}{t_{M^+} - t_i}\beta_i \tag{3.20}$$

3.5. Variational Approach

A more complex approach is to approximate the posterior process by a Markov jump process Q with a time dependent rate function $g_t(x'|x)$. Ruttor et al. (2009) use a variational approximation for finding a good estimate of this rate function. The resulting process can be used as a proposal for either blocks of the path data (see 3.5.2) or the whole path (3.5.3).

The *Kullback-Leibler* (KL) divergence is a measure to compare two probability distributions P and Q and was first described in Kullback and Leibler (1951). For two discrete probability distributions P and Q it is defined as:

$$KL(Q||P) = \sum_i Q(i) \log \frac{Q(i)}{P(i)} \tag{3.21}$$

For our purpose it is necessary to know that $KL(Q,P) \geq 0$ with $KL(Q,P) = 0$ if and only if $P = Q$. This means if we set P to the posterior $\pi(Y, \Theta|D)$ from (3.7) we can use $KL(Q, \pi(Y, \Theta|D))$ as a measure of our approximated posterior process Q.

$$KL(Q||P_{post})) = \log Z + KL\left(Q||P_{prior}\right) - \sum_{i \in \{0,m,\ldots,n\}} \pi(D^i|Y^i, \Theta) \tag{3.22}$$

with

$$P_{prior} = \pi(Y^0) \prod_{i=0}^{n-1} \pi(Y^{i+1}|Y^i, \Theta) \tag{3.23}$$

and

$$P_{post} = \pi(Y, \Theta|D) = \frac{1}{Z} P_{prior} \prod_{i \in \{0,m,\ldots,n\}} \pi(D^i|Y^i, \Theta) \tag{3.24}$$

The posterior process can be described as a Markov jump process with a rate function $g_t(x'|x)$ (Ruttor et al., 2009). The KL divergence between two Markov jump processes is (Opper and Sanguinetti, 2008)

$$KL(Q||P) = \sum_Y Q(Y) \log \frac{Q(Y)}{P(Y)} \tag{3.25}$$

$$= \sum_{i=0}^{n-1} \sum_{Y^i} Q(Y^i) \sum_{Y^{i+1}} Q(Y^{i+1}|Y^i) \log \frac{Q(Y^{i+1}|Y^i)}{P(Y^{i+1}|Y^i)} + KL^0 \tag{3.26}$$

with

$$KL^0 = \sum_{Y^0} Q(Y^0) \log \frac{Q(Y^0)}{P(Y^0)} \tag{3.27}$$

and Y is a path of length n.

If we assume that the marginal probabilities at time $t(0)$ are fixed, we can set (3.27) to zero. We let $n \to \infty$ (i.e. $\Delta t \to 0$) to get

$$KL(Q||P_{prior}) = \int_0^T \sum_Y q(Y,t) \sum_{Y':Y' \neq Y} \left(g_t(Y'|Y) \log \frac{g_t(Y'|Y)}{\mu(Y'|Y)} + \mu(Y'|Y) - g_t(Y'|Y) \right) \mathrm{d}t \tag{3.28}$$

with $f(Y'|Y)$ being the rate function of P_{prior}. Now we can minimize the KL divergence in (3.22) with regard to $g_t(Y'|Y)$ and $Q(Y,t)$. By doing this, $g_t(Y'|Y)$ becomes the rate function of the posterior process P_{post}. During the optimization it needs to be taken into account that the rate function and the marginal probability depend on each other through the master equation (2.10). These constraints can be incorporated into the KL divergence with a Lagrange multiplier $\lambda(Y,t)$. This yields the Lagrangian:

$$L = KL(Q||P_{post}) - \int_0^T \left(\sum_Y \lambda(Y,t) \left(\frac{\partial}{\partial t} Q(Y,t) - \sum_{Y' \neq Y} \left(g_t(Y|Y')Q(Y',t) - g_t(Y'|Y)Q(Y,t) \right) \right) \right) \mathrm{d}t \tag{3.29}$$

As described in Ruttor et al. (2009) we derive the Lagrangian with respect to $g_t(Y'|Y)$ to get

$$\frac{g_t(Y|Y)}{f(Y'|Y)} = \frac{r(Y',t)}{r(Y,t)} \tag{3.30}$$

where $r(Y,t) = \exp(-\lambda(Y,t))$. This result combined with the derivative with respect to the marginal probabilities $Q(Y,t)$ form the linear backward differential equation (Ruttor et al., 2009)

$$\frac{\partial}{\partial t} r(Y,t) = \sum_{Y' \neq Y} \mu(Y'|Y) \left(r(Y,t) - r(Y,t) \right) \tag{3.31}$$

with jump conditions around the measurements

$$\lim_{t \to t_i^-} r(Y,t) = \pi(D^i|Y^i) \lim_{t \to t_l^+} r(Y,t). \tag{3.32}$$

.

3.5.1. Weak Noise Approximation

If the number of states in the backward equation (3.31) is limited it becomes a finite set of linear equations and therefore could be solved analytically. For s possible states there would be s^k equations (k being the dimensionality of the path) and it is easy to see that even for a relative small number of states this task becomes intractable. For this reason a weak noise approximation is used, which assumes that the diffusion noise of the system is low in comparison to number of molecules of each species. This prerequisite is induced into (3.31) by replacing Y' with $Y + \epsilon(Y' - Y)$ and using a second order Taylor expansion around ϵ, which leads to

$$\left(\frac{\partial}{\partial t} + \epsilon f(Y)^T \nabla + \frac{1}{2} \epsilon^2 (tr(D(Y) \nabla \nabla^T) \right) r(Y,t) = 0 \tag{3.33}$$

with $\mu(Y)$ and $D(Y)$ being the drift vector and diffusion matrix of (2.20) and (2.21) respectively.

Further we set $Y = b(t) + \epsilon z$ and $r(Y,t) = \psi(z,t)$, representing the assumption that there is a standard path $b(t)$, which Y only slightly differs from. It is obvious to set $\frac{db}{dt} = \mu(b(t))$, because the standard path will evolve according to the drift.

Again a second-order expansion around ϵ is done as required by vanKampen's system size expansion (van Kampen, 1961). This leads to

$$\left(\frac{\partial}{\partial t} + z^T J(Y)^T(b(t))\nabla + \frac{1}{2} tr(D(b(t))\nabla\nabla^T)\right)\psi(z,t) = 0, \tag{3.34}$$

where ϵ was set to one and with

$$J(Y) = \begin{pmatrix} \frac{\partial \mu_1}{\partial y_1} & \cdots & \frac{\partial \mu_1}{\partial y_k} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mu_k}{\partial y_1} & \cdots & \frac{\partial \mu_k}{\partial y_k}. \end{pmatrix} \tag{3.35}$$

Ruttor et al. (2009) solve this equation by

$$r(Y,t) \approx \eta(t) \exp\left(-\frac{1}{2}(Y - b(t))^T B^{-1}(t)(Y - b(t))\right) \tag{3.36}$$

with

$$\frac{dB}{dt} = J(b(t))B(t) + B(t)(b(t)^T - D(b(t))) \tag{3.37}$$

and

$$\frac{d\eta}{dt} = \eta(t) tr(J(b(t))). \tag{3.38}$$

To incorporate the observations into the solution we let b and B jump at each observation according to

$$B(t_i) = \left(B'(t_i)^{-1} + \Sigma^{-1}\right)^{-1} \tag{3.39}$$

$$b(t_i) = B(t_i)\left(B(t_i)^{-1} b'(t_i) + \Sigma^{-1} D^i\right) \tag{3.40}$$

for $i \in \{0, m, \ldots, n\}$ and with $B'(t_i)$ and $b'(t_i)$ being the values of b and B, if they had evolved without the jump at t_i.

We use this solution to compute the posterior drift by applying (3.30) to get

$$g(Y,t) = \sum_{Y' \neq Y} (Y' - Y)\frac{r(Y',t)}{r(Y,t)}\mu(Y|Y) \tag{3.41}$$

and as described in Ruttor et al. (2009) we can use a weak noise expansion of the posterior master equation and first order Taylor expansion of $r(Y,t)$ around Y to give the final approximation of the posterior drift:

$$g(Y,t) \approx \mu(Y) - D(b(t))B^{-1}(t)(Y - b(t)). \tag{3.42}$$

3.5.2. Block Update

Now we can use the posterior drift (3.42) for a proposal density. Analogue to the approach of Golightly and Wilkinson (2009) described in section 3.4 we update the path in overlapping blocks of size $2m-1$. As input for updating the block around the Measurement D^M the algorithm gets the current values of the path at the start and end of the block (Y^{M^-} and Y^{M^+}), which are fixed, and the measurement itself with the given observation noise covariance matrix Σ. With this $B(t)$ and $b(t)$ are calculated in order to compute $g(Y,t)$ according to (3.42). Thus the proposal block $Y^{M^-}+1,\ldots,Y^{M^+}-1$ is drawn from the following density:

$$p^V(Y^{i+1}|Y^{M^-},Y^i,Y^{M^+},D^M,\Theta)=\mathcal{N}(Y^{i+1};Y^i+g(Y^i,t_i)\Delta t,\beta(Y^i,\Theta)\Delta t) \quad (3.43)$$

with $\beta(Y^i,\Theta)$ being the original diffusion from (2.21).

3.5.3. Complete Update

One disadvantage of the update strategy presented in the last section, is its high computational cost and that only one observation at a time influences the proposal density.

To use the benefits of the variational approach to its full extend we now do not update the path in blocks, but compute a completely new proposal path and either accept or reject it entirely.

This means that $B(t)$ and $b(t)$ are computed with the full set of observations D and the corresponding observation noise covariance Σ. Then a proposal path is drawn from

$$p^V(Y^{i+1}|Y^i,D,\Theta)=\mathcal{N}(Y^{i+1};Y^i+g(Y^i,t_i)\Delta t,\beta(Y^i,\Theta)\Delta t). \quad (3.44)$$

In contrast to the previous proposal densities, the approach used here is independent of the previous path, because only the measurements are used to compute $B(t)$ and $b(t)$, where in the block update approach of section 3.5.2 the old values of the block's boundary points are used. This means that multiple updates of the path could be done in parallel and if the parameters Θ are held fixed, the backward equation would only need to be solved once. Especially the parallel approach could benefit from the current advances of multi-core processors and grid computing.

The proposal's independence from the previous path however has one drawback. The algorithm in section 3.1 initializes the path with a simple linear interpolation between the observations. This works well, if the proposal takes the former path into account. On the other hand, the proposal described here will likely be considerably different from the initial path. It is easy to see

that $Q(x(t_0)|x^*)/Q(x^*|x(t_0))$, from the acceptance probability of the Metropolis-Hastings algorithm (2.2), will be very low in most cases, because the linear interpolation is unlikely to be drawn from p^V. Simulations have shown that the acceptance probability is so low that the algorithm rejects thousands of proposals before the initial path is updated. After this the acceptance rises to a normal level. Since the choice of the linear interpolation as a start value was arbitrary, we instead use a normal proposal from (3.44) as the starting point of this algorithm.

3.5.4. Discrete Solution

The previous approaches tried to solve a continuous stochastic differential equation, to get the drift of the posterior process. Since the path is only sampled at discrete times, it might be better to work with a discretized version of equation (3.31). For this we assume that the marginal probabilities $r_t(Y)$ and the transition probabilities over a time interval $\Delta t > 0$ are Gaussian with

$$r_t(Y) \sim \mathcal{N}(Y; b(t), B(t)) \tag{3.45}$$

$$p_{\Delta t}(Y'|Y) \sim \mathcal{N}(Y'; Y + \Delta t\mu(Y, \Theta), \Delta t\beta(Y, \Theta)). \tag{3.46}$$

For the sake of clarity we drop the dependence of μ and β on Y and Θ in the next steps. Thus the backward equation becomes

$$\begin{aligned} r_{t-\Delta t}(Y) &= \int r_t(Y')p_{\Delta t}(Y'|Y)dY' \\ &= \int \left[\frac{\exp\left(-\frac{1}{2}(Y' - b(t))^T B^{-1}(t)(Y' - b(t))\right)}{\sqrt{\det(2\pi B(t))}} \right. \\ &\quad \left. \cdot \frac{\exp\left(-\frac{1}{2}(Y' - (Y + \Delta t\mu))^T (\Delta t\beta)^{-1}(Y' - (Y + \Delta t\mu))\right)}{\sqrt{\det(2\pi\Delta t\beta)}} \right] \mathrm{d}Y' \\ &\overset{1}{=} \int \left[Z \frac{\exp\left(-\frac{1}{2}(Y' - m)^T C^{-1}(Y' - m)\right)}{\sqrt{\det(2\pi K)}} \right] \mathrm{d}Y' \end{aligned} \tag{3.47}$$

with

$$m = (B^{-1}(t) + (\Delta t\beta)^{-1})^{-1}(B^{-1}(t)b(t) + (\Delta t\beta)^{-1}(Y + \Delta t\mu)) \tag{3.48}$$

$$C = (B^{-1}(t) + (\Delta t\beta)^{-1})^{-1} \tag{3.49}$$

$$Z = \frac{\exp\left(-\frac{1}{2}(Y + \Delta t\mu - b(t))^T (B(t) + \Delta t\beta)^{-1}(Y + \Delta t\mu - b(t))\right)}{\sqrt{\det(2\pi(B(t) + \Delta t\beta))}} \tag{3.50}$$

[1] (Petersen and Pedersen, 2008, p. 41)

Since Z does not depend on Y' we can write

$$\begin{aligned} r_{t-\Delta t}(Y) &= \int \left[Z \frac{\exp\left(-\frac{1}{2}(Y'-m)^T C^{-1}(Y'-m)\right)}{\sqrt{\det(2\pi K)}} \right] \mathrm{d}Y' \\ &= Z \int \left[\frac{\exp\left(-\frac{1}{2}(Y'-m)^T C^{-1}(Y'-m)\right)}{\sqrt{\det(2\pi K)}} \right] \mathrm{d}Y' \\ &= Z \end{aligned} \tag{3.51}$$

where in the last step the integral vanishes, because the integrand is a normalized Gaussian in Y'.

So in summary we've shown

$$r_{t-\Delta t}(Y) \sim \mathcal{N}(Y; b(t) - \Delta t\mu, \Delta t\beta + B(t)). \tag{3.52}$$

it is important to notice that this distribution is not Gaussian in Y, since μ and β depend on Y, too. We apply Laplace's method (see e.g. Bishop, 2007, p. 213 ff.) to approximate $r_{t-\Delta t}(Y)$ by a Gaussian around the maximum Y^{max}, i.e. we get

$$r_{t-\Delta t}(Y) \approx \mathcal{N}(Y; m(t-\Delta t), B(t-\Delta t)) \tag{3.53}$$

with $m(t-\Delta t) = Y^{max}$ and $B(t-\Delta t) = H^{-1}$, where H is the Hesse matrix of $\ln(r_{t-\Delta t}(Y^{max}))$ from (3.52). Since the derivatives of $r_{t-\Delta t}(Y)$ become very complicated we use numerical methods to gain the Hessian and Y^{max}.

With this approximation we now compute $B(t)$ and $b(t)$ backwards through time, while adjusting to the measurements as described in (3.39). Using the posterior drift function of (3.42) we can sample the same way as in section 3.5.2 and 3.5.3. For the same reasons as in section 3.5.3 we use a sample drawn from the proposal density to initialize the augmented data.

4. Parameter Proposal

The previous chapter dealt with possible proposal densities for the path, while in this chapter two approaches for updating the reaction parameters are presented.

4.1. Gaussian Random Walk on the Logarithm

This approach was used by Golightly and Wilkinson (2009) and is fairly simple. A multivariate Gaussian with a constant covariance matrix is centred around the logarithm of the current parameters values. A sample θ^*_{ln} is drawn from this density and $\exp(\theta^*_{ln})$ is used as a proposal. The logarithm is used because reaction parameters need to be non-negative and the exponential is strictly positive. This means we draw a proposal from the following density:

$$Q_g(\theta^*|\theta) = \mathcal{N}(\ln(\theta^*); \ln(\theta), C_p) \tag{4.1}$$

with C_p being a predefined, constant covariance matrix. The proposal is then accepted with probability given by (2.2).

4.2. Free Energy Minimization

In order to get a good approximation of the parameters of the model we use the *free energy* as a measure. The free energy F can be expressed in terms of the partition function Z as $F = -\ln(Z)$ (MacKay, 2003). We want to minimize the free energy, which means a maximization of $\ln(Z) = \ln(P(D))$, the log-likelihood of the data. As shown by Ruttor et al. (2009), we can compute the log-likelihood through

$$\ln(Z) = \ln(\int r(Y, t_0)\mathrm{d}Y) = \frac{k}{2}\ln(2\pi) + \frac{1}{2}\ln(\det(B(t_0)) + \ln(\eta(t_0)) \tag{4.2}$$

To use this as a proposal density we apply a Laplace approximation. First a gradient descent method is used to find the minimum of the free energy with respect to the parameters θ. Then the Hessian $H_{\theta_{min}}$ of the logarithm around the minimum is computed numerically. As for the Gaussian random walk in the last section, we update the logarithms of the parameters, thus all proposals are non-negative. In summary we draw new parameters $\theta*$ from

$$Q_f(\theta^*) = \mathcal{N}(\ln(\theta^*); \ln(\theta_{min}), H_{\theta_{min}}), \tag{4.3}$$

with θ_{min} as the minimum of the free energy (thus the maximum of (4.2)) and $H_{\theta_{min}}$ the Hessian of the free energy at θ_{min}. As explained before, the proposal is accepted with probability given by the Metropolis-Hastings algorithm in (2.2).

An important difference to the simple Gaussian random walk on the logarithm of section 4.1 is that, as we did for the path in 3.5.3, the proposal density does not depend on the previous value of the parameters. This should significantly lower the correlation between the samples[1], but on the other hand the success depends highly on the initial approximation and the thereby obtained values of θ_{min} and $H_{\theta_{min}}$.

Since it would take many iterations until the first proposal is accepted, as described in section 3.5.3 for the path update, we use a sample from Q_f to initialize the parameters.

4.3. Decorrelating State and Parameter Updates

A problem of the proposed approach is that for high discretization (i.e. large values of m) the system's diffusion can be obtained through the quadratic variation of Y. This was highlighted by Roberts and Stramer (2001) and renders the algorithm reducable for $m \to \infty$. For low values of m this is not important (see Golightly and Wilkinson, 2009) but as m increases, so does the autocorrelation and therefore the algorithm's mixing time.

Roberts and Stramer (2001) furthermore proposed a method to overcome the dependency between the latent data and the diffusion coefficient of the system by finding a specific non-centered parametrization of the latent data. While this generates an algorithm without any convergence issues even for $m \to \infty$, finding such a parametrization for multivariate diffusion models is impossible in most cases as Darren J. Wilkinson showed in the discussion of Papaspiliopoulos et al. (2003).

4.3.1. Innovation Scheme

Another approach was proposed by Golightly and Wilkinson (2008). Inspired by the *innovation scheme* of Chib et al. (2004), a sampler is used, which in one step samples the Brownian motion (W in (3.1)) instead of the augmented data $\hat{Y}$. In the other step the algorithm samples from the parameter posterior given the Brownian motion.

[1] At first glance one would assume that the samples would be completely uncorrelated. This would be the case if we simply drew the samples from the proposal density, but through the Metropolis-Hastings acceptance probability the previous value still effects the new value. If the proposal is rejected, the old value stays and therefore low acceptance rates can lead to a high autocorrelation. In section 5.2 these problems are investigated more elaborately.

As for the augmented data $\hat{Y}$ we define $\hat{W} = (W^0, \ldots, W^n)$ as the path of the Brownian motion, which drives the system's diffusion. It is obvious that from equation (3.1) $\hat{W}$ can be calculated deterministically from $\hat{Y}$ and vice versa, if Θ is fixed. This can be done by using the following algorithm:

1. Set starting values for the parameters Θ and the augmented data $\hat{Y}$
2. Update the augmented data by drawing a sample from $\pi(\hat{Y}|\Theta, D)$, thereby generating a new Brownian sample path $\hat{W}$.
3. Sample a new Θ by drawing from $\pi(\Theta|\hat{W}, D)$.
4. If enough cycles have been performed then stop, otherwise return to step 2.

Step 2 and 3 will be performed via a Metropolis-Hastings update. Chapter 3 and 4 presented different proposal densities, which can be used for these updates. The proposals used by Golightly and Wilkinson (2008) were described in section 3.4 and 4.1.

The important difference between the innovation scheme and the algorithm in 3.2 is that before step 3 the Brownian motion path $\hat{W}$ is deterministically obtained and during step 3, when new parameter values are proposed a corresponding sample path $\hat{Y}$ is generated deterministically by combing the parameter proposal with the previously calculated $\hat{W}$. This variable transformation guarantees that the algorithm does not degenerate for high m by decorrelating the update of the parameters from the latent state update.

4.3.2. Variable Transformation

Golightly and Wilkinson (2008) used equation (3.1) as transformation following the consideration that the Brownian motion can be updated independently of the parameters. It was furthermore noted in Golightly and Wilkinson (2009) that any other deterministic transformation $W = f(Y, \Theta)$, which corresponds to a locally equivalent diffusion will have the same effect. This for example applies to the modified diffusion bridge described in section 3.4. When dealing with measurement error of unknown variance however, the first part proposal used in Golightly and Wilkinson (2009) is better suited, because it includes the current value of the variance and thus guarantees that the current path $\hat{Y}$ fits the current parameters Θ. This transformation will be used in this thesis to overcome the convergence problems of simple Gibbs sampling approaches.

For this purpose we describe the system as a discrete SDE like (3.1) but with the drift and diffusion given by the proposal density (3.15). This leads to

$$Y^{i+1} = Y^i + \widetilde{\mu}(Y^i)\Delta t + \sqrt{\widetilde{\beta}(Y^i, \Theta)}\Delta W^i \tag{4.4}$$

with $\widetilde{\mu}$ and $\widetilde{\beta}$ according to (3.16) and (3.17) respectively. Also we defined $\Delta W^i = W^{i+1} - W^i \sim \mathcal{N}(0, I\Delta t)$. If we solve this equation for ΔW^i, we get

$$\Delta W^i = \left(\widetilde{\beta}(Y^i, \Theta)\right)^{-\frac{1}{2}} \left(Y^{i+1} - Y^i - \widetilde{\mu}(Y^i)\Delta t\right). \tag{4.5}$$

Since the first part proposal is conditioned on an observation we set $i = j, j+1, \ldots, j+m-1$ for $j = 0, m, \ldots, n-m$. This means that the transformation is always influenced by the earliest observation D^M, for which $M > i$.

Because we're transforming a probability variable, we need to compute the Jacobian associated with the transformation. From Golightly and Wilkinson (2009) we know it to be

$$J(\hat{Y}, \Theta) \propto \left(\prod_{i=0}^{n-1} \widetilde{p}\left(Y^{i+1} | Y^i, D^{(\lfloor i/m+1 \rfloor)m}, \Theta\right)\right)^{-1}. \tag{4.6}$$

This means after a new parameter vector Θ_* is drawn from a suitable proposal density a new path $\hat{Y}_*$ is computed deterministically from the Brownian motion $\hat{W}$ via (4.4). The new parameters and the new path are accepted with probability

$$\min\left(1, \frac{\pi(\Theta^*)\pi(\hat{Y}^*|\Theta^*, D)\pi(D|\hat{Y}^*, \Theta^*)J(\hat{Y}^*, \Theta^*)Q(\Theta|\Theta^*)}{\pi(\Theta)\pi(\hat{Y}|\Theta, D)\pi(D|\hat{Y}, \Theta)J(\hat{Y}, \Theta)Q(\Theta^*|\Theta)}\right), \tag{4.7}$$

where $Q(x|x')$ is the proposal density.
In summary the sampling algorithm works in the following way:

1. Set starting values for the parameters Θ and the augmented data $\hat{Y}$
2. Update the augmented data
   a) Draw a new sample path $\hat{Y}^*$ (in whole or parts, see section 3.3)
   b) Accept the new path with the probability associated with the chosen method, otherwise keep the old path.
   c) Compute the Brownian increments ΔW_i from (4.5)
3. Update the parameters
   a) Draw a new parameter vector Θ^* from a proposal density $Q(x|x')$
   b) Compute the corresponding augmented Data $\hat{Y}^*$ by using (4.4)
   c) Accept the new parameters and associated path with probability (4.7), otherwise keep the old values
4. If enough cycles have been performed then stop, otherwise return to step 2.

5. Simulation Results

To compare the proposal densities, which were presented in the last two chapters, several simulations were run and their results will be presented in this chapter[1]. The simulations cover the following path proposals

- Modified diffusion bridge (MDB, Section 3.4)
- Variational block update (VBU, Section 3.5.2)
- Variational complete update (VCU, Section 3.5.3)
- Discrete variational complete Update (DVCU, Section 3.5.4)[2]

and two different proposal densities for the parameter vector θ

- Gaussian random walk on the Logarithm (GRW, Section 4.1)
- Minimized free energy (MFE, Section 4.2).

As a measure of quality, we use the acceptance rate (i.e. the rate at which the proposals are accepted by the MCMC algorithm) and the inefficiency factor, since high acceptance rates can also be a sign of highly correlated samples. Chib et al. (2004) define the inefficiency factor as

$$INF(L) = 1 + 2\sum_{i=1}^{L} xcorr(i), \tag{5.1}$$

where $xcorr(i)$ is the autocorrelation at lag i and L is set to a sufficiently high number. The inverse of this factor was first used by Geweke (1989), but for our purpose, this formulation has the convenience that $INF(L)$ can be interpreted as the necessary thinning factor to get i.i.d. samples from the algorithm's output. For the block updates, the path was only saved as a new sample after one complete cycle. Because of the overlapping blocks this means that every path variable not being at the time of an observations can be updated two times in one run.

[1] For more details on the design and parameters of the simulation see the appendix A.

[2] The Discrete Variational Complete Update is only directly compared to the Variational Complete Update in section 5.1.5.

5.1. Sampling only the Path

In this section only the path was sampled, while the parameters stayed the same during the whole simulation. The true parameter values were know by the algorithm to ensure the best results. Omitting the parameter sampling should make it easier to compare the different methods of proposing the path presented in section 3.3. Additionally, this approach results in significantly better computation time per iteration and shortens the burn-in period, which is normally much longer for the parameters than for the path.

5.1.1. Discretization

One advantage of the decorrelation technique of Golightly and Wilkinson (2008) described in section 4.3 is that the algorithm improves with higher discretization. Figure 5.1 shows how this effects the acceptance rate and the inefficiency factor of the different algorithms for the Lotka-Volterra model. It is clear to see that both variational approaches need higher values of m to work well, while the modified diffusion bridge does not improve beyond the acceptance rate for $m = 5$. This conforms to the statement in Golightly and Wilkinson (2009) that the algorithm's performance does not enhance significantly for $m > 10$ when used on the prokaryotic auto-regulatory gene network described in section 2.3.2. The variational complete update is the only one of the three proposal distributions, which improves considerably for values of $m > 20$ and even surpasses the variational block update for $m > 30$. The inefficiency factors show the same behaviour as the acceptance rates confirming their validity for comparisons. Even though the variational block and complete update work well for the Lotka-Volterra model, their acceptance rates (VBU: $\approx 33\%$ at $m = 50$, VCU: $\approx 37\%$) are still clearly inferior to the modified diffusion bridge's ($\approx 56\%$). From the inefficiency factors we can assume that the variational algorithms need approximately 2.5 times more iterations than the modified diffusion bridge to get the same amount of independent samples.

5.1.2. Measurement Noise

For realistic applications it is important to know how the different proposal distributions cope with measurement noise. To study this, multiple data sets of a Lotka-Volterra simulation were used for inference. Because only a relatively low number of measurements are taken, the randomness of independently drawn measurement noise could lead to sets of observation not representing the strength of the noise they were corrupted with, accordingly. To circumvent this, the observation noise was drawn from a standard normal distribution once and then multiplied by standard derivation values $\sigma = 0.5, 1, \ldots, 3$. The resulting sets of observations for the prey population are shown in figure 5.2.

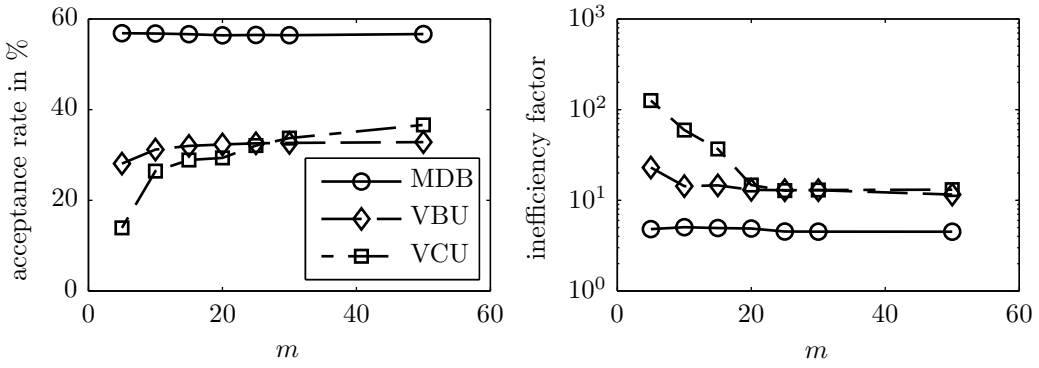

Figure 5.1.: Effect of the discretization factor m on the acceptance rates (left) and inefficiency factor (right) of the proposal distributions. The data was generated from a Lotka-Volterra simulation with parameters $c_1 = c_4 = 0.05$ and $c_2 = c_3 = 0.01$ known by the algorithm. Each simulation lasted 100,000 iterations, from which the first 10,000 were discarded as burn-in. The inefficiency factor was plotted on a semi-logarithmic scale for better clarity. While the modified diffusion bridge does not benefit from discretization beyond $m = 5$, the variational block update reaches its maximum acceptance rate at $m = 15$. The variational complete update seems to improve even beyond $m = 50$ and surpasses the block update at $m = 30$. Both variational methods work well, but the acceptance rate is roughly half as high as for the modified diffusion bridge. The inefficiency factors support these observations. For $m = 50$ they are roughly 2.5 times as high for the variational methods as for the modified diffusion bridge.

The acceptance rates and inefficiency factors for the proposals are drawn as functions of σ in figure 5.3. As expected the modified diffusion bridge acceptance rate drops (and the inefficiency factor grows) when the observations noise rises. Interestingly, for the variational complete update method this effect is reversed and at $\sigma = 3$ it even surpasses the modified diffusion bridge.

The variational block update gains performance until $\sigma = 1$ then the inefficiency factors rise as for the modified diffusion bridge. It is possible that the variational algorithms have problems with low observation noise because they generate samples with a high variation despite the restrictions by the very precise observations. This can be seen in figure 5.4 by comparing a number of proposals with the measurements.

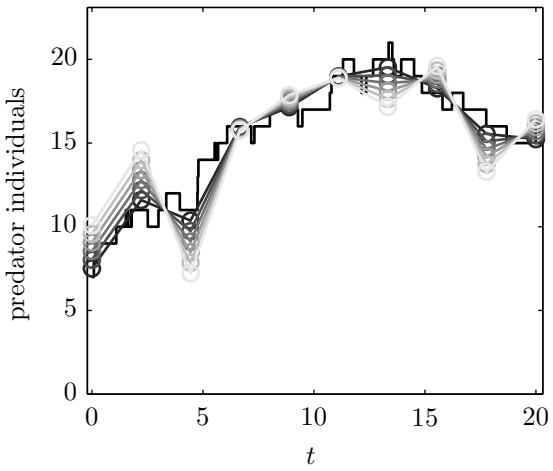

Figure 5.2.: To reduce the effect of randomly chosen measurement noise, for each observation the noise was drawn from a standard normal distribution and later scaled to 6 different values of the noise covariance $\Sigma = diag(\sigma^2)$. The black line shows the real process of the predator population in a Lotka-Volterra model, while the measurements are drawn as circles connected by lines with the intensity corresponding to the standard derivation σ. For the darkest line σ was set to 0.5, then it rises in steps of 0.5 up to 3 represented by the lightest line.

5.1.3. Reactions per Measurement

Simulations have shown that all proposal methods have problems dealing with high observation intervals. But the questions remains how to compare the timescales of different models. For this purpose we use the number of reactions happening in the real process divided by the number of observations. Figure 5.5 shows how the proposal distributions react to different distances between the observations, for both auto-regulatory gene networks. It can be clearly seen, that all three methods break down for high observation intervals. While the modified

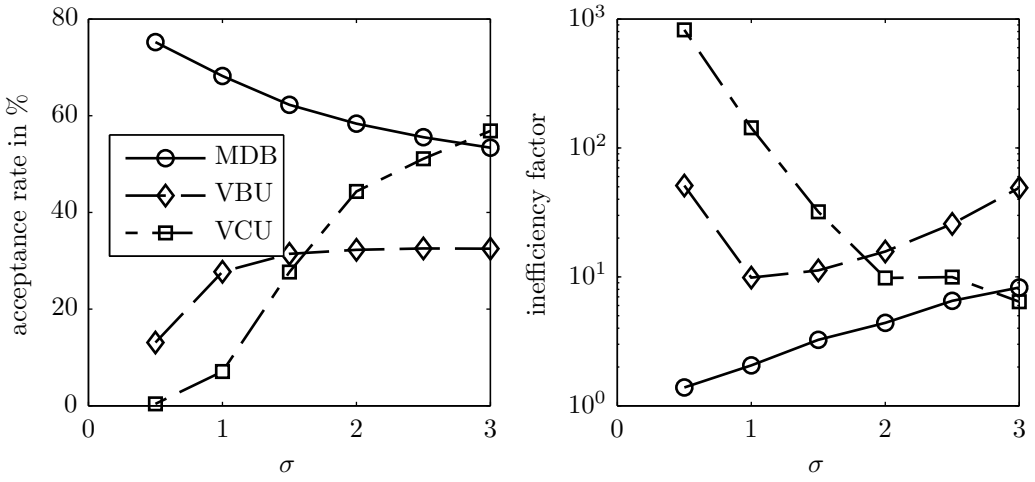

Figure 5.3.: The acceptance rates and inefficiency factors as functions of the observation noise standard derivation σ. The data was obtained from simulations of a Lotka-Volterra model with parameters $c_1 = c_4 = 0.05$ and $c_2 = c_3 = 0.01$ and a discretization of $m = 15$. For better clearness the inefficiency factor was plotted on a semi-logarithmic scale. Every simulation consisted of 100,000 iterations with the first 10,000 dropped as burn-in. Both the declining acceptance rate and the rising inefficiency factor show that the modified diffusion bridge's performance suffers when the observations noise rises. On the other hand, the variational methods first improve until $\sigma = 1$ (block update) and $\sigma = 1.5$ (complete update). For higher noise both methods worsen, indicated by the growing inefficiency factors, even though the acceptance rate of the variational block update stays at approximately 32%.

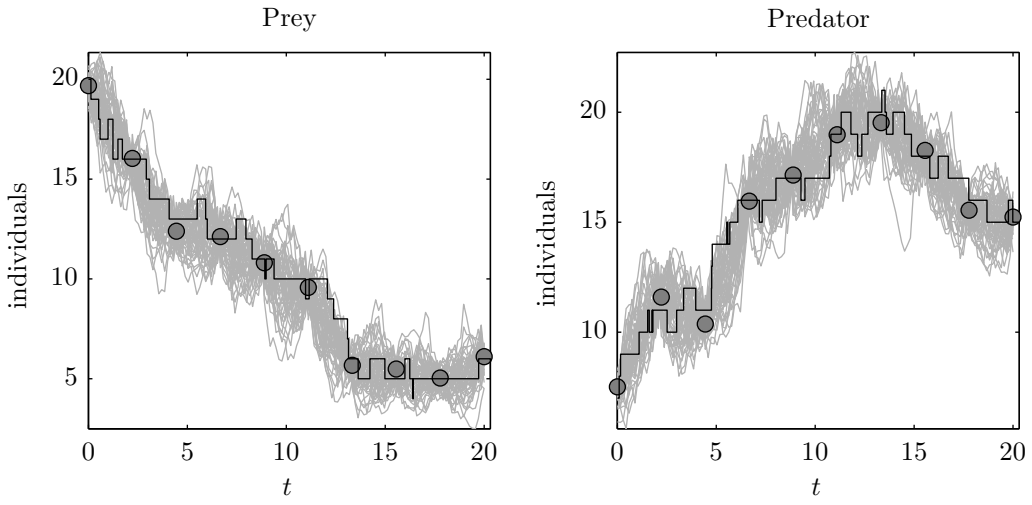

Figure 5.4.: 50 proposals generated by the variational complete update algorithm for a Lotka-Volterra simulation with parameters $c_1 = c_4 = 0.05$ and $c_2 = c_3 = 0.01$. The proposals are drawn as light grey lines. The black line is the real process and the measurements are depicted as grey discs. Since the measurement noise standard derivation is only $\sigma = 0.5$ many proposals differing too much from the measurements are rejected.

diffusion bridge gives good results for both models, the variational complete update works poorly for the prokaryotic gene network model but shows excellent performance for the protein downregulation model. The acceptance rate is above 95% and the inefficiency factor is almost 1. This means the samples are nearly uncorrelated. This might be explained by the high observations noise ($\sigma = 20$) employed on the data as we have seen in section 5.1.2 that the variational complete update even profits from this factor. On the other hand, the variational block update behaves similar to the modified diffusion bridge proposal, but a little bit worse, for the protein downregulation model and gives average results for the prokaryotic gene network model if the number of reactions per measurement is low enough.

It is possible that the proposal distributions' approximation of the real process gets worse when too many reactions happen between two observations, because the marginals become increasingly non-Gaussian. To test this hypothesis the kurtosis is used as a measure of normality as suggested by D'Agostino and Belanger (1990). In detail the measure employed here is defined by

$$NG(\hat{Y}) = \frac{1}{nk} \sum_{i=0}^{n} \sum_{j=1}^{k} |kurt(\hat{Y}_j^i) - 3|, \tag{5.2}$$

with *kurt* being the sample kurtosis and $\hat{Y}$ being a set of samples from Y. It is easy to see that $NG \geq 0$, with equality if all marginals are normal distributed,

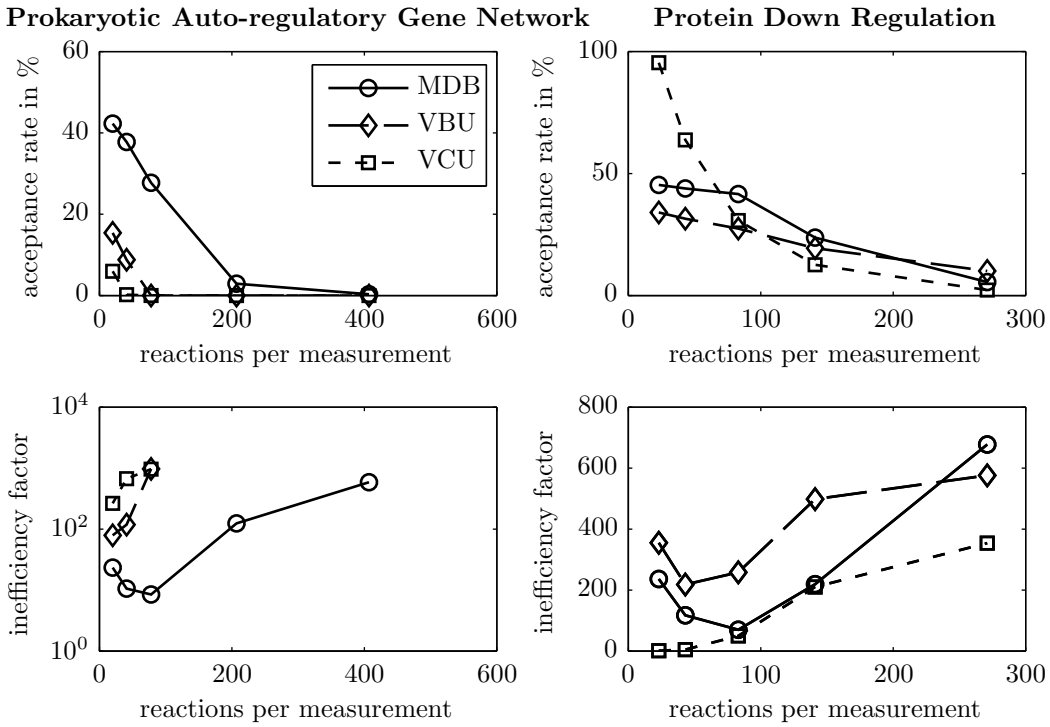

Figure 5.5.: Comparison of the impact of the number of observations per measurement on the acceptance rates. The data on the left was generated from a prokaryotic auto-regulatory gene network model simulation with parameters $c_1 = 0.1, c_20.7, c_3 = 0.35, c_4 = 0.2, c_5 = 0.1, c_6 = 0.9, c_7 = 0.3, c_8 = 0.1$ and $K = 200$. The right graph shows the same data for the protein downregulation network with the parameters $c_1 = 90, c_2 = 1, c_3 = 2.5, c_4 = 0.5$ and $P_c = 200$. The data augmentation was set two $m = 25$ for both models. Each simulation lasted 100,000 iterations, from which the first 10,000 were discarded as burn-in. The algorithms' performance clearly suffers when more reactions happen between the measurements, but the modified diffusion bridge proves to be more robust generally. For the protein downregulation model, however, the variational methods perform well, especially the complete update. The inefficiency factors for the last two data sets of the prokaryotic gene network were not computed for the variational proposal because the acceptance rates were almost 0.

because the sample kurtosis of a normal distribution is 0. The higher NG gets, the more non-Gaussian are the marginals[3]. Figure 5.6 shows a NG plotted over the mean number of reactions per observation, taken from simulations of both gene networks presented in this thesis. As expected for high distances between the observations the marginals become more non-Gaussian. Since the values for less than 100 reactions per measurements vary little it might be necessary to run more simulations for data sets with $200-300$ reactions per measurement in order to confirm the results.

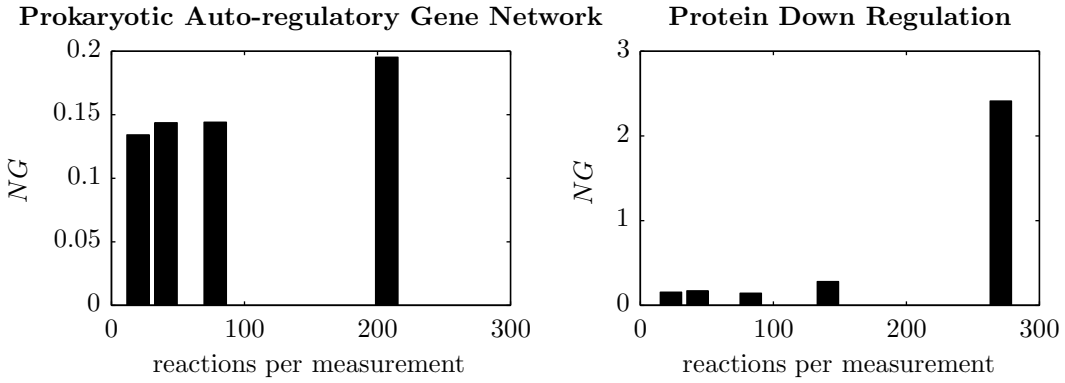

Figure 5.6.: The measure of non-Gaussianity as a function of the number of reactions per observation. The data was taken from the results of the simulations used in figure 5.5 employing the modified diffusion bridge with. It can be seen that the marginals become increasingly non-Gaussian, if more reactions are happening in between the measurements. This could partly explain the dropping acceptance rates of all algorithms for more reactions per measurement. For the prokaryotic gene network the data set with 407 observations per measurement was dropped because, the low acceptance rate of all algorithms did not allow a satisfactory estimation of the posterior distribution.

5.1.4. Bicoid Dynamics & Double Well Model

The Bicoid dynamics model with 8 bins has the most dimensions of all models presented in this thesis. Regarding the good results of the variational methods for the two-dimensional Lotka-Volterra and protein downregulation models, it is possible that the bad performance of the variational proposal distributions for the Bicoid model shown in table 5.1 is a consequence of the high dimensionality. Additionally, both variational algorithms did not give good results for the four-dimensional prokaryotic auto-regulatory gene network. The variational block update performed better than the complete update with the inefficiency factor

[3]There are surely more elaborate techniques for this purpose, especially for multivariate distributions (see e.g. Smith and Jain, 1988), but this simple method proved good enough.

	MDB	VBU	VCU
Acceptance rate in %	27.9	6.9	2.0
Inefficiency factor	15.7	173.7	408.3

Table 5.1.: Acceptance rates and inefficiency factors for Bicoid dynamics model

	MDB	VBU	VCU
Acceptance rate in %	40.8	21.1	1.3
Inefficiency factor	16.8	127.1	600.3

Table 5.2.: Acceptance rates and inefficiency factors for double well model

being roughly 11 (block update) and 26 (complete update) times higher than for the modified diffusion bridge.

Since the double well model has only one dimension we could assume good performance of the variational methods. This is not the case as the results in table 5.2 show. The variational block update has a fairly average acceptance and an inefficiency factor about 7 times as high as the modified diffusion bridge. The complete update on the other hand has an acceptance rate just above 1% and an inefficiency factor of approximately 600. Note that this means 600 iterations will give 1 uncorrelated sample.

For the Bicoid dynamics and prokaryotic gene network model the variational complete update also had bad performance values, but still the Markov chain converged roughly to the desired posterior. In the case of the double-well model this was not the case even after 100,000 iterations. This is most likely due to the bad proposals the variational complete update generates. This can be seen in figure 5.7, which compares ten example proposals of the variational complete update and the modified diffusion bridge.

5.1.5. Discrete Variational Complete Update

As another variational approach the variational discrete complete update was derived in section 3.5.4. To determine if the discrete approach improves the variational method, the continuous and discrete complete proposals are directly compared. Table 5.3 shows the acceptance rates and inefficiency factor of both proposal distributions for all five models. Overall, both algorithms perform extremely similar for all models with the continuous approach performing slightly better, which is not surprising if we compare the values for $b(t)$ and $B(t)$ plotted in figure 5.8.

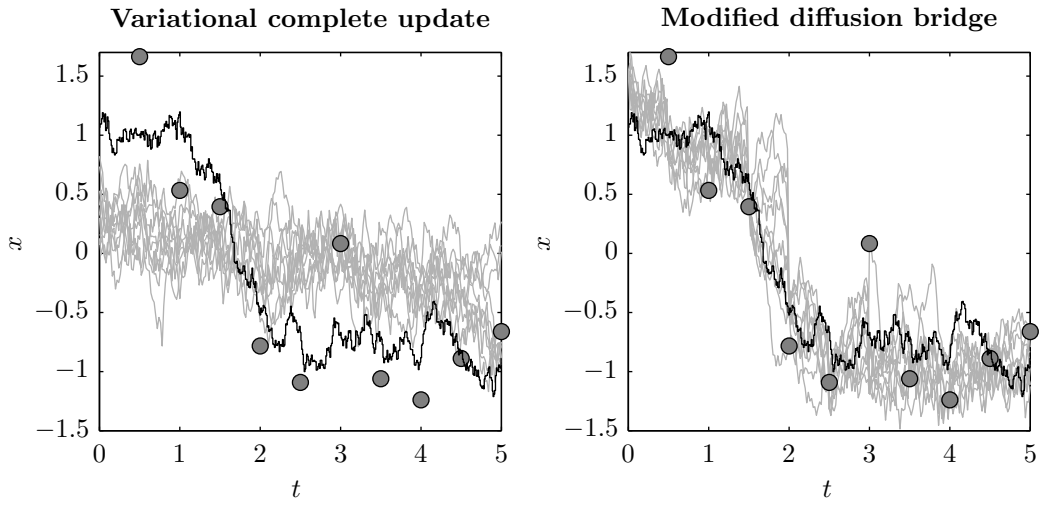

Figure 5.7.: Proposals for a double well simulation. The black line denotes the true process, the grey discs are the noisy measurements. Ten proposals from the variational complete update (left) and the modified diffusion bridge (right) are drawn as grey lines. The proposals of the modified diffusion bridge are composed of all block proposals of one run. Evidently the variational complete proposals do not manage to infer the essence of the true process.

	acceptance rate in %		inefficiency factor	
model	DVCU	VCU	DVCU	VCU
Lotka-Volterra	25.0	32.0	39.5	12.8
Protein downregulation	90.2	95.4	1.2	1.3
Prokaryotic gene network	1.5	6.0	507.2	264.3
Bicoid dynamics	0.1	2.0	613.7	408.0
Double well	0.3	0.4	853.7	809.1

Table 5.3.: Comparison of discrete and continuous variational complete update

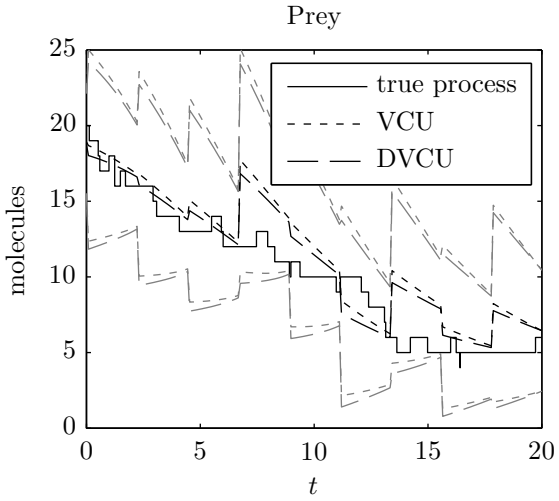

Figure 5.8.: Comparison of the discrete and continuous variational complete update. The true process is the solid black line. The dotted lines represent $b(t)$ (black) with $2\sqrt{B(t)}$ as a confidence interval (grey) for the continuous variational complete update. The discrete variational update is drawn as a dashed line. Again the black line is $b(t)$ and the grey lines around it are a confidence interval of width $2\sqrt{B(t)}$ around $b(t)$. The minimal differences explain why both algorithms behave alike for all models.

5.1.6. Computational Costs

Complicated proposal distributions can perform extremely well in MCMC algorithms if they accurately approximate the real posterior. This often comes at the price of very high computation times, negating the benefits of the high acceptance rates. In figure 5.9 the computational costs of the path proposal distributions presented in this thesis are compared. Because the implementation of the algorithms were not optimized, the results are no clear evidence but they indicate that the variational methods can be calculated with computational costs in the same order of magnitude as for the modified diffusion bridge. Since the parameter were not changed during these simulations, the discrete and continuous complete update only solved the backward equation once. If parameter inference were conducted, their computational costs would probably rise to a level similar to the variational block update's. If satisfactory starting values for the parameters are obtained[4] the complete update's proposal distributions could be left independent of the current parameter values in order to still achieve the current performance.

[4]E.g. from the free energy minimization described in section 4.2.

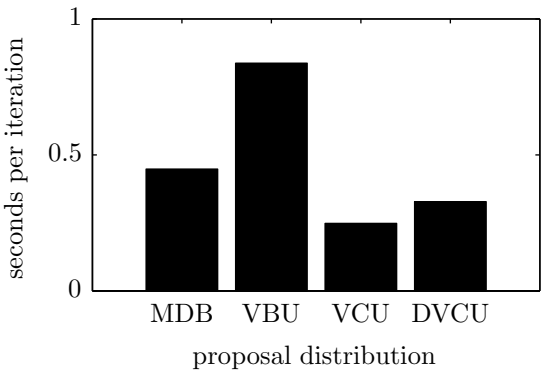

Figure 5.9.: Computational costs of the proposal distributions for a Lotka-Volterra model with 10 measurements and $m = 25$. The values were averaged over 1000 iterations on an Intel® Core™2 Duo CPU T8300 running at 2.40GHz. Because the algorithms were not always programmed for the fastest computation time, this comparison should only be seen as an indication of the scale of the computational costs. For the discrete and continuous variational complete update, the backward equation was only solved one time.

5.2. Sampling the Path and the Parameters

In order to compare the two parameter proposal distributions presented in chapter 4, the following simulations all employ the modified diffusion bridge, since it performed sufficient for all models. Because of the problems of the variational methods approximating the double well model's path, no parameter inference was done on this model.

The acceptance rates of the algorithm are presented in table 5.4. As the variational approach for the path update, the minimized free energy distribution showed bad results for the higher-dimensional models. The acceptance rates for the Lotka-Volterra and protein downregulation model were also low, but suffice for a more detailed analysis.

The Gaussian random walk on the logarithm has good acceptance rates for the Lotka-Volterra and Bicoid dynamics model. It performs as the free energy minimization proposal distribution for the protein downregulation and a little bit better for the prokaryotic gene network. The bad performance for the latter model might be explained by the algorithm being adjusted on smaller parameter values because Golightly and Wilkinson (2009) showed that this method can give good results for this model.

Generally the parameter samples were highly correlated for all methods, probably because the covariance of the Gaussian random walk was chosen too small ($C_p = diag(0.001^2)$). To reduce the computational costs, all simulations in this section were run with m set to 15.

	acceptance rate in %	
model	MFE	GRW
Lotka-Volterra	4.03	49.78
Protein downregulation	4.76	4.47
Prokaryotic gene network	0.24	0.54
Bicoid dynamics	0.01	49.64

Table 5.4.: Acceptance rates for the parameter update. The values were obtained by simulations with 500,000 iterations, from which the first 10,000 were discarded as burn-in and the rest was thinned by a factor of 1,000. The rate constants of the simulations can be taken from tables 5.5, 5.6, 5.7 and 5.8.

5.2.1. Lotka-Volterra model & Protein Downregulation Model

The resulting posteriors for the Lotka-Volterra model are presented in table 5.5 and visualized in figure 5.10. It can be seen that the minimized free energy proposal distribution is an good estimate of the true parameter values, but it does not necessarily overlap with the posterior of the data[5]. Additionally the variance is so small that seldom highly probable parameter values are drawn. This explains the low acceptance rates. It can also be seen that the posterior mean for the free energy minimization lies between the proposal distribution's mean and the mean of the Gaussian random walk on the logarithm's posterior.

The results of both parameter update algorithms for the protein downregulation model are summarized in table 5.6 and illustrated in figure 5.11. The minimized free energy proposals are good estimates of the real parameter values but again don't overlap with the regions of high probability of the Gaussian random walk on the logarithm's posterior. Why the Gaussian random walk reaches the high values for c_1 is unclear and might be a problem of the dataset or the algorithms implementation.

5.2.2. Prokaryotic Gene Network & Bicoid Dynamics Model

The last two models used for parameter inference are the prokaryotic autoregulatory gene network and the Bicoid dynamics model. The performance of both methods for these models was rather bad and because of this a visualization of the results does not give satisfactory results they are, however, presented in table 5.7 and 5.8. It is still interesting to see that both algorithms give good estimates of the true parameters but seemingly do not converge to the real posterior. The bad performance of the minimized free energy proposal is perhaps a consequence of the low variation of the proposal distribution, generating only samples very near to the initial estimate.

[5] We assume that the Gaussian random walk on the logarithm converged to the posterior.

	c_1	c_2	c_3	c_4
		True values		
	5.00×10^{-2}	1.00×10^{-2}	1.00×10^{-2}	5.00×10^{-2}
		Starting Values		
	2.00×10^{-2}	2.00×10^{-2}	2.00×10^{-2}	2.00×10^{-2}
		MFE proposal		
Mean	2.51×10^{-2}	6.59×10^{-3}	5.34×10^{-3}	5.40×10^{-2}
SD	4.89×10^{-2}	4.23×10^{-3}	4.36×10^{-3}	6.19×10^{-2}
		MFE posterior		
Mean	2.49×10^{-2}	6.58×10^{-3}	5.33×10^{-3}	5.25×10^{-2}
SD	2.90×10^{-3}	7.00×10^{-5}	5.00×10^{-5}	7.40×10^{-3}
		GRW posterior		
Mean	1.55×10^{-2}	9.58×10^{-3}	2.34×10^{-3}	5.10×10^{-2}
SD	2.30×10^{-3}	1.60×10^{-3}	3.64×10^{-4}	1.78×10^{-2}

Table 5.5.: Inference results for the parameter posterior of a Lotka-Volterra simulation. These results are generated from 500,000 iterations of the MCMC algorithm after dropping 20% as burn-in and thinning the rest to give 400 samples. The MFE proposal shows good approximation of the true parameter values but does not give highly proposal samples from the posterior.

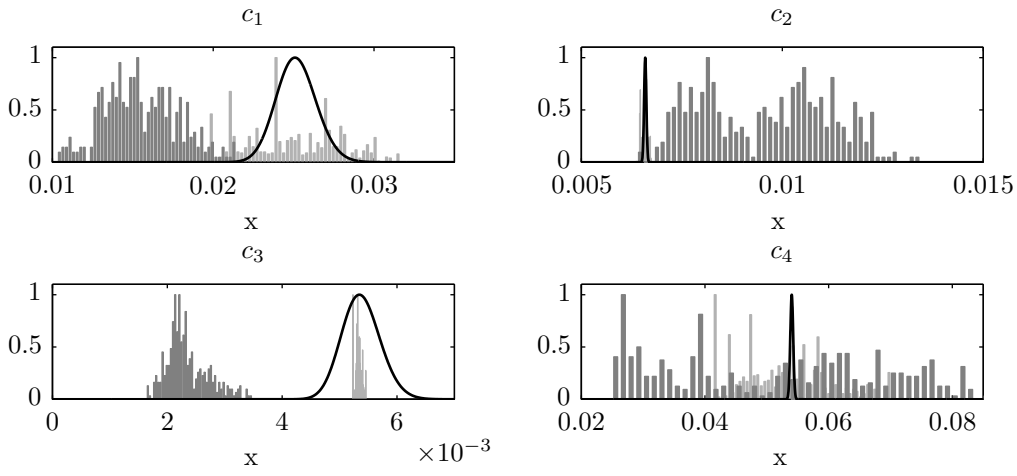

Figure 5.10.: Posterior of the minimized free energy method (light grey blocks) and the Gaussian random walk on the logarithm (dark grey blocks). The proposal distribution of the minimized free energy approach is drawn as a black line. For a better overview the histograms and proposal density have been normalized with respect to their maximum. The true parameter values are found in table 5.5. It can be seen, that most accepted proposals are not from the regions with high probability of the proposal distribution, resulting in a low acceptance rate for the minimized free energy.

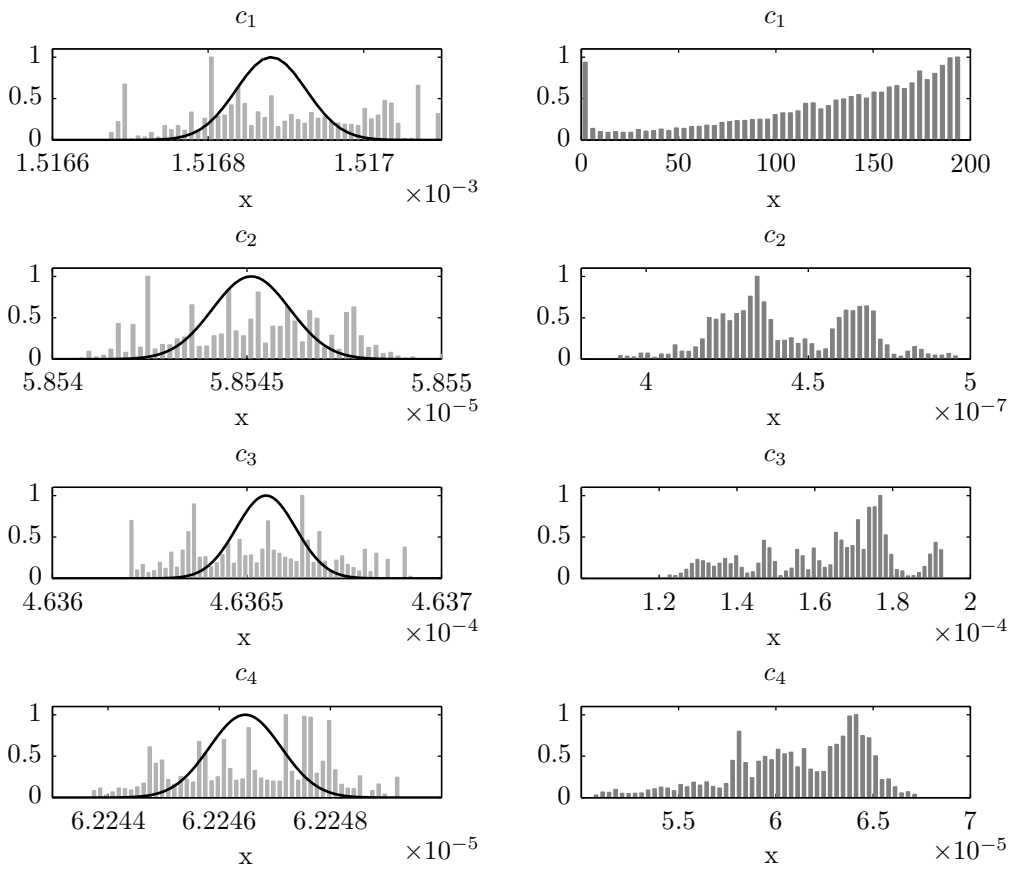

Figure 5.11.: Visualization of the posteriors of the minimized free energy proposal distribution (left) and the Gaussian random walk on the logarithm (right). The proposal distribution for the former method is drawn as a black line. The parameters of the simulations can be found in table 5.6 The histograms and proposal density have been normalized to their maximum to give a better overview. It remains unclear why the parameter c_1 seems to have been drawn in the direction of the highest parameter values allowed by the Gaussian random walk. Again, many accepted proposals of the free energy minimization are from the outer regions of the proposal distribution.

	c_1	c_2	c_3	c_4
		True values		
	2.00×10^{-3}	6.00×10^{-5}	5.00×10^{-4}	7.00×10^{-5}
		Starting values		
	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}
		MFE proposal		
Mean	1.52×10^{-3}	5.85×10^{-5}	4.64×10^{-4}	6.22×10^{-5}
SD	2.96×10^{-5}	1.70×10^{-5}	1.65×10^{-5}	1.04×10^{-5}
		MFE posterior		
Mean	1.52×10^{-3}	5.85×10^{-5}	4.64×10^{-4}	6.22×10^{-5}
SD	1.12×10^{-7}	1.99×10^{-9}	1.85×10^{-8}	1.27×10^{-9}
		GRW posterior		
Mean	1.29×10^{2}	4.44×10^{-7}	1.64×10^{-4}	6.10×10^{-5}
SD	5.48×10^{1}	2.12×10^{-8}	1.77×10^{-5}	3.43×10^{-6}

Table 5.6.: Inference results for the parameter posterior of a protein downregulation simulation with 500,000 iterations from which the first 100,000 were dropped as burn-in and the rest was thinned by a factor of 1000. The minimized free energy proposal's mean is near the true parameters but it is unclear if the Markov chain converged to the true posterior.

	c_1	c_2	c_3	c_4
		True values		
	2.00×1^{-1}	7.00×10^{-1}	3.50×10^{-1}	2.00×10^{-1}
		Starting values		
	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}
		MFE proposal		
Mean	1.64×10^{-1}	1.52×10^{0}	2.53×10^{-1}	2.16×10^{-1}
SD	7.56×10^{-3}	8.79×10^{-3}	6.89×10^{-3}	8.04×10^{-3}
		MFE posterior		
Mean	1.17×10^{-1}	1.51×10^{0}	2.51×10^{-1}	2.12×10^{-1}
SD	2.65×10^{-3}	1.50×10^{-2}	3.08×10^{-3}	2.93×10^{-3}
		GRW posterior		
Mean	9.16×10^{-3}	6.22×10^{-2}	1.39×10^{-1}	1.94×10^{-1}
SD	9.16×10^{-6}	3.05×10^{-4}	5.74×10^{-3}	1.40×10^{-3}
	c_5	c_6	c_7	c_8
		True values		
	1.00×10^{-1}	9.00×10^{-1}	3.00×10^{-1}	1.00×10^{-1}
		Starting values		
	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}
		MFE proposal		
Mean	3.85×10^{-1}	1.49×10^{0}	2.90×10^{-1}	1.38×10^{-1}
SD	7.30×10^{-4}	1.12×10^{-2}	1.01×10^{-2}	7.54×10^{-3}
		MFE posterior		
Mean	3.86×10^{-1}	1.52×10^{0}	2.87×10^{-1}	1.37×10^{-1}
SD	4.00×10^{-4}	2.63×10^{-2}	3.35×10^{-3}	1.77×10^{-3}
		GRW posterior		
Mean	1.38×10^{-1}	1.24×10^{0}	1.02×10^{0}	2.30×10^{-1}
SD	2.02×10^{-5}	3.46×10^{-3}	7.15×10^{-2}	8.21×10^{-4}

Table 5.7.: Inference results for the parameter posterior of a prokaryotic gene network simulation. The results were taken from 500,000 iterations of the MCMC algorithm after the first 100,000 were left out as burn-in and the rest was thinned to give 400 samples. Even though the mean of the free energy minimization proposal distribution is near the true parameters, the standard derivation is so low, that the real parameters are seldom inside a 95% confidence interval.

	k_1	k_2	d
		True values	
	1.00×10^{-3}	4.00×10^{-1}	5.00×10^{-2}
		Starting values	
	5.00×10^{-3}	1.00×10^{-1}	1.00×10^{-2}
		MFE proposal	
Mean	3.30×10^{-5}	3.18×10^{-1}	2.63×10^{-2}
SD	8.29×10^{-3}	8.30×10^{-1}	4.80×10^{-3}
		MFE posterior	
Mean	3.22×10^{-5}	1.61×10^{-2}	2.64×10^{-2}
SD	2.64×10^{-19}	2.64×10^{-16}	1.67×10^{-16}
		GRW posterior	
Mean	3.45×10^{-3}	2.08×10^{-1}	3.64×10^{-3}
SD	6.62×10^{-4}	7.10×10^{-2}	1.50×10^{-3}

Table 5.8.: Results of parameter inference for a Bicoid dynamics simulation with 500,000 iterations. The first 100,000 samples were dropped as burn-in the rest was thinned by a factor of 1000. Interestingly, the Gaussian random walk estimates k_1 and k_2 well but not d. On the other hand the minimized free energy proposal approximates k_2 and d well but not k_1.

6. Summary & Discussion

6.1. Summary

In this thesis the effect of different proposal distributions for MCMC inference of diffusion processes has been studied. The variational approaches based on the work of Ruttor et al. (2009) have proven that they can be an alternative to the modified diffusion bridge of Golightly and Wilkinson (2009). While the latter method has shown to give generally good results for all models in this thesis, the variational method can compete and even surpass it for certain models.

It was shown that for the two-dimensional Lotka-Volterra and protein downregulation models the variational approaches performed well. Especially the variational complete update showed excellent acceptance rates for the protein downregulation model even for high distances between the observations. For the higher-dimensional prokaryotic gene network and Bicoid dynamics models only very few variational proposals were accepted. The displayed acceptance rates were so low that efficient posterior estimation was not possible. Interestingly, the one-dimensional double well model proved particularly difficult for the variational proposals being unable to predict the well change.

By applying the innovation scheme of Chib et al. (2004) the algorithm no longer breaks down when the discretization is increased but rather improves. For the Lotka-Volterra model the modified diffusion bridge proposal reaches its maximum performance if 10 data points are inserted between every two observations. Further increasing the discretization factor m has no significant effect. In contrast to this, the variational updates gain performance even for values of $m = 50$.

One interesting property of the variational complete update seems to be that their acceptance rates improve when the observation noise strengthens contrariwise to the behaviour of the modified diffusion bridge. The variational block update showed both properties, gaining performance until $\sigma = 1$, after which the correlation of the samples increased.

An informal comparison of the computational costs of the four path proposal algorithms showed best results for the variational complete update and highest computational costs for the block update. This emphasizes the practicability of the variational complete update for certain cases even though further investigation is needed.

Parameter inference proved to be a complicated task. As the variational methods for the path update, the free energy minimization showed bad results for the

Bicoid dynamics and the prokaryotic gene network models. Their performance for the two-dimensional models was not good but proved that the method gives good estimations of the true parameters but often does not overlap the posterior distribution resulting in high rejection rates. The Gaussian random walk on the logarithm had good acceptance rates for the low-dimensional models but because of a bad adjustment of the algorithm the samples were very highly correlated. Additionally the performance for the higher-dimensional models was not good either.

6.2. Discussion & Outlook

6.2.1. Advantages and Problems of the Variational Proposals

In this thesis a proposal distribution derived from the variational approach of Ruttor et al. (2009) was tested on different models and compared with the algorithm of Golightly and Wilkinson (2009). Although Elerain et al. (2001) states that updating the complete path in one step is not practical, the variational complete update performed very well for the two-dimensional models presented in this thesis. The results for the Bicoid dynamics and prokaryotic gene network models suggested that the variational complete update suffers when the number of species is large. The Bicoid dynamics model itself is an ideal candidate to further test this assumption because the number of species can be chosen arbitrary.

Furthermore it might be of interest to consider employing the mean field approach presented by Opper and Sanguinetti (2008) in a proposal distribution. Opper and Sanguinetti (2008) state that by factorizing across the species, their algorithm scales well for systems with a large number of species. It might be of interest if this could be used to create path proposals for high-dimensional models, like the Bicoid dynamics model. Opper and Sanguinetti (2008) work with discrete states, which means the method must be adjusted to work with the diffusion approximation of Golightly and Wilkinson (2005) used in this thesis. Another option would be to use MCMC methods conserving the discreteness of the path, e.g. the recently proposed method of Henderson et al. (2010).

The double well model proved to be a hard problem for the variational complete update. Nevertheless Archambeau et al. (2008) showed that variational inference for this model can be performed. It was furthermore noted that the variational approach underestimates the variation in certain regions of the path. Because the minimized free energy parameter proposal depends on the variational approach, this might be a reason for its low variation.

Both the minimized free energy parameter proposal and the variational complete update proposal were not conditioned on the current value of the parameter and path respectively[1]. A general problem of this approach was encountered

[1]This is sometimes called *independence sampling.*

during the simulations, when values, which are rudimentary plausible from the observation data but very unlikely to be drawn from the proposal distribution, are nevertheless drawn. They will almost surely be accepted and can stay for thousands of iterations because the Metropolis-Hastings acceptance probability (2.2) will be very low for the more obvious proposals.

The performance of the variational block update was constantly worse then the modified diffusion bridge's. One possible reason for this is that the border values of the blocks are not necessarily members of a mutual sample path. It might be considered updating the blocks from back to front since the variational proposals are derived from the backward equation of the diffusion process.

While this thesis only compared the block updating scheme, which was described in Golightly and Wilkinson (2008) with the complete update of the path, other block sizes should be studied. Shephard and Pitt (1997) e.g. choose the size of the blocks randomly during each iteration, guaranteeing that the blocks overlap at different points. Another compromise between the two extremes would be to make the blocks span a constant number of observations, larger than in the approach of Golightly and Wilkinson (2008). It would be interesting to study if the variational block update would benefit from a larger block size, while not having the disadvantages of the complete update.

6.2.2. Prior Distributions

The prior distributions employed in this thesis are pretty simple, only restricting the parameters to a wide range of values. It was pointed out by Shen et al. (2009) that priors have a strong influence on the inference results of variational methods.Golightly and Wilkinson (2006) suggested using the estimated posterior of a dataset as a prior for new datasets from the same source. This would be especially interesting for actual measurements, e.g. for the prokaryotic auto-regulatory network or the Bicoid dynamics model, because it would enable the algorithm to progressively learn the characteristics of a real model.

6.2.3. Enhacing the Parameter Proposals

The minimized free energy parameter proposal proved that it cannot be used directly as a proposal for most cases, but the good estimates of the true parameters could be used in more elaborate ways. One possible way would be to use it as an initialization. Especially for models with many parameters the burn-in period can be very long if a simple Gaussian random walk is applied. By initializing the parameters with the maximum likelihood estimate, they would start in regions of high probability reducing the convergence times significantly. Additionally the approximation could be combined with other methods, e.g. by using a mixture of the fixed distribution and a random walk update. Furthermore hybrid Monte Carlo could be used for parameter estimation, avoiding random walk behaviour.

6.2.4. Alternatives to MCMC

In contrast to this thesis Fearnhead (2008) considers alternative approaches to MCMC methods. In a formal way, optimal proposal distributions for importance sampling algorithms are derived and evaluated. In addition, sequential Monte Carlo methods and the Forward-Backward algorithm are discussed. The latter was used inside an MCMC algorithm by Scott (2002) and could be employed for our purpose if we formulate the discretely observed diffusion processes as a hidden Markov model. Variational approximations without MCMC have the disadvantage that they do not give the real posterior distribution. Nonetheless their results are good enough for many cases as Shen et al. (2009) noted and they are many times more computational efficient then MCMC sampling strategies.

A. Simulation Design

A.1. Parameters

A.1.1. Model Parameters

When not noted otherwise the parameters used for simulations were:

- **Lotka-Volterra model:** $c_1 = 0.05, c_2 = 0.01, c_3 = 0.01, c_4 = 0.05$
- **Protein downregulation model:** $c_1 = 90, c_2 = 1, c_3 = 2.5, c_4 = 0.5, P_c = 200$
- **Prokaryotic auto-regulatory gene network model:** $c_1 = 0.1, c_2 = 0.7, c_3 = 0.35, c_4 = 0.2, c_5 = 0.1, c_6 = 0.9, c_7 = 0.3, c_8 = 0.1, K = 200$
- **Bicoid dynamics model:** $k_1 = 0.001, k_2 = 0.4, d = 0.05$
- **Double well model:** $\theta = 1, \beta = 0.5$

A.1.2. Observations

From the real process data observations were taken at equidistant time points. Unless stated otherwise 11 measurements were taken and corrupted by Gaussian noise with the covariance matrix $\Sigma = diag(\sigma^2)$, where

- $\sigma = 2$ for the Lotka-Volterra model
- $\sigma = 20$ for the protein downregulation model
- $\sigma = 2$ for the Prokaryotic auto-regulatory gene network model
- $\sigma = 2$ for Bicoid dynamics model
- $\sigma = 0.5$ for the double well model.

A.1.3. Prior distributions

An uniform distribution for all positive numbers was used as prior over the start values as of the path:

$$\pi(Y^0) = \begin{cases} 1 & \text{if } Y^0 > 0 \\ 0 & \text{otherwise} \end{cases} \tag{A.1}$$

For the parameters the prior was defined on the logarithm

$$\pi(\Theta) = \begin{cases} 1 & \text{if } -10 \leq \log(\Theta) \leq 5 \\ 0 & \text{otherwise} \end{cases}, \tag{A.2}$$

which effectively just prevents the parameters to get too small or large. For the sake of simplicity both prior distributions defined above are written in a not normalized form. In the algorithm only ratios of the prior values are used therefore normalization is unnecessary for our cause.

A.1.4. Other Parameters

For the Gaussian random walk on the logarithm (section 4.1) a covariance of $C_p = diag(0.001^2)$ was used, while for the modified diffusion bridge in the first block the start value was chosen from a Gaussian distribution centred on the current value, with covariance matrix $diag(1)$.

The continuous variational approaches were calculated with the integration step-size set to Δt meaning the time distance between two adjacent points of the augmented data.

The inefficiency factor of equation (5.1) was computed with $L = 500$.

A.2. Implementation Details

All programs used for simulations in this thesis were implemented in MATLAB ® version R2008b and R2009b. For simulations employing the variational methods presented in section 3.5.2 and 3.5.3 a software framework programmed by Dr. Andreas Ruttor was used to draw proposals. For the free energy minimization parameter proposal (section 4.2) and the discrete variational update (section 3.5.4) the Nelder-Mead simplex algorithm[1] was used for numerical minimization.

[1]The algorithm was first introduced in Nelder and Mead (1965). Because the notation in this publication is considered ambiguous, several variations have been implemented. For this thesis the *fminsearch* function of MATLAB® was employed, which follows the interpretation of Lagarias et al. (1998).

A.3. List of Acronyms

DVCU	Discrete variational complete update
GRW	Gaussian random walk
INF	Inefficiency factor
KL	Kullback-Leibler (divergence)
MCMC	Markov chain Monte Carlo
MDB	Modified diffusion bridge
MFE	Minimized free energy
MJP	Markov jump process
NG	(Measure of) non-Gaussianity
SD	Standard deviation
SDE	Stochastic differential equation
VBU	Variational block update
VCU	Variational complete update

A.4. Path Posteriors

Because for every proper proposal distribution the Markov chain should converge to the real posterior, they are not suitable for comparison of different proposal methods in most cases. For the sake of completeness and an illustration of the quality of the predictions figure A.1, A.2, A.3, A.4 and A.5 show the path posterior distributions for the various models of this thesis.

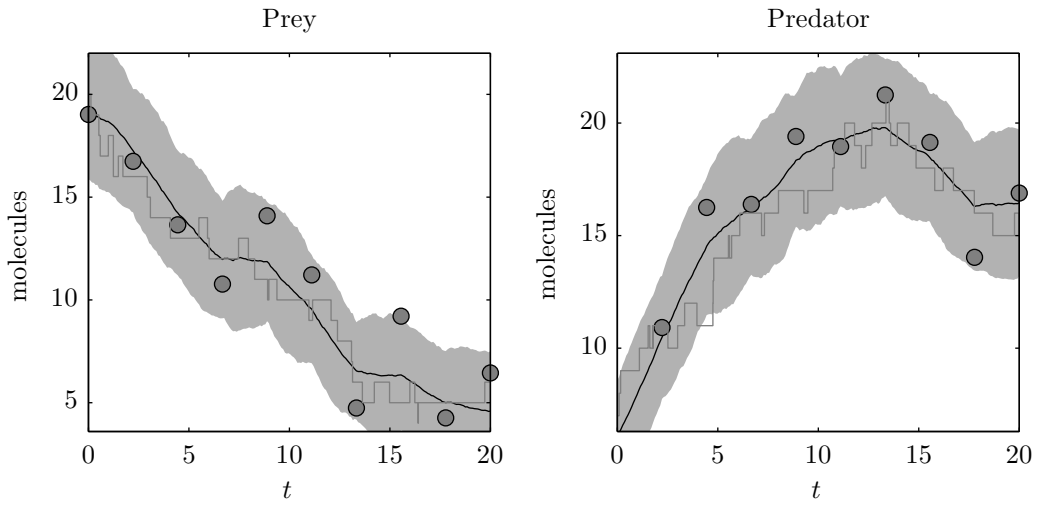

Figure A.1.: Posterior of the path for a Lotka-Volterra simulation with parameters $c_1 = c_4 = 0.05, c_2 = c_3 = 0.01$. The true process is the grey line, from which ten measurements were taken corrupted with Gaussian noise with standard deviation $\sigma = 2$. The black line denotes the posterior mean, the grey area around it is a confidence interval of two times the standard deviation. The posterior was taken from 100,000 iterations of the algorithm with the modified diffusion bridge as path proposal. The first 10,000 iterations were omitted as burn-in and the rest was thinned by a factor of 100.

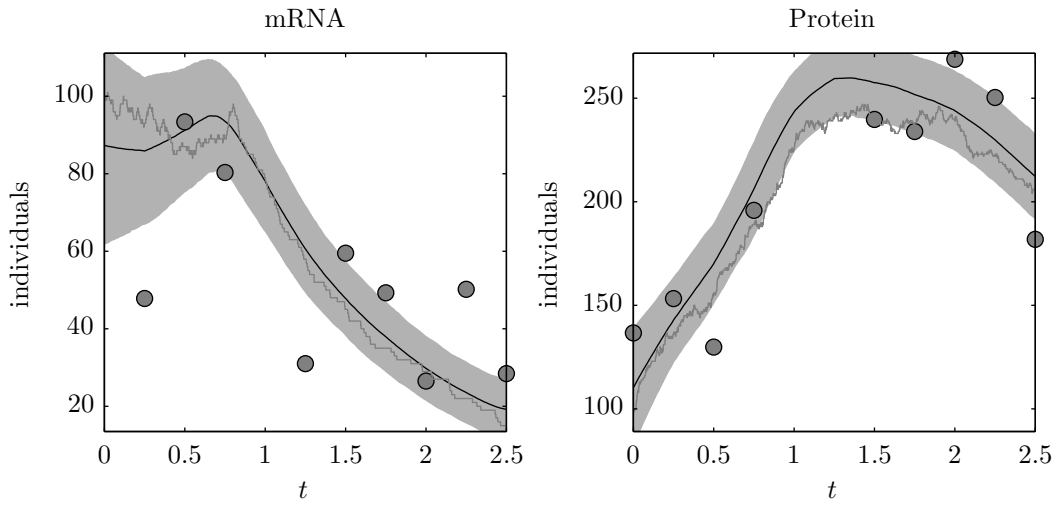

Figure A.2.: Posterior of the path for a protein downregulation simulation with parameters $c_1 = 90, c_2 = 1, c_3 = 2.5, c_4 = 0.5$ and $P_c = 200$. The grey line is the true process, from which measurements were taken corrupted by Gaussian observation noise with standard deviation $\sigma = 20$. The mean of the posterior is drawn as a black line surrounded by a double standard deviation confidence interval. 100,000 iterations were used to get the posterior, from which 10,000 were discarded as burn-in and the rest were uncorrelated through a thinning factor of 100. The path proposal used for inference was the variational complete update.

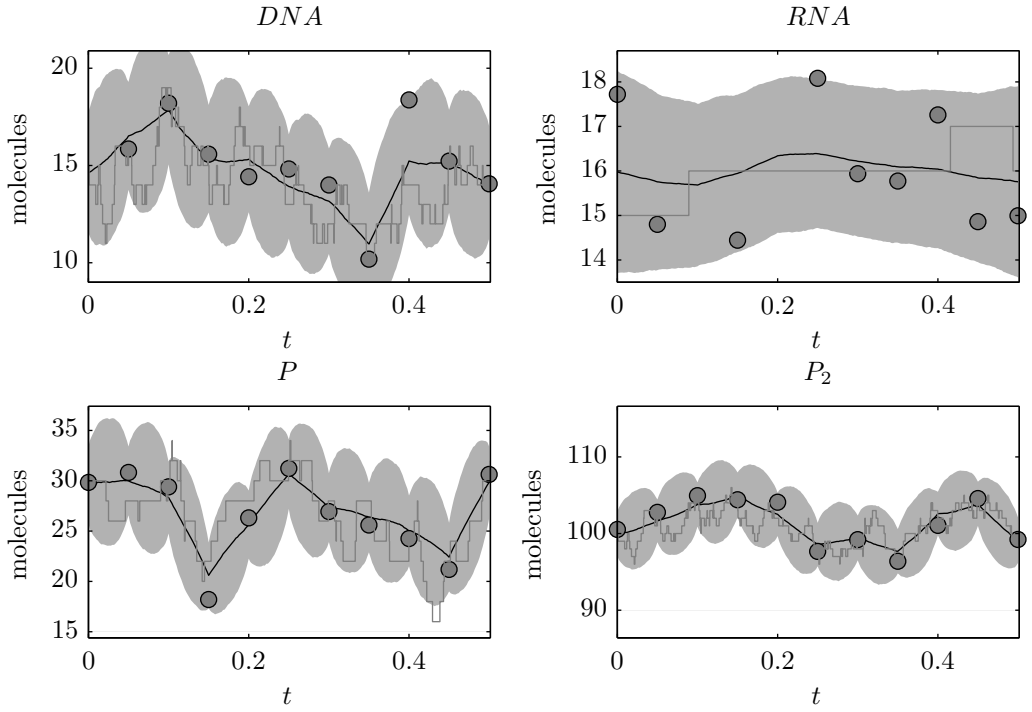

Figure A.3.: Posterior path of a prokaryotic gene network simulation with rate constants $c_1 = 0.1, c_2 0.7, c_3 = 0.35, c_4 = 0.2, c_5 = 0.1, c_6 = 0.9, c_7 = 0.3, c_8 = 0.1$ and $K = 200$. The true process is drawn as a grey line, from which measurements were taken. The observation noise was Gaussian with standard deviation $\sigma = 2$. The black line represents the posterior mean with a double standard deviation confidence interval around it, depicted as a grey area. The posterior is the results of 100,000 iterations of the MCMC algorithm with the modified diffusion bridge. The first 10,000 were left out as a burn-in period and the rest was thinned to give 900 samples.

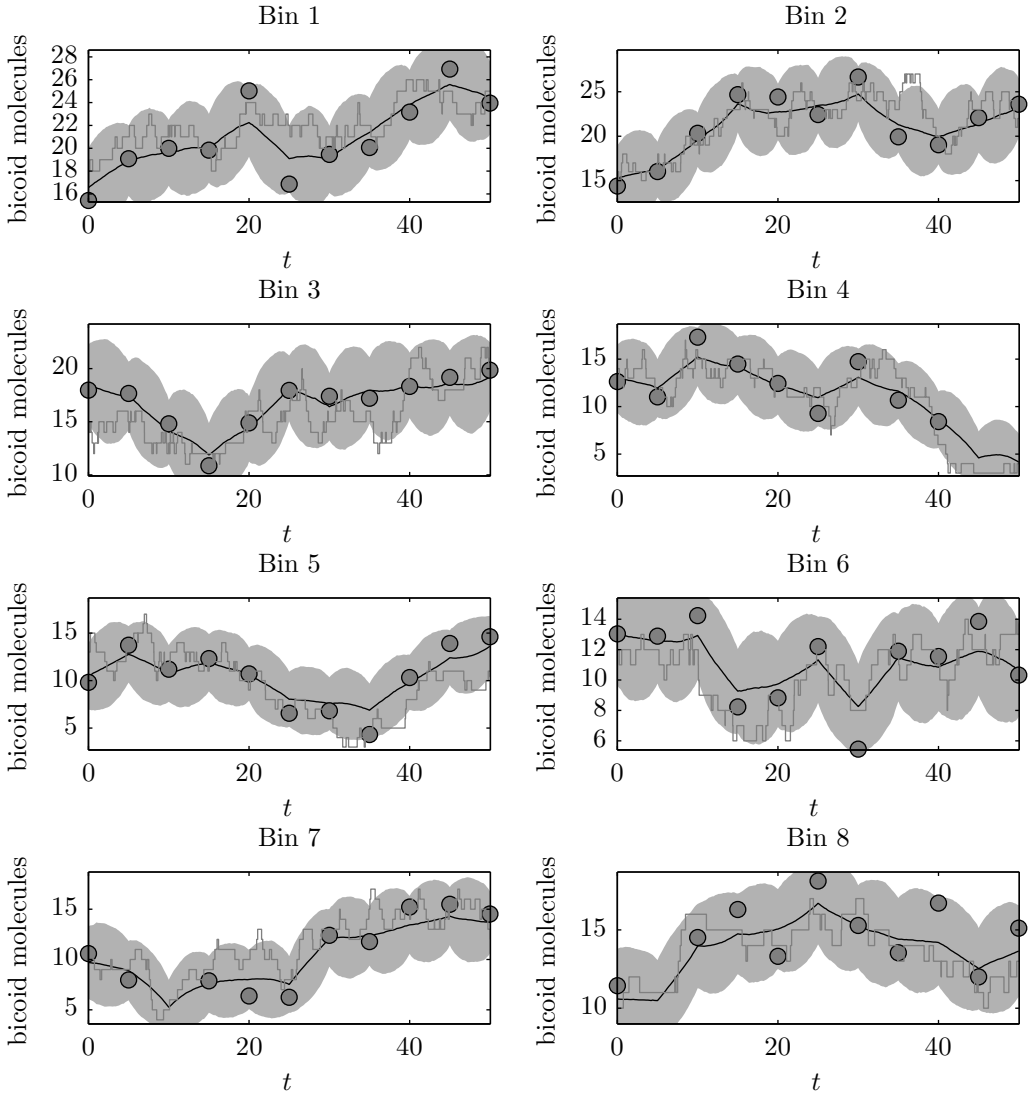

Figure A.4.: Posterior for the spatial distribution of Bicoid molecules. The simulation was run with the following parameters: $k_1 = 0.001, k_2 = 0.4$ and $d = 0.05$. The grey line represents the true process, from which noisy measurements were taken corrupted by Gaussian noise with standard deviation $\sigma = 2$. The posterior mean is drawn as a black line surrounded by a grey area denoting a confidence interval of two times the standard deviation. The posterior resulted from 100,000 iterations. The first 10,000 were dropped as burn-in and the rest was thinned by a factor of 100. The modified diffusion bridge was used as the path proposal of the algorithm.

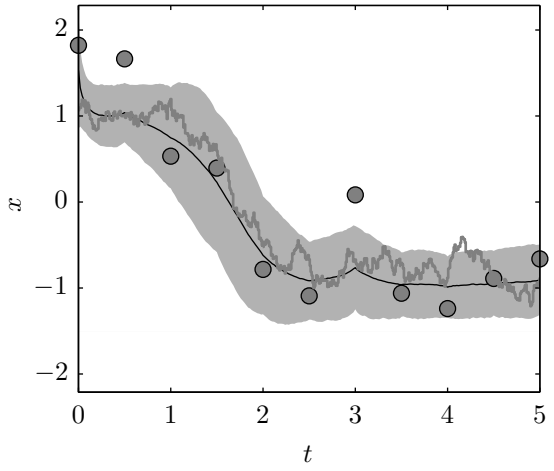

Figure A.5.: Posterior distribution for a simulation of the double well model. The parameters used for simulation are $\theta = 1$ and $\beta = 0.5$. Measurements (grey discs) were taken from the true process (grey line) after adding Gaussian noise with standard deviation $\sigma = 0.5$. The posterior is drawn as a black line, denoting the mean, and a grey area representing a double standard deviation confidence interval. The MCMC algorithm ran 100,000 iterations. From this 10,000 were omitted as burn-in and the rest was thinned to give 900 samples of the posterior. As path proposal distribution the modified diffusion bridge was used.

Bibliography

Alexander, F., Eyink, G., and Restrepo, J. (2005). Accelerated Monte Carlo for optimal estimation of time series. *Journal of Statistical Physics*, 119(5-6):1331–1345.

Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine Learning*, 50(1):5–43.

Archambeau, C., Opper, M., Shen, Y., Cornford, D., and Shawe-Taylor, J. (2008). Variational Inference for Diffusion Processes. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems 20*, pages 17–24. MIT Press, Cambridge, MA.

Bar-Even, A., Paulsson, J., Maheshri, N., Carmi, M., O'Shea, E., Pilpel, Y., and Barkai, N. (2006). Noise in protein expression scales with natural protein abundance. *Nature genetics*, 38(6):636–643.

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

Bishop, C. M. (2007). *Pattern Recognition and Machine Learning*. Springer-Verlag, New York, Inc., New York, USA, 1st edition.

Boys, R. J., Wilkinson, D. J., and Kirkwood, T. B. (2008). Bayesian inference for a discretely observed stochastic kinetic model. *Statistics and Computing*, 18(2):125–135.

Chen, K.-C., Wang, T.-Y., Tseng, H.-H., Huang, C.-Y. F., and Kao, C.-Y. (2005). A stochastic differential equation model for quantifying transcriptional regulatory network in Saccharomyces cerevisiae. *Bioinformatics*, 21(12):2883–2890.

Chib, S., Pitt, M. K., and Shephard, N. (2004). Likelihood based inference for diffusion driven models. Economics Papers 2004-W20, Economics Group, Nuffield College, University of Oxford.

Connors, K. A. (1990). *Chemical kinetics: The Study of Reaction Rates in Solution*. VCH Publishing, New York, USA, 1st edition.

D'Agostino, R. B. and Belanger, A. (1990). A Suggestion for Using Powerful and Informative Tests of Normality. *The American Statistician*, 44(4):316–321.

Dewar, M. A., Kadirkamanthan, V., Opper, M., and Sanguinetti, G. (2009). Parameter estimation and inference for stochastic reaction-diffusion systems: Application to morphogenesis in drosophila melanogaster. (to be published).

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208.

Driever, W. and Nüsslein-Volhard, C. (1988). The bicoid protein determines position in the Drosophila Embryo in a concentration-dependent manner. *Cell*, 54(1):95–104.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222.

Duffie, D., Pan, J., and Singleton, K. (2000). Transform analysis and asset pricing for affine jump-diffusions. *Econometrica*, 68(6):1343–1376.

Durham, G. B. and Gallant, A. R. (2001). Numerical techniques for maximum likelihood estimation of continuous-time diffusion processes. *Journal of Business and Economic Statistics*, 20:297–338.

Eckhardt, R. (1987). Stan Ulam, John von Neumann, and the Monte Carlo method. *Los Alamos Science*, (15):131–137.

Elerain, O., Chib, S., and Shephard, N. (2001). Likelihood inference for discretely observed nonlinear diffusions. *Econometrica*, 69(4):959–993.

Elerian, O., Chib, S., and Shephard, N. (1998). Likelihood inference for discretely observed non-linear diffusions. Technical report, Economics Group, Nuffield College, University of Oxford.

Elowitz, M. B., Levine, A. J., Siggia, E. D., and Swain, P. S. (2002). Stochastic gene expression in a single cell. *Science*, 297(5584):1183–1186.

Eraker, B. (2001). MCMC analysis of diffusion models with application to finance. *Journal of Business and Economic Statistics*, 19(2):177–191.

Ethier, S. N. and Kurtz, T. G. (1986). *Markov Processes: Characterization and Convergence*. John Wiley and Sons, New York, NY, USA; London, UK; Sydney, Australia.

Fearnhead, P. (2008). Computational methods for complex stochastic systems: a review of some alternatives to MCMC. *Statistics and Computing*, 18(2):151–171.

Gardiner, C. (2009). *Stochastic Methods*. Springer-Verlag Berlin Heidelberg, Berlin, Germany, 4th edition.

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57(6):1317–1339.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361.

Gillespie, D. T. (1992). A rigorous derivation of the chemical master equation. *Physica A: Statistical and Theoretical Physics*, 188(1-3):404–425.

Golightly, A. and Wilkinson, D. J. (2005). Bayesian inference for stochastic kinetic models using a diffusion approximation. *Biometrics*, 61(3):781–788.

Golightly, A. and Wilkinson, D. J. (2006). Bayesian sequential inference for nonlinear multivariate diffusions. *Statistics and Computing*, 16(4):323–338.

Golightly, A. and Wilkinson, D. J. (2008). Bayesian inference for nonlinear multivariate diffusion models observed with error. *Computational Statistics & Data Analysis*, 52(3):1674–1693.

Golightly, A. and Wilkinson, D. J. (2009). Markov chain Monte Carlo algorithms for SDE parameter estimation. In *Learning and Inference for Computational Systems Biology*, pages 253–275. MIT Press.

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

Henderson, D. A., Boys, R. J., Proctor, C. J., and Wilkinson, D. J. (2010). Linking systems biology models to data: a stochastic kinetic model of p53 oscillations. In *Handbook of Applied Bayesian Analysis*. Oxford University Press. in press.

Jiang, G. J. and Knight, J. L. (1997). A nonparametric approach to the estimation of diffusion processes, with an application to a short-term interest rate model. *Econometric Theory*, 13(05):615–645.

Jones, C. S. (1998). Bayesian estimation of continuous-time finance models. Working paper, Simon School of Business, University of Rochester.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Lagarias, J. C., Reeds, J. A., Wright, M. H., and Wright, P. E. (1998). Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9:112–147.

Lotka, A. (1924). *Elements of Physical Biology*. Williams and Wilkins, Baltimore.

MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

Malek-Mansour, M. and Houard, J. (1979). A new approximation scheme for the study of fluctuations in nonuniform nonequilibrium systems. *Physics Letters A*, 70(5-6):366–368.

Manninen, T., Linne, M.-L., and Ruohonen, K. (2006). Developing Itô stochastic differential equation models for neuronal signal transduction pathways. *Comput. Biol. Chem.*, 30(4):280–291.

McAdams, H. H. and Arkin, A. (1999). It's a noisy business! genetic regulation at the nanomolar scale. *Trends Genet*, 15(2):65–69.

Merton, R. C. (1970). Optimum consumption and portfolio rules in a continuous-time model. Working papers 58, Massachusetts Institute of Technology (MIT), Department of Economics.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092.

Metropolis, N. and Ulam, S. (1949). The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341.

Miller, R. N., Ghil, M., and Gauthiez, F. (1994). Advanced data assimilation in strongly nonlinear dynamical systems. *Journal of the Atmospheric Sciences*, 51(8):1037–1056.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Springer-Verlag New York, Inc., New York, NY, USA.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, 7(4):308–313.

Øksendal, B. (1992). *Stochastic Differential Equations: An Introduction with Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 3rd edition.

Opper, M. and Sanguinetti, G. (2008). Variational inference for Markov jump processes. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems 20*, pages 1105–1112. MIT Press, Cambridge, MA.

Papaspiliopoulos, O., Roberts, G. O., and Skôld, M. (2003). Non-centered parameterizations for hierarchical models and data augmentation. In Bernardo, J. M., Bayarri, M. J., Berger, J. O., Dawid, A. P., Heckerman, D., Smith, A. F. M., and West, M., editors, *Baysian Statistics 7*, pages 307–326. Oxford University Press.

Pedersen, A. R. (1995). A new approach to maximum likelihood estimation for stochastic differential equations based on discrete observations. *Scandinavian Journal of Statistics*, 22(1):55–71.

Petersen, K. B. and Pedersen, M. S. (2008). The matrix cookbook. Version 20081110, visited at 12th of February 2010, available at http://www2.imm.dtu.dk/pubdb/p.php?3274.

Roberts, G. and Stramer, O. (2001). On inference for partially observed nonlinear diffusion models using the Metropolis-Hastings algorithm. *Biometrika*, 88:603–621.

Ruttor, A., Sanguinetti, G., and Opper, M. (2009). Approximate inference for stochastic reaction processes. In *Learning and Inference in Computational Systems Biology*, pages 189–205. MIT Press.

Scott, S. L. (2002). Bayesian methods for hidden Markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, 97(457):337–351.

Shen, Y., Archambeau, C., Cornford, D., Opper, M., Shawe-Taylor, J., and Barillec, R. (2009). A comparison of variational and markov chain monte carlo methods for inference in partially observed stochastic dynamic systems. *Journal of Signal Processing Systems*.

Shephard, N. and Pitt, M. K. (1997). Likelihood analysis of non-Gaussian measurement time series. *Biometrika*, 84:653–667.

Slepchenko, B. M., Schaff, J. C., Carson, J. H., and Loew, L. M. (2002). Computational cell biology: Spatiotemporal simulation of cellular events. *Annual Review of Biophysics and Biomolecular Structure*, 31:423–441.

Smith, S. P. and Jain, A. K. (1988). A test to determine the multivariate normality of a data set. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):757–761.

Sørensen, H. (2004). Parametric inference for diffusion processes observed at discrete points in time: a survey. *International Statistical Review*, 72(3):337–354.

Tanner, M. A. and Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398):528–540.

van Kampen, N. G. (1961). A power series expansion of the master equation. *Canadian Journal of Physics*, 39:551–567.

van Kampen, N. G. (2007). *Stochastic Processes in Physics and Chemistry*. North Holland, Amsterdam, The Netherlands, 3rd edition.

Volterra, V. (1926). Fluctuations in the abundance of a species considered mathematically. *Nature*, 118:558–560.

Wu, Y. F., Myasnikova, E., and Reinitz, J. (2007). Master equation simulation analysis of immunostained bicoid morphogen gradient. *BMC Systems Biology*, 1(1):52.