DELFT UNIVERSITY OF TECHNOLOGY

REPORT 03-06

Probability Density Estimation in Stochastic Environmental Models using Reverse Representations

E. van den Berg, A.W. Heemink, H.X. Lin,
and J.G.M. Schoenmakers

ISSN 1389-6520

Reports of the Department of Applied Mathematical Analysis

Delft 2003

Probability density estimation in stochastic environmental models using reverse representations

E. van den Berg
A.W. Heemink*
H.X. Lin

Delft University of Technology
Faculty of Information Technology and Systems
Department of Applied Mathematical Analysis
Mekelweg 4, 2628 CD Delft, The Netherlands

J.G.M. Schoenmakers

Weierstrass Institute for Applied Analysis and Stochastics
Mohrenstrasse 39, D-10117 Berlin, Germany

May 23, 2003

Abstract

The estimation of probability densities of variables described by systems of stochastic differential equations has long been done using forward time estimators, which rely on the generation of realizations of the model, forward in time. Recently, an estimator based on the combination of forward and reverse time estimators has been developed. This estimator has a higher order of convergence than the classical approach. In this article, we explore the new estimator and compare the forward and forward-reverse estimators by applying them to a biochemical oxygen demand model. Finally, we show that the computational complexity of the forward-reverse estimator is superior to the classical approach, and discuss the algorithmic aspects of the estimator.

*Corresponding author, E-Mail: A.W.Heemink@its.tudelft.nl, Phone: +31-15-2785813, Fax: +31-15-2787295

1 Introduction

Many environmental models used today are based on deterministic differential equations (Loucks, Stedinger, and Haith 1981). However, the abilities of these models for prediction in real life applications, are often hampered by uncertainties in initial conditions, model parameters and/or sources. Decision makers have also recognized the inherent uncertainty in all modeling efforts. Therefore, they are more and more asking for risk estimates. For example, in a water pollution problem, they want to know what is the probability of exceeding a critical concentration of a substance.

Uncertainties can be included into the model by introducing noise processes as forcing terms. As a result a stochastic differential equation is obtained (Jazwinsky 1970). In general stochastic differential equations cannot be solved analytically and have to be approximated numerically to derive a discrete stochastic model (Kloeden and Platen 1992).

A well-known technique for estimating probabilistic characteristics of a stochastic model is Monte Carlo simulation (Hammersley and Handscomb 1964). Here many different realizations of the stochastic model are generated to get information about the probability density of the model results. This approach is conceptually very simple and can be applied for many different types of (highly nonlinear) problems. However, Monte Carlo techniques do consume a large amount of CPU time. One of the reasons for this is that many of the realizations hardly contribute to the final risk estimate and that this is known only after the samples have been generated. Another reason is that the accuracy of the estimates improves only slowly with the increase of sample size.

The efficiency of Monte Carlo methods can be improved by variance reduction (Kloeden and Platen 1992). Here an approximation of the Kolmogorov backwards equation is required. In some simple applications analytical approximations of this partial differential equation can be used. In general, however, a numerical solution is required. For high dimensional systems this may become very time consuming (Schoenmakers, Heemink, Ponnambalam, and Kloeden 2002), (Milstein and Schoenmakers 2002).

Recently, (Milstein, Schoenmakers, and Spokoiny 2002) introduced the concept of reverse time diffusion. The classical Monte Carlo estimate is based on forward realizations of the original stochastic model. (Milstein, Schoenmakers, and Spokoiny 2002) derived a reverse system from the original model and showed that the classical Monte Carlo estimate can also be based on realization of this reverse system. For many applications it is more efficient to use realizations of the reverse system instead of the original model. The most efficient implementation is, however, obtained if the forward realizations and the reverse system realizations are combined. (Milstein, Schoenmakers, and Spokoiny 2002) called this the Forward Reverse Estimator.

In this paper we apply the forward reverse method to estimate the probability density of a stochastic Biochemical-Oxygen Demand (BOD) model. In Section 2 we describe the classical Monte Carlo estimation, present a general probabilistic representation of this estimator based on forward realizations and introduce the Forward Reverse Estimator. In Section 3 the BOD model is given, and its reverse model is

derived. We describe in detail an application of the method to estimate the probability density of the BOD model in Section 4 and compare the performance of the forward with forward-reverse estimators in Section 5.

2 Transition density estimators for SDEs

Consider the stochastic differential equation (SDE) in the Itô sense

$$dX = a(s,X)ds + \sigma(s,X)dW(s), \qquad t_0 \le s \le T, \qquad X(t_0) = x_0 \tag{1}$$

where $X = (X^1, ..., X^d)^\top$, $a = (a^1, ..., a^d)^\top$ are d-dimensional vectors, $W = (W^1, ..., W^m)^\top$ is an m-dimensional standard Wiener process, and $\sigma = \{\sigma^{ij}\}$ is a $d \times m$-matrix, $m \ge d$. We assume that the $d \times d$-matrix $b := \sigma\sigma^\top$, $b = \{b^{ij}\}$, is of full rank for every (s,x), $s \in [t_0, T]$, $x \in R^d$. The functions $a^i(s,x)$ and $\sigma^{ij}(s,x)$ are assumed to be sufficiently good in analytical sense (for example, their first derivatives are continuous and bounded) such that we have existence and uniqueness of the solution $X_{t,x}(s) \in R^d$, $X_{t,x}(t) = x$, $t_0 \le t \le s \le T$, of (1), smoothness of the transition density $p(t,x,s,y)$ of the Markov process X, and existence of all the moments of $p(\cdot,\cdot,\cdot,y)$.

The solution of SDE (1) may be approximated by different numerical methods, e.g. see (Kloeden and Platen 1992), (Milstein 1995). Here we consider the Euler scheme,

$$X(t_{k+1}) = X(t_k) + a(t_k, X(t_k))(t_{k+1} - t_k) + \sigma(t_k, X(t_k))\sqrt{t_{k+1} - t_k}\ \varsigma_k, \tag{2}$$

with $t_0 < t_1 < \cdots < t_K = T$, and $\varsigma_k \in R^m$, $k = 0, \ldots, K-1$, being i.i.d. standard normal random variables. In fact, (2) is the most simple numerical scheme for integration of SDE (1).

2.1 Classical and non-classical density estimators

We now review some known estimators for the density $p(t,x,T,y)$ in connection with (1).

A. The Parzen-Rosenblatt (pure forward or classical) estimator.

Let $\bar{X}_{t,x}$ be a numerical approximation of $X_{t,x}$ obtained by the Euler method and let $\overline{X}_n := \bar{X}^{(n)}_{t,x}(T)$, $n = 1, \ldots, N$, be a sample of independent realizations of $\bar{X}_{t,x}(T)$. Then one may estimate the transition density $p(t,x,T,y)$ from this sample by using standard techniques of non-parametric statistics such as the classical kernel (Parzen-Rosenblatt) estimator. The kernel (Parzen-Rosenblatt) density estimator with a kernel K and a bandwidth δ is given by

$$\hat{p}_{FE}(t,x,T,y) = \frac{1}{N\delta^d} \sum_{n=1}^{N} K\left(\frac{\overline{X}_n - y}{\delta}\right), \tag{3}$$

see (Devroye and Györfi 1985), (Silverman 1986). For example, in (3) one could take the Gaussian kernel $K(x) = (2\pi)^{-d/2} \exp(-|x|^2/2)$.

B. The pure reverse estimator (RE).

In order to proceed with more sophisticated density estimators we introduce a reverse diffusion system for (1). In a more special setting the notion of reverse or adjoint diffusion is due to Thomson (Thomson 1987). The here introduced reverse system can be seen as a generalization of Thomson's approach and is derived in (Milstein, Schoenmakers, and Spokoiny 2002) in a more transparent *and* more rigorous way.

We first introduce a reversed time variable $\tilde{s} = T + t - s$ and define the functions

$$\begin{aligned} \tilde{a}^i(\tilde{s}, y) &= a^i(T + t - \tilde{s}, y), \\ \tilde{b}^{ij}(\tilde{s}, y) &= b^{ij}(T + t - \tilde{s}, y). \end{aligned}$$

For a vector process $Y_{t,y}(s) \in I\!R^d$ and a scalar process $\mathcal{Y}_{t,y}(s)$ we then consider the reverse time stochastic system

$$\begin{array}{ll} dY = \alpha(s, Y)ds + \tilde{\sigma}(s, Y)d\tilde{W}(s), & Y(t) = y, \\ d\mathcal{Y} = c(s, Y)\mathcal{Y}ds, & \mathcal{Y}(t) = 1, \end{array} \tag{4}$$

with $\tilde{W}$ being an m-dimensional standard Wiener process and

$$\alpha^i(s, y) = \sum_{j=1}^{d} \frac{\partial \tilde{b}^{ij}}{\partial y^j} - \tilde{a}^i, \tag{5}$$

$$c(s, y) = \frac{1}{2} \sum_{i,j=1}^{d} \frac{\partial^2 \tilde{b}^{ij}}{\partial y^i \partial y^j} - \sum_{i=1}^{d} \frac{\partial \tilde{a}^i}{\partial y^i}, \tag{6}$$

$$\tilde{\sigma}(s, y) = \sigma(T + t - s, y). \tag{7}$$

It is possible to construct an alternative density estimator in terms of the reverse system (4). Suppose that $(\bar{Y}_{0,y}^{(m)}, \bar{\mathcal{Y}}_{0,y}^{(m)})$, $m = 1, ..., M$, is an i.i.d. sample of numerical solutions of (4) starting at $t = 0$, obtained by the Euler scheme. Then a pure reverse estimator is given by

$$\hat{p}_{RE}(t, x, T, y) := \frac{1}{M\delta^d} \sum_{m=1}^{M} K\left(\frac{x - \bar{Y}_{0,y}^{(m)}(T)}{\delta}\right) \bar{\mathcal{Y}}_{0,y}^{(m)}(T). \tag{8}$$

In fact, the reverse estimator (8) can be obtained as a side case from the forward-reverse estimator discussed discussed in C.

C. The forward-reverse estimator (FRE).

By combining the forward (1) and reverse (4) estimators via the Chapman-Kolmogorov equation with respect to an intermediate time t^*, Milstein, Schoenmakers, and Spokoiny constructed the so called forward-reverse estimator,

$$\hat{p}_{FRE}(t, x, T, y) = \frac{1}{M}\left[\frac{1}{N\delta^d} \sum_{m=1}^{M} \sum_{n=1}^{N} K\left(\frac{\bar{X}_{t,x}^{(n)}(t^*) - \bar{Y}_{t^*,y}^{(m)}(T)}{\delta}\right) \bar{\mathcal{Y}}_{t^*,y}^{(m)}(T)\right]. \tag{9}$$

In (Milstein *et al.* 2002) it is shown that the forward-reverse estimator (9) has superior properties in comparison with density estimators based on pure forward (3) or pure reverse (8) representations. Obviously, by taking $t^* = T$ and $t^* = 0$, the estimator (9) collapses to the pure forward estimator (3) and pure reverse estimator (8), respectively.

Remark. In (Milstein, Schoenmakers, and Spokoiny 2002) a forward-reverse projection estimator is also studied. Although the projection estimators has in some sense additional advantages, we only consider applications of (9) in this article, and refer to (Milstein, Schoenmakers, and Spokoiny 2002) for details with respect to projection estimators.

2.2 Accuracy and complexity of the different estimators

In general, for estimating a target value p by an estimator $\hat{p}$ it is natural and usual to define the accuracy of the estimator by

$$Accuracy(\hat{p}) := \epsilon(\hat{p}) := \sqrt{E(\hat{p}-p)^2} = \sqrt{Deviation^2(\hat{p}) + Bias^2(\hat{p})}. \qquad (10)$$

Loosely speaking, for a second order kernel applied in (9) and any choice of $0 < t^* < T$, the FRE has root-N ($\mathcal{O}(N^{-1/2})$) accuracy for dimension $d \leq 4$. For $d > 4$ root-N accuracy is lost but then the FRE accuracy order is still the square of the FE/RE accuracy order (see Table 1). Moreover, it can be shown that root-N accuracy of (9) can also be achieved for $d > 4$ by using higher order kernels in (9).

By definition (10) it is possible to relate the "expected" accuracy of the different density estimators to the number of simulated trajectories involved. However, simulating trajectories is not the only costly issue in the density estimation. For all estimators one has to evaluate a functional of the simulated trajectories. In case of the FE and RE estimators, this functional consists of a single summation, whereas for the FRE estimator a more complicated double summation needs to be evaluated. Therefore, for a proper comparison it is better to consider the *complexity* of the different estimators which is defined as the required computation cost for reaching a given accuracy ϵ. For instance, naive evaluation of the double sum in (9) would require a computational cost of order $\mathcal{O}(MN)$ in contrast to $\mathcal{O}(N)$ for the FE and RE estimators! Clearly, such an naive approach would have a dramatic impact on the complexity of the FRE. Fortunately, smarter procedures for evaluating this double sum exist, which utilize the small support of the kernel K. Particularly, in (Greengard and Strain 1991) it is shown for Gaussian kernels that this sum can be evaluated at a cost of $\mathcal{O}(N)$ in case $M = N$. In Table 1 we summarize the results of (Milstein, Schoenmakers, and Spokoiny 2002) and list the accuracy and complexity of the forward estimator (FE), reverse estimator (RE) and the forward-reverse estimator (FRE) where the latter is implemented with an efficient procedure for evaluating the double sum, for instance, according to the method of Greengard and Strain. For the FRE estimator we assumed $N = M$ and a second order kernel. It is because of the second order kernel that we have to distinguish for the FRE between $d < 4$ and $d \geq 4$.

Remark. Naturally, all estimators FE, RE and FRE have a bias which involves a component due to application of the Euler scheme. Using results of (Bally and

Estimator	δ_N	*Accuracy*	*Complexity*	$\frac{Compl.\{FE,RE\}}{Compl.\{FRE\}}$
FE/RE	$N^{-1/(4+d)}$	$\mathcal{O}(N^{-2/(4+d)})$	$\mathcal{O}(\epsilon^{-2-d/2})$	
FRE $d \leq 4$	$N^{-1/d}\log^{1/d} N$	$\mathcal{O}(N^{-1/2})$	$\mathcal{O}(\lvert\log\varepsilon\rvert\epsilon^{-2})$	$\lvert\log\varepsilon\rvert^{-1}\epsilon^{-d/2}$
FRE $d > 4$	$N^{-2/(4+d)}$	$\mathcal{O}(N^{-4/(4+d)})$	$\mathcal{O}(\lvert\log\varepsilon\rvert\epsilon^{-1-d/4})$	$\lvert\log\varepsilon\rvert^{-1}\epsilon^{-1-d/4}$

Table 1: Summary of accuracy and complexity of the forward (FE), reverse (RE), and forward-reverse (FRE) estimators.

Talay 1996b) and (Bally and Talay 1996a), it is proven in (Milstein, Schoenmakers, and Spokoiny 2002) that for all estimators the accuracy loss due to the Euler scheme applied with time discretization step h is of order $\mathcal{O}(h)$ where, most importantly, the order coefficient may be chosen independent of the bandwidth δ.

2.3 The choice of t^* in the Forward Reverse Estimator

The results concerning *the order* of accuracy and complexity of the FRE estimator in Section 2.2 do not depend on the particular choice of t^* with $t < t^* < T$. However, the order coefficients do depend on this choice. To investigate this in more detail we consider Theorem 6.1. in (Milstein, Schoenmakers, and Spokoiny 2002) for $d < 4$. According to this theorem we have for $M = N$ and $\delta_N = N^{-1/d}\log^{1/d} N$,

$$E(\hat{p}_{FRE} - p)^2 = \frac{D}{N} + o(N^{-1})$$

with

$$\begin{aligned} D &:= \int r(u)\lambda^2(u)du + \int r^2(u)\mu_2(u)\,q(u)\,du - 2p^2 \\ &= \int r(u)q^2(u)\mu^2(u)du - p^2 + \int r^2(u)\mu_2(u)\,q(u)\,du - p^2 \\ &= D_{(1)} + D_{(2)}, \end{aligned} \tag{11}$$

where $r(u) := p(t,x,t^*,u)$ is the density of the random variable $X_{t,x}(t^*)$, $q(u)$ denotes the density of $Y_{t^*,y}(T)$, $\mu(u)$ is defined as the conditional mean of $\mathcal{Y}_{t^*,y}(T)$ given $Y_{t^*,y}(T) = u$, $\lambda(u) := q(u)\mu(u)$, and $\mu_2(u) := E(\mathcal{Y}^2_{t^*,y}(T) \mid Y_{t^*,y}(T) = u)$. We so have in (11),

$$\begin{aligned} D_{(1)} &= \int r(u)q^2(u)\mu^2(u)du - \left(\int r(u)q(u)\mu(u)du\right)^2 \\ &= Var\left(q(X_{t,x}(t^*))\mu(X_{t,x}(t^*)\right) =: f(t^*) \end{aligned}$$

and

$$\begin{aligned} D_{(2)} &= \int r^2(u)E[\mathcal{Y}^2_{t^*,y}(T) \mid Y_{t^*,y}(T) = u]\,q(u)\,du - p^2 \\ &= E[r^2(Y_{t^*,y}(T))\mathcal{Y}^2_{t^*,y}(T)] - \left(E[r(Y_{t^*,y}(T))\mathcal{Y}_{t^*,y}(T)]\right)^2 \\ &= Var(r(Y_{t^*,y}(T))\mathcal{Y}_{t^*,y}(T)) =: g(t^*). \end{aligned}$$

Since for $t^* \uparrow T$, $q \to \delta_y$ and for $t^* \downarrow t$, $r \to \delta_x$, it follows that

$$\begin{aligned} \lim_{t^* \uparrow T} f(t^*) &= +\infty, \\ \lim_{t^* \downarrow t} g(t^*) &= +\infty. \end{aligned}$$

Further,

$$\lim_{t^* \uparrow T} g(t^*) = \lim_{t^* \downarrow t} f(t^*) = 0.$$

Therefore, there exists an optimal choice t^*_{opt} which satisfies

$$f(t^*_{opt}) + g(t^*_{opt}) = \min_{t<t^*<T} \{f(t^*) + g(t^*)\}.$$

However, determination of such an exact optimum is not easy and in practice not necessary also. One rather should seek by experiment for a t^* such that both $f(t^*)$ and $g(t^*)$ are small. For instance, one could choose some candidates for t^* and compare them by estimating the variances $f(t^*)$ and $g(t^*)$ roughly via classical (e.g. Parzen-Rozenblatt) approximations for q and r using small sample sizes. In Section 4.3 a heuristic method for the determination of t^* will be presented.

Without going into detail we note that similar considerations concerning a proper choice of t^* apply for the case $d > 4$.

3 Test case: a stochastic BOD model

The biochemical oxygen demand model presented in this section is often used in water quality modeling of river and estuarine systems and is an extension of the so-called Streeter-Phelps model. In this model, the concentration levels of Carbonaceous Biochemical Oxygen Demand (CBOD), Dissolved Oxygen (DO), and Nitrogenous Biochemical Oxygen Demand (NBOD) are related to each other by a set of differential equations. With b, o, and n representing the levels (mg/l) of CBOD, DO, and NBOD, respectively, the system is given by

$$\begin{bmatrix} db/dt \\ do/dt \\ dn/dt \end{bmatrix} = \begin{bmatrix} -k_b & 0 & 0 \\ -k_c & -k_2 & -k_n \\ 0 & 0 & -k_n \end{bmatrix} \begin{bmatrix} b \\ o \\ n \end{bmatrix} + \begin{bmatrix} s_1 \\ s_2^* \\ s_3 \end{bmatrix}. \tag{12}$$

where we defined $k_b := k_c + k_3$ and $s_2^* := k_2 d_s + p - r + s_2$. A description of the parameters used in the model, along with their units and typical values is given in Table 2. Although the concentrations are time dependent, the intention behind the model is often to monitor the concentration levels within a fixed volume of water, flowing downstream a river. An underlying assumption is that the velocity of the water flow is constant and thus time and distance are linearly related. For more information about the model and its background, the reader is referred to e.g. (Stijnen 2002).

Par.	Description	Unit	Value
k_c	Reaction rate coefficient	(l/day)	0.763
k_2	Reaeration rate coefficient	(l/day)	4.250
k_3	Sedimentation and adsorption loss rate for CBOD	(l/day)	0.254
k_n	Decay rate of NBOD	(l/day)	0.978
p	Photosynthesis of oxygen	(l/day)	7.280
r	Respiration of oxygen	(l/day)	7.750
d_s	Saturation concentration of oxygen	(l/day)	10.00
s_1	Independent source for CBOD	(l/day)	3.000
s_2	Independent source for OD	(l/day)	0.000
s_3	Independent source for NBOD	(l/day)	0.000

Table 2: Description and typical values for the parameters used in the BOD model

3.1 Forward BOD model

As for many environmental models, the BOD model is also subject to various uncertainties. These uncertainties can be incorporated into the model by adding white noise processes to each of the three equations of the model, effectively changing deterministic sources and sinks into stochastic ones:

$$\begin{bmatrix} dB \\ dO \\ dN \end{bmatrix} = \begin{bmatrix} -k_b & 0 & 0 \\ -k_c & -k_2 & -k_n \\ 0 & 0 & -k_n \end{bmatrix} \begin{bmatrix} B \\ O \\ N \end{bmatrix} dt + \begin{bmatrix} s_1 \\ s_2^* \\ s_3 \end{bmatrix} dt + \begin{bmatrix} \sigma_B & 0 & 0 \\ 0 & \sigma_O & 0 \\ 0 & 0 & \sigma_N \end{bmatrix} dW_t, \tag{13}$$

where the dW_t term denotes the Wiener process increment at time t. The stochastic BOD model is Markov and its variables Gaussian. Because the noise terms added do not depend on the values of B, O, or N, the model can be interpreted in both the Itô and Stratonovich sense. The stochastic equation (13) is linear, and an analytical solution is easily found.

A disadvantage of the simple additive white noise terms, is the possibility of getting negative concentration levels. This problem can be solved by scaling the noise term using a suitably chosen function, eg.:

$$s(x) = \begin{cases} 0 & x < 0 \\ \frac{1}{2}\left(\frac{2x}{\tau}\right)^d & 0 \leq x < \tau/2 \\ 1 - \frac{1}{2}\left(\frac{2(\tau - x)}{\tau}\right)^d & \tau/2 \leq x < \tau \\ 1 & \tau \leq x \end{cases}$$

with $d > 0$ the order of the polynomial used, and τ the threshold below which the noise should be damped. Applying this to the BOD model would result in the following set of equations

$$\begin{bmatrix} dB \\ dO \\ dN \end{bmatrix} = \begin{bmatrix} -k_b & 0 & 0 \\ -k_c & -k_2 & -k_n \\ 0 & 0 & -k_n \end{bmatrix} \begin{bmatrix} B \\ O \\ N \end{bmatrix} dt + \begin{bmatrix} s_1 \\ s_2^* \\ s_3 \end{bmatrix} dt + \begin{bmatrix} s(B)\sigma_B & 0 & 0 \\ 0 & s(O)\sigma_O & 0 \\ 0 & 0 & s(N)\sigma_N \end{bmatrix} dW_t,$$

Although this solves the problem of negative concentrations, the linearity of the model is lost, along with the analytical solution and the Gaussian property. Being a test case for the forward-reverse estimator, the presence of an analytical solution of the presented model outweighs the disadvantage of a reduced practical meaning, as it allows for verification of the outcome. Therefore, it was decided to keep the simple stochastic model despite its obvious shortcoming.

Given this model, we want to determine $p(t, x, T, y)$, i.e. the probability density associated with the transition between a given start (x) and end point (y) indicated by two $(B, O, N)^{\mathrm{T}}$ vectors, at two distinct times $t \leq T$. Based on the forward model, it is possible to apply (3), which requires a set of realizations of the model. These can be obtained from numerical integration of (13), as shown in Figure 1a.

3.2 Reverse BOD model

In order to apply a forward-reverse estimator (Section 2.1C), the reverse representation of the forward model needs to be derived. Since the presented BOD model is already given in the form prescribed by (1), straightforward substitution yields,

$$a(t, X) = \begin{bmatrix} -k_b X^{(1)} + s_1 \\ -k_c X^{(1)} - k_2 X^{(2)} - k_n X^{(3)} + s_2^* \\ -k_n X^{(3)} + s_3 \end{bmatrix},$$

and

$$\sigma(t, X) = \begin{bmatrix} \sigma_B & 0 & 0 \\ 0 & \sigma_O & 0 \\ 0 & 0 & \sigma_N \end{bmatrix}, \quad b = \sigma\sigma^{\mathrm{T}} = \begin{bmatrix} \sigma_B^2 & 0 & 0 \\ 0 & \sigma_O^2 & 0 \\ 0 & 0 & \sigma_N^2 \end{bmatrix}.$$

Next, using equations (5) and (6), the expressions for α and c can be derived to be as follows:

$$\alpha(t, X) = \begin{bmatrix} k_b X^{(1)} - s_1 \\ k_c X^{(1)} + k_2 X^{(2)} + k_n X^{(3)} - s_2^* \\ k_n X^{(3)} - s_3 \end{bmatrix},$$

$$c = k_b + k_2 + k_n.$$

with $X^{(1)} = B$, $X^{(2)} = O$, and $X^{(3)} = N$. For the reverse σ process we have $\tilde{\sigma} = \sigma$. Now, with all variable terms in equation (4) known, the final reverse process becomes

$$dY = \begin{bmatrix} k_b X^{(1)} - s_1 \\ k_c X^{(1)} + k_2 X^{(2)} + k_n X^{(3)} - s_2^* \\ k_n X^{(3)} - s_3 \end{bmatrix} ds + \begin{bmatrix} \sigma_B & 0 & 0 \\ 0 & \sigma_O & 0 \\ 0 & 0 & \sigma_N \end{bmatrix} d\tilde{W}(s), \tag{14}$$

with initial condition

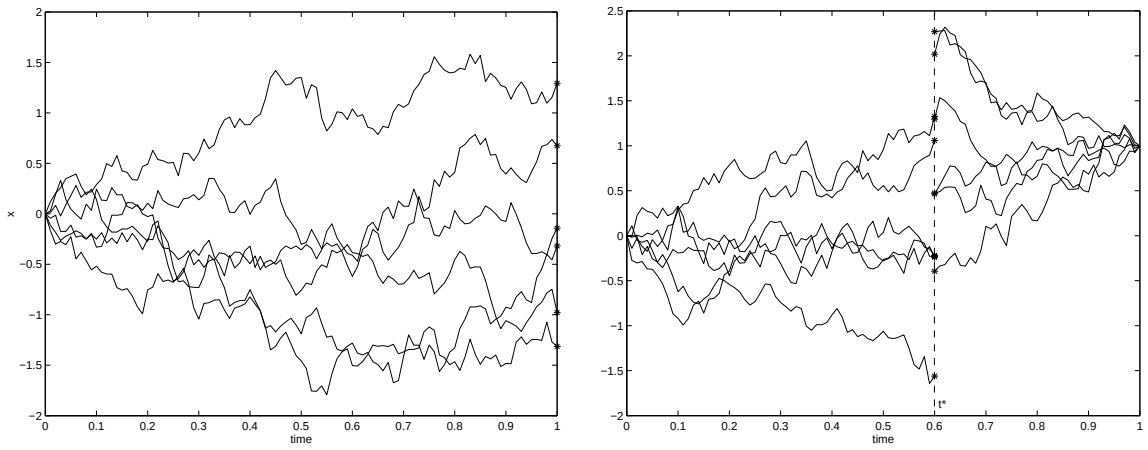
$$Y(t_0) = y, \tag{15}$$

and

$$d\mathcal{Y} = (k_b + k_2 + k_n)\mathcal{Y} dt, \tag{16}$$

with

$$\mathcal{Y}(t_0) = 1. \tag{17}$$



(a) **(b)**

Figure 1: Realization tracks of **(a)** the forward process X, and **(b)** the X and Y processes used in the forward-reverse approach. The asterisk symbols indicate the end-points of the respective tracks, i.e. the realizations that are used in the Monte Carlo estimator.

With all constants in the reverse model positive, both the Y and $\mathcal{Y}$ processes are instable, and grow exponentially. This relation is true in general; processes that are stable in the forward formulation will be instable in the reverse formulation and vice versa, while oscillating processes will remain so in both formulations.

As for the forward formulation of the given BOD model, the reverse formulation is a linear system of equations, and the closed-form analytical solution is easily derived.

How exactly the forward-reverse estimator is applied to obtain the probability density given earlier, is explained in the next section.

4 Application of the Forward-Reverse Estimator

One way to carry out the forward-reverse estimation as presented in Section 2.1 is by means of a Monte Carlo simulation. To do so, a number of realizations for both the forward and reverse models are generated (eg. for the CBOD model this is done by numerically integrating Equation (13) for the forward part and Equations (14) and (16) for the reverse part). Based on the forward realizations, kernel estimation techniques are used to evaluate the transition density from the starting point of the simulation to the end point of each of the realizations of the reverse part, as illustrated by Figure 1. There are a number of choices to be made for the implementation of the algorithm. These will be discussed below.

4.1 Numerical integration

It is important to note that in the context of forward-reverse estimation, two sources of errors other than the kernel estimation exist. The first one is due to the use of numerical schemes, while the other originates from the use of Monte Carlo simulation. Numerous integration schemes can be used to generate numerical solutions of a stochastic differential equation. For our experiments, we used the Euler scheme, which was given by (2). The error of the numerical scheme can be reduced by using higher order schemes. The statistical error of the Monte Carlo simulation can be reduced by increasing the number of tracks generated. Obviously, there should be a balance between these two sources of error. Reducing only one of the two will not always attribute to a reduction of the global error. As such, there should also be a balance between the stepsize used in the numerical scheme and the number of tracks generated, which is discussed in more detail in (Schoenmakers and Heemink 1997).

4.2 Kernel estimation

The two most important aspects of kernel estimation are the kernel shape used, and the scale of it, as determined by the bandwidth parameter. These aspects are in close connection with the computational effort required to carry out a kernel estimation. Although of minor importance for single point evaluations, as used in forward density estimation, where a kernel estimation based on N realizations can easily be implemented in $\mathcal{O}(N)$ time complexity, it is of major importance in cases where the approximated density function needs evaluation at a large number of points. This is the case for forward-reverse density estimation. Based on N forward realizations, it produces the first part of the transition density for M reverse realizations, which are themselves the second part of the density estimation. Naive algorithms easily lead to an $\mathcal{O}(N \cdot M)$ computational time, which is clearly prohibitively expensive. Greengard and Strain (Greengard and Strain 1991) developed an optimal algorithm for Gaussian kernels, based on the properties of Gaussian curves, that leads to an $\mathcal{O}(N + M)$ time complexity. One known drawback of their method, however, is the large constant multiplication factor. Unless the number of sample and evaluation points is very large, this constant will cause the computational time to be exceedingly long. We therefore decided to aim for a more modest $\mathcal{O}(M \log N)$ algorithm, with lower overhead, instead.

By using a parabolic kernel, the domain of influence of a realization is limited, as opposed to the Gaussian kernel which has an unlimited support. Because of this limited support, most of the $N \cdot M$ kernel evaluations will be zero, which means that those realizations will not contribute to the probability density estimation at the given location. Based on this observation and the fact that our parabolic kernel only has a support radius of, say, ϕ, it is possible to group the realizations in such a way that most of the zero kernel evaluations can be avoided, saving a lot of work. Given a set of N forward realizations, x_i, in a d-dimensional space, with $1 \leq i \leq N$, we proceed as follows.

1. Find the bounding box enclosing all realizations;

2. Create a regular grid such that the length of each side equals the kernel support, ϕ;
3. Record which realizations are contained in which grid cell;

Once this information is known, the kernel estimation of a point at location y can be obtained by following the procedure outlined below

1. Determine the cell containing y. Note that this cell does not necessarily be an existing cell, nor does it need to be inside the bounding box;
2. Find all cells that share at least one cornerpoint with the aforementioned cell. This way, a total of 3^d cells are selected, including the cell containing y.
3. Retrieve all samples stored in the selected cells and use them to get a kernel estimation.

The distance from y to a particle contained in any other cell is guaranteed to exceed the support range, ϕ, because of the choice of cell size. Therefore, no information is lost, and the resulting kernel estimation is exactly the same as the one obtained using a 'brute-force' approach. In case of the forward-reverse estimator, the above three steps are simply repeated for each of the reverse realizations.

Some remarks regarding the part of the algorithm where the datastructure is set up are appropriate. This datastructure stores the cells and the realizations contained in them. Once constructed, the datastructure can be used to support fast evaluation of the kernel estimation based on these realizations.

It is important to realize that only those cells containing realizations should be stored, which means a maximum of N cells. Especially when the diffusion of realizations is large, compared to the kernel support, and the dimension of the problem is high, the potential number of cells that would otherwise be stored is enormous. To allow for both the selective storage and fast lookup of cells, a hashtable construction was used. Each entry in the table contains cells sharing the same hash value. The hash value is based on the coordinates of the cell which are carefully combined to allow for an even distribution of cells over table entries. Because the number of cells stored in each table entry is not known *a priory* and differs between entries, each entry merely contains a pointer to a linked list of cells. For similar reasons, each cell only contains a pointer to a linked list of the realizations it encloses, as shown in Figure 2.

Retrieval of cells from the datastructure proceeds in a series of steps. First, the hash value based on the cell's coordinates is computed. This value is then used to index the hashtable and access the list of all cells stored under that particular number. Next, the list is traversed and the coordinates of each cell compared to the given coordinates, until either a match was found, or the end of the list reached. The realizations stored by the given cell can be retrieved likewise.

Although the above algorithm can be shown to have a worst-case $\mathcal{O}(N \cdot M)$ time complexity, it performed well in our experiments, showing the desired $\mathcal{O}(M \log N)$ behavior in most cases. In the cases where this time complexity was not achieved, the support to realization dispersion ratio was found to be high. That is, either the support of the kernel was too large or the realizations were highly concentrated in

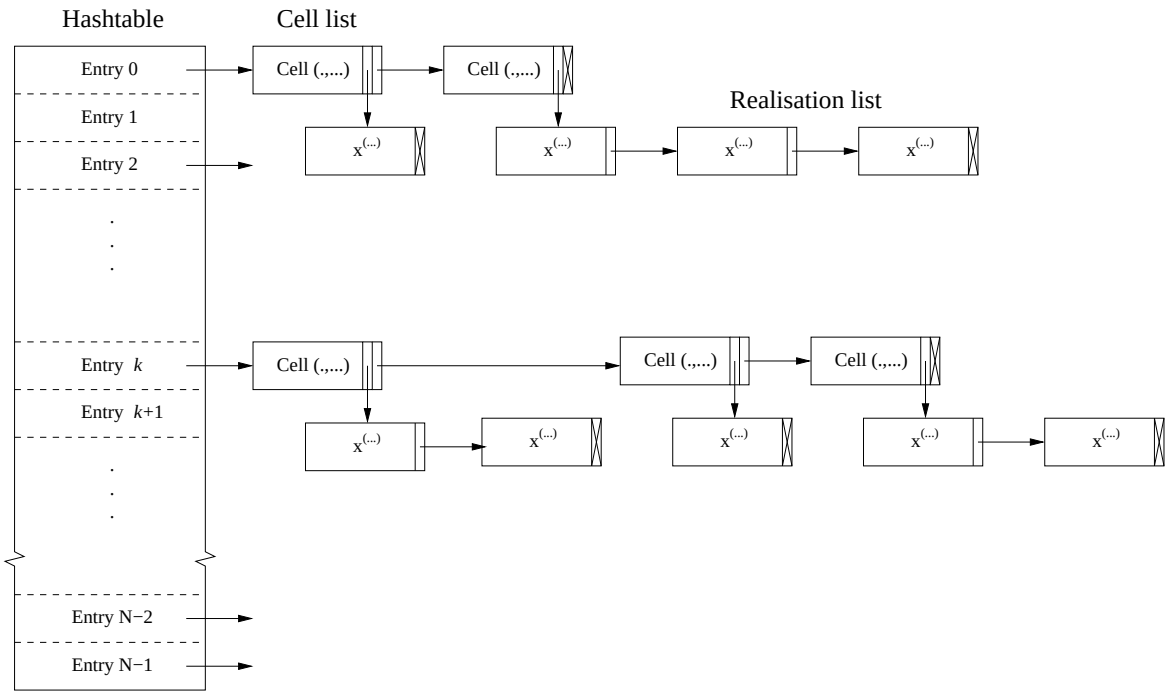


Figure 2: Datastructure used for storing realizations on which a kernel evaluation is based.

a small region. Ultimately, it comes down to the problem that all realizations are contained in a small number of adjacent cells. When a kernel evaluation is made in nearby regions, most, if not all, realizations are included in the kernel evaluation leading to worst-case times. In the forward-reverse estimator, the optimal bandwidth used for kernel estimation depends on the number of forward realizations:

$$\delta_{_N} = N^{-1/d} \log^{1/d} N. \tag{18}$$

When the number of realizations is increased, the bandwidth and cellsize decrease and, as a result, the number of kernel evaluations made is reduced. Even for our three dimensional BOD model, $\mathcal{O}(M \log N)$ kernel estimations were made with N well below $10,000$.

As a final remark it should be noted that the above technique can also be applied to Gaussian kernels, as long as their support is limited. This is done by cutting off the kernel at a distance of several standard deviations from the kernel center, where the influence is considered not to significantly attribute to the kernel estimation. In contrast to the use of the parabolic kernel, this speedup is realized at the expense of a loss in accuracy. Besides, the parabolic kernel shape was shown to be optimal by (Silverman 1986). Finally parabolic kernel evaluations only require a few basic floating point operations, making them even more efficient.

4.3 Combination point t^* and number of tracks

The remaining parameters to be discussed are the number of realizations generated and the location, t^*, where these realizations are combined using a kernel estimation. The effect the number of realizations have on the forward-reverse estimator is well known, at least, in terms of orders of convergence. Given a d-dimensional problem, the overall estimator accuracy is of order $\mathcal{O}(N^{-1/2})$ for $d \leq 4$ and of order $\mathcal{O}(N^{-4/(4+d)})$, otherwise, with $N = M$ the number of forward and reverse realizations, respectively. Until now, the influence t^* has on the estimator has not been thoroughly studied, probably because its exact value is irrelevant for theoretical purposes, provided of course that the estimator retains its forward and reverse parts. In this section, we will show that in practice, this parameter is of utmost importance to the efficiency of the estimator.

The experimental results given below are all based on the application of the forward-reverse estimator on the BOD model. Although many experiments were done to validate results, we only selected a single set of parameters to illustrate our results. The parameter settings are summarized in Table 3. The parabolic kernel was used for kernel evaluations, with a bandwidth of $(N/\log N)^{-1/3}$, which is equivalent to a support of $\sqrt{7/3}(N/\log N)^{-1/3}$.

Parameter	**Value**
Start time (t)	$t = 0$
End time (T)	$T = 4$
Model parameters	$\sigma_B = 1.5$, $\sigma_O = 1.5$, $\sigma_N = 1.0$, see further Table 2
Initial point	$B_t = 15$, $O_t = 8.5$, $N_t = 5$
Default evaluation point	$B_T = 3$, $O_T = 9$, $N_T = 0.1$
Exact transition density	0.13477720...

Table 3: Default parameter settings for the BOD model.

In our first experiment, we determined the influence of t^* on the estimator. The continuous parameter space of t^* was first discretized, and only values of t^* between 0.01 and 3.99, inclusive, at a regular interval of 0.01 were considered. For each of those values, 100 independent density estimations were done, to determine the mean and variance of the estimator. To avoid errors due to the use of a numerical integration scheme, the exact solution was sampled to provide realizations of the forward and reverse processes. Not only does this reduce the noise in the estimator, thus showing the influence of t^* more clearly, it also greatly reduces the time requirement for the $39,900$ estimations that were made. The results obtained for these runs, with $N = M = 1,000$, are summarized in Figure 3.

For small t^*, the results show that many estimations are not very accurate, compared to the known answer of approximately 0.13. A closer look at the results for $t^* \in [2.7, 4)$, given by Figure 4, however, shows that the forward-reverse estimator is indeed capable of giving accurate results for a range of t^* values. Reassuring as this is, the question remains what exactly causes the highly inaccurate answers for small t^*? The most obvious answer is to say that N and M are insufficiently large, and

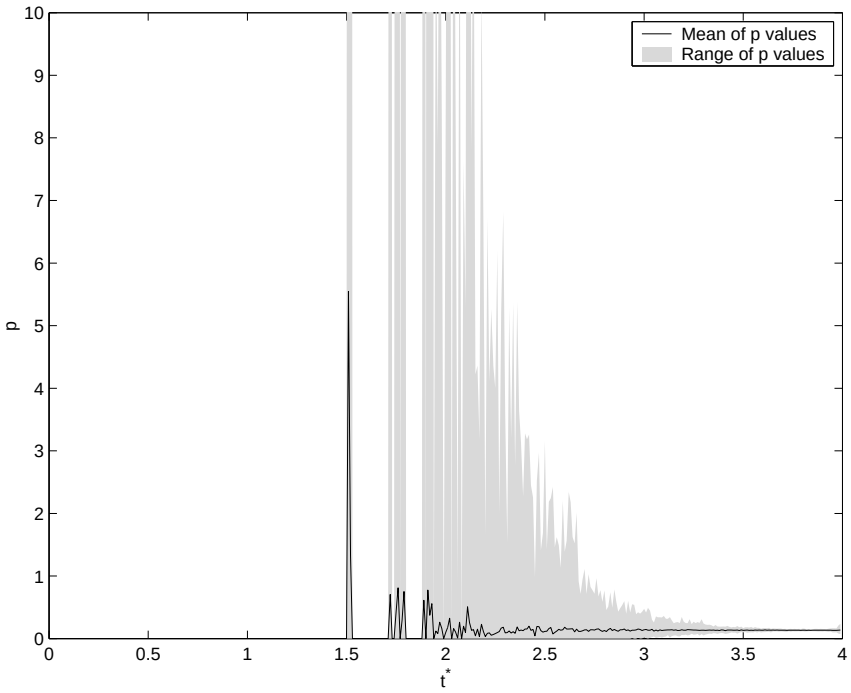


Figure 3: Range and mean of p values for different $t^* \in (0, 4)$, based on 100 forward-reverse estimations at each of the t^* locations, with N=M=1,000 and BOD parameters as given by Table 3.

indeed, increasing them will result in more accurate estimations. In fact, all other reasons given below can be reduced to the lack of realizations in either direction. Nevertheless, they give valuable insight in the application of the FRE and therefore deserve mentioning.

Given the limited number of realizations used, one reason behind the erratic outcome of the FRE is the kernel estimation. During each density estimation the number of reverse realizations that fell within the kernel support was counted. The support domain was taken to be the smallest box enclosing all forward realizations, enlarged at both sides in each dimension by the bandwidth of the kernel. This way, only those reverse realizations that are within the support can eventually attribute to the probability density. In Figure 5 the average and range of these counts are plotted to t^*. For values of t^* less than 2.5, the number of reverse realizations is zero in all but a few cases, causing the summation of all kernel estimations to evaluate to zero as well. In those cases where some kernel estimations were done, the density estimation is often exceedingly large. This is caused by a combination of factors. First, the process is instable (see Figure 14 for the covariance trace of the reverse process) which leads to an extremely flat Gaussian distribution. Combined with the small sample size, a realization that does end up inside the support domain can do so virtually anywhere. In addition, the already inaccurate estimation will then be inflated by the scaling factor $\mathcal{Y}$, which grows exponential in reverse time for the BOD

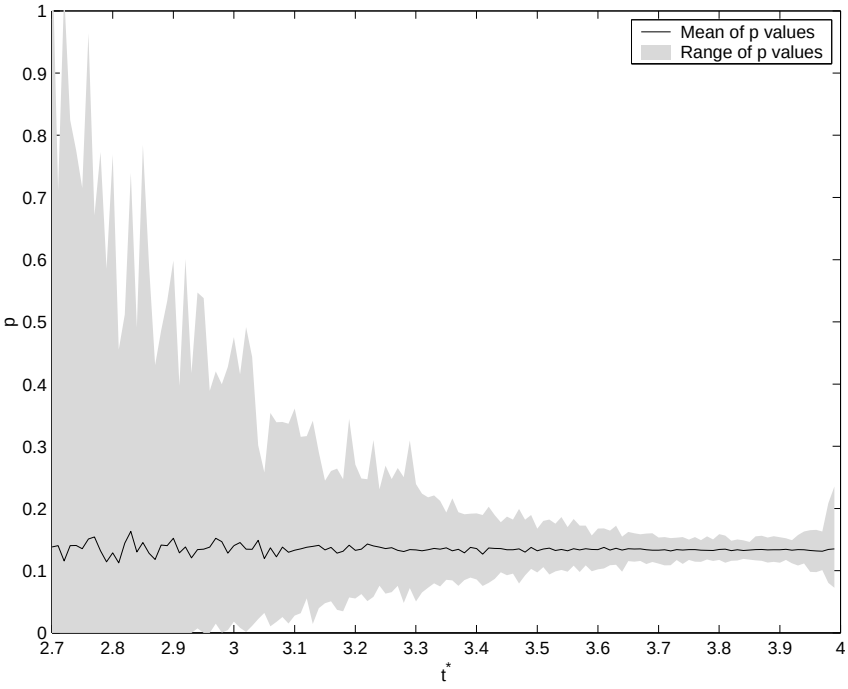


Figure 4: Range and mean of p values for different $t^* \in [2.7, 4)$, based on 100 forward-reverse estimations at each of the t^* locations, with N=M=1,000 and BOD parameters as given by Table 3.

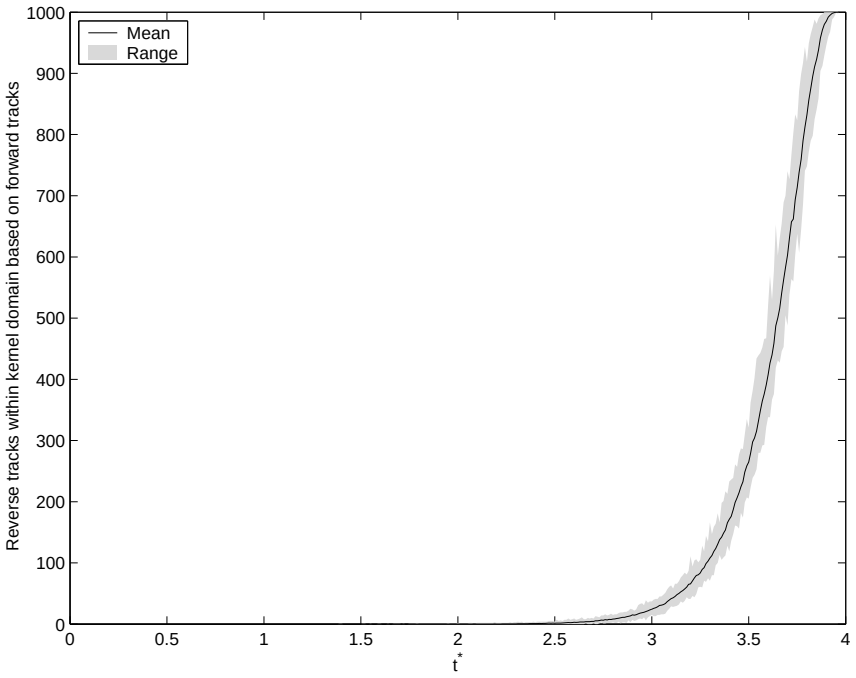


Figure 5: Range and mean of number of reverse realizations that fall within the support of the kernel estimation at different $t^* \in (0, 4)$. Based on 100 forward-reverse estimations with N=M=1,000 and BOD parameters as given by Table 3.

model, resulting in the worst case in massive overestimation of the probability density. It is important to note that this phenomenon also occurs when the forward part of the estimator, including the kernel estimation, is replaced by a function evaluating the exact density, as shown in Figure 6.

For the above problem, two solutions exist; (a) use more realizations to improve accuracy, and (b) avoid values of t^* where the problem occurs. Both solutions will be considered below as we work towards a practical approach.

The influence of the number of realizations on the estimation is perhaps best studied by looking at the variance in the estimation observed for different values of N and M. Because of the results found earlier, we will henceforth only consider $t^* \in [2.8, 4)$.

To start with, the ratio between variance obtained using $N = 1,000$ and $N = 10,000$, while keeping M fixed at $1,000$ was determined and is shown in Figure 7. For $t^* < 3.5$ there isn't any notable improvement with variance ratio's around 1. It is only for values of t^* near T that considerable improvements are made. These results compare well with those shown in Figure 6, where the variance of the estimators based on kernel estimations and exact density function are roughly proportional to each other for $t^* < 3.5$, and diverge as t^* is moved closer to T, leaving room for improvement. This improvement cannot have been caused by the larger distance over which the forward realization had to be integrated, since we directly sampled the exact distribution. Nor is it likely to be because of the increasingly diffusion

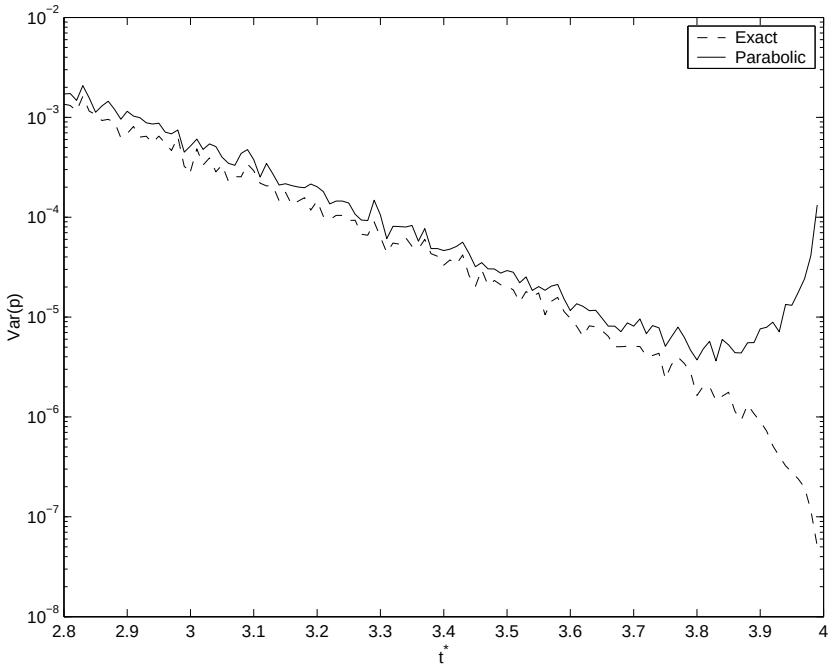


Figure 6: Variance in p values for different $t^* \in [2.8, 4)$, based on 100 forward-reverse estimations at each of the t^* locations, with N=M=10,000 and BOD parameters as given by Table 3, using parabolic kernel estimation (solid line) or the analytical transition density function (dashed line).

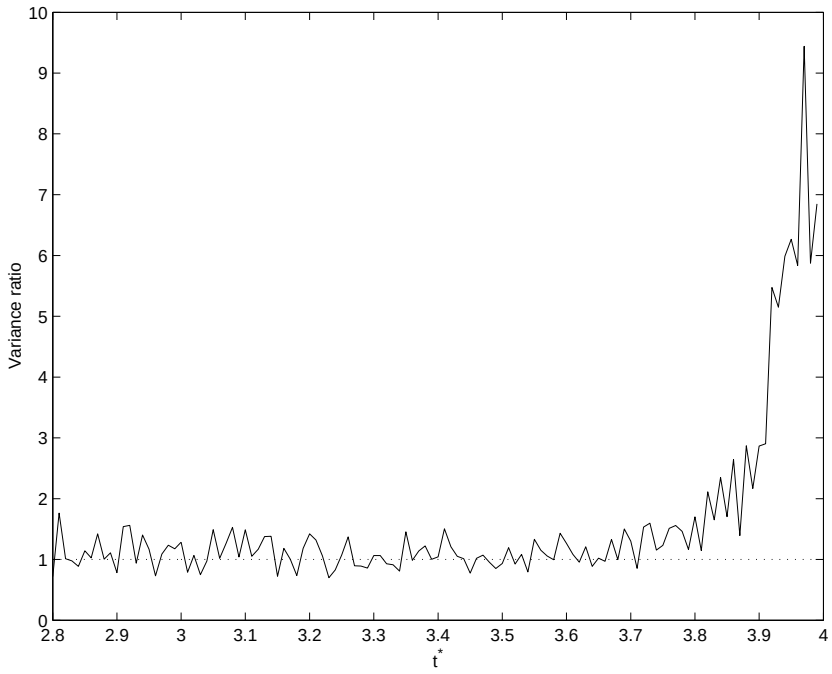


Figure 7: The effect of increasing the number of forward realizations N from 1,000 to 10,000, while retaining M = 1,000. Measured in the ratio of variance in p variables for different $t^* \in (2.8, 4]$, before and after the change. Based on 100 forward-reverse estimations with BOD parameters as given by Table 3.

of the forward process, because most of that occurs at the beginning of the time spectrum. The real reason for the lack of improvement is the fast rate at which the reverse process diffuses, and the reduction of non-zero kernel evaluations that follows. Therefore, improving the kernel estimation accuracy will only result in marginal improvements.

In the reverse experiment, we increase M from $1,000$ to $10,000$ while maintaining $N = 1,000$ and again determine the variance improvement. The results are shown in Figure 8. This time, there is no improvement in the estimation for t^* near T, with a ratio of 1. Better results are obtained towards t where the ratio increases to the maximum possible value of 10. In between, the kernel estimation's accuracy and reverse Monte Carlo accuracy are clearly not balanced, or more precisely, their balance was disturbed by the increase of reverse realizations. In case both M and N are simultaneously increased to $10,000$, the results are as given by Figure 9. Here, the results are good overall with variance improvement ratio's of around 10.

Increasing the number of realizations adds to the computation time of the estimator. Therefore, we would like to know the best combination of N, M, and t^* to reach a certain level of accuracy in the estimation. Going back to the results obtained with $N = M = 1,000$, we will explain how this can be done. In Figure 10 the variance in 100 estimations is plotted to t^* along with a dotted line indicating the minimum level of variance. Given the number of realizations, this is clearly the best result we

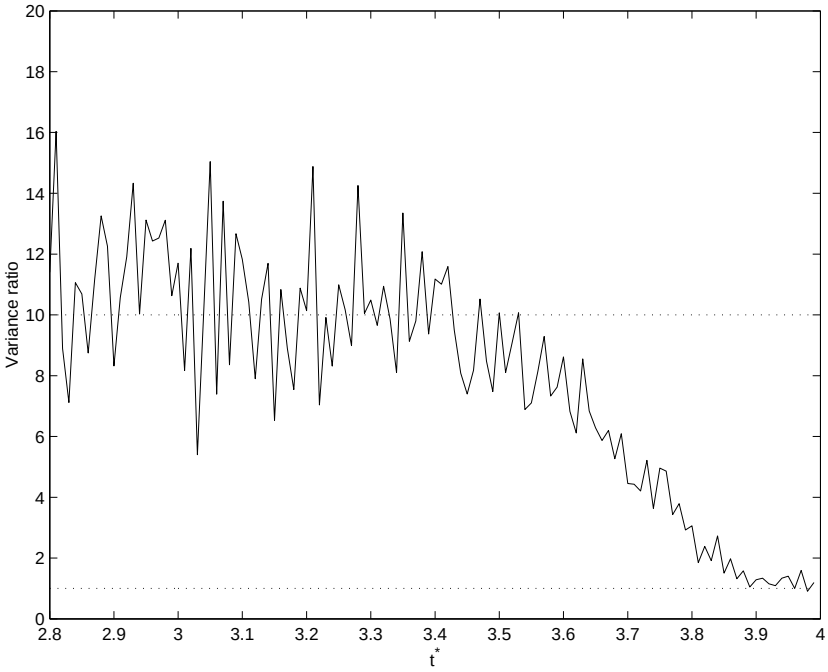


Figure 8: The effect of increasing the number of reverse realizations M from 1,000 to 10,000, while retaining N = 1,000. Measured in the ratio of variance in p variables for different $t^* \in (2.8, 4]$, before and after the change. Based on 100 forward-reverse estimations with BOD parameters as given by Table 3.

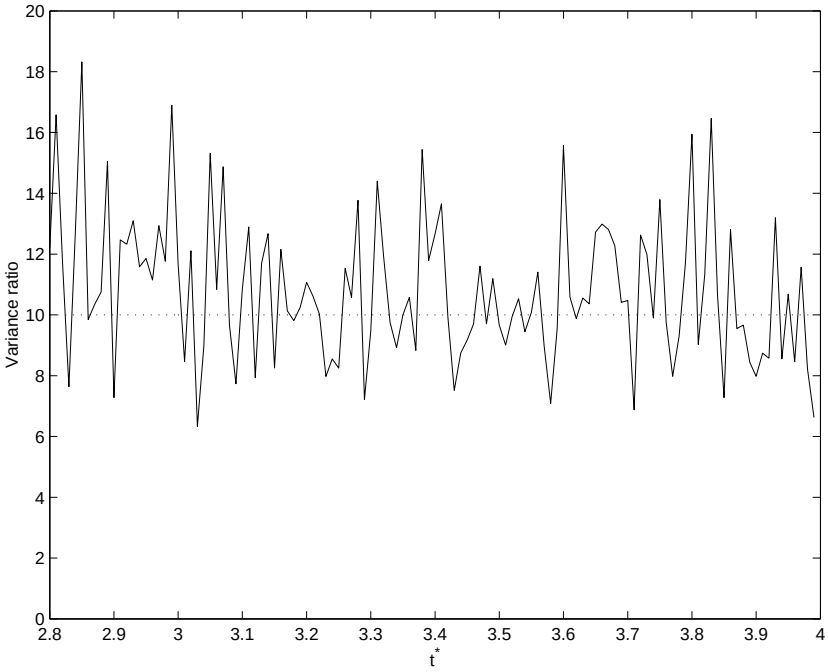


Figure 9: The effect of simultaneously increasing the number of forward and reverse realizations from 1,000 to 10,000. Measured in the ratio of variance in p variables for different $t^* \in (2.8, 4]$, before and after the change. Based on 100 forward-reverse estimations with BOD parameters as given by Table 3.

can obtain, but at what expense? Because of the sampling, we could only measure the kernel estimation times for each $t*$ is given by Figure 11. It can be seen that the kernel time at the assumed optimal t^* is high, compared to those for lower valued combination points. This raises the question whether the given t^* is indeed optimal, or if lower variations could be reached using a different t^* and more realizations, at the same computational time.

Suppose that it is possible to decrease variance linear with the number of forward and reverse realizations (we assume $N = M$ from now, unless specified otherwise) infinitely, and that such increase will affect the computation time linearly. The fact that this is practically impossible, with the kernel evaluation order and the need to decrease the numerical integration stepsize, will only strengthen our reasoning. Then, based on the minimum variance seen in Figure 10, it is possible to determine the factor by which the number of realizations be multiplied to achieve the same level of variance, as shown in Figure 12. By multiplying the kernel times with this factor, we obtain optimistic computation time for kernel evaluation to get iso-variance estimations. The location where this time is minimum, say t^*_{Low}, is shifted towards t slightly from the location of the original minimum variance, t^*_{High}. Under the assumption that evaluation of the numerical integration scheme for the forward and reverse is equally expensive in terms of computational cost, the real optimum, t^*_{Opt}, is limited as follows:

$$t^*_{Low} < t^*_{Opt} \leq t^*_{High}. \tag{19}$$

The lower bound will shift further towards t^*_{High} when numerical integration is included and the kernel evaluation times are determined more realistically. In case the reverse process is much more expensive per timestep than the forward process, the lower bound may shift a little towards t.

Normally, it is not practically feasible to use the above approach to determine a near-optimal t^*, because of the computational effort required, at least, not in a similar level of detail. As suggested earlier the process can be done coarsely but that will result in an equivalently coarse selection of t^*. It would be better if, somehow, we could determine this location directly from the two processes. For the BOD process, and other Gaussian processes in general, it was noted that the difference in variance (for higher dimensional systems, the covariance matrix' trace) between the forward and reverse processes was reflected in the variance in the estimation, as can be seen by comparing Figures 14 and 10. This would lead to a simple and elegant solution to the problem of finding a good value for t^*. Simply generate a small set of realizations and repeatedly advance them by use of the numerical integration scheme. While doing so, maintain the sum of the variance of each component of the realizations, for each timestep. Next, find t^* where the absolute difference between the obtained forward and reverse variances is at its minimum and use this location as the combination point for the full-run. The results from experiments based on a simple diffusion algorithm (see Milstein, Schoenmakers, and Spokoiny 2002 for further information) where the analytical variance is known, were very promising as the values found for t^* proved to be good indications of the optimal location.

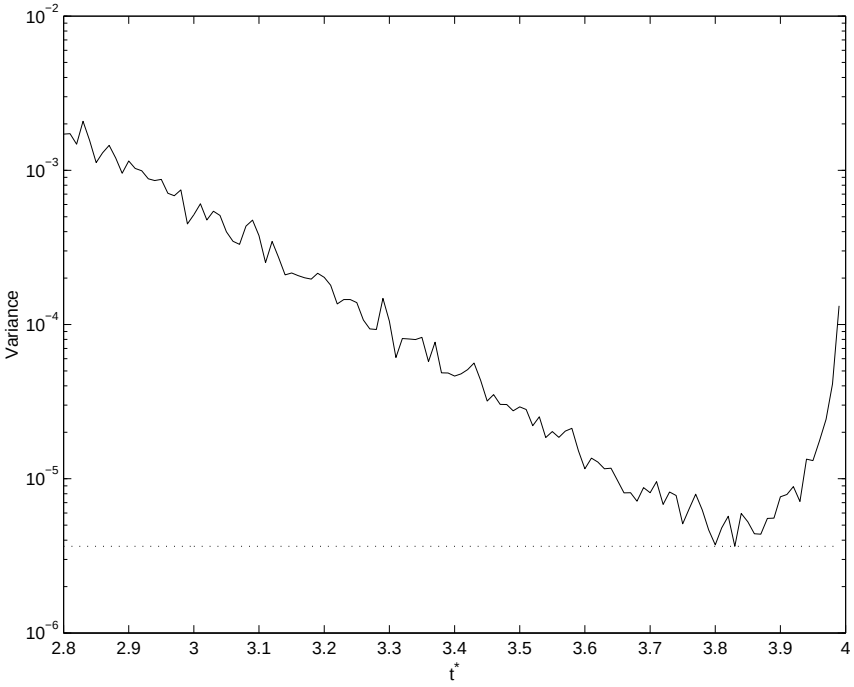


Figure 10: Variance in p values for different $t^* \in [2.8, 4)$, based on 100 forward-reverse estimations at each of the t^* locations, with N=M=10,000, parabolic kernel estimation, and BOD parameters as given by Table 3

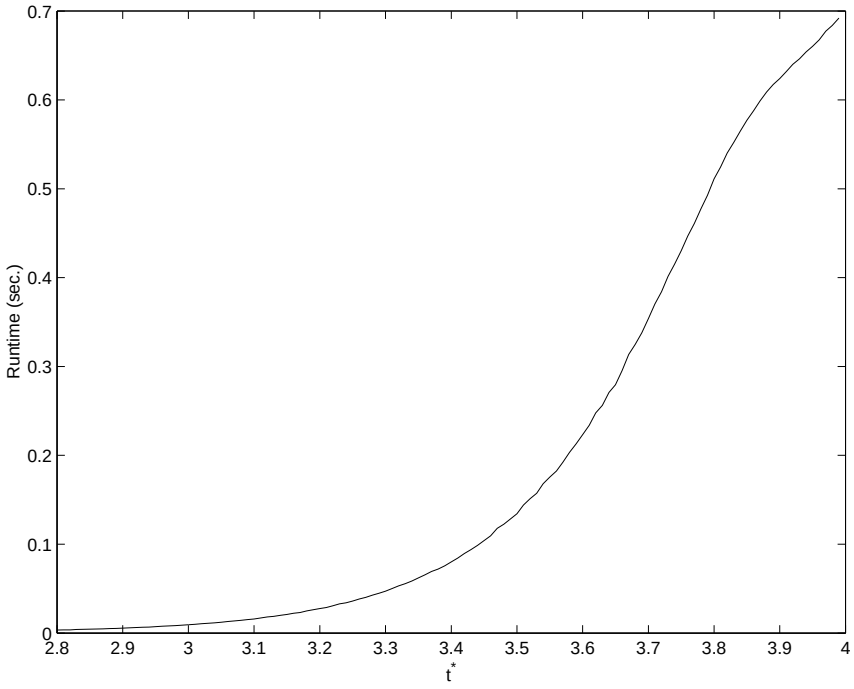


Figure 11: Average kernel estimation times recorded for different $t^* \in [2.8, 4)$. Based on 100 forward-reverse estimations with N=M=10,000 and BOD parameters as given by Table 3

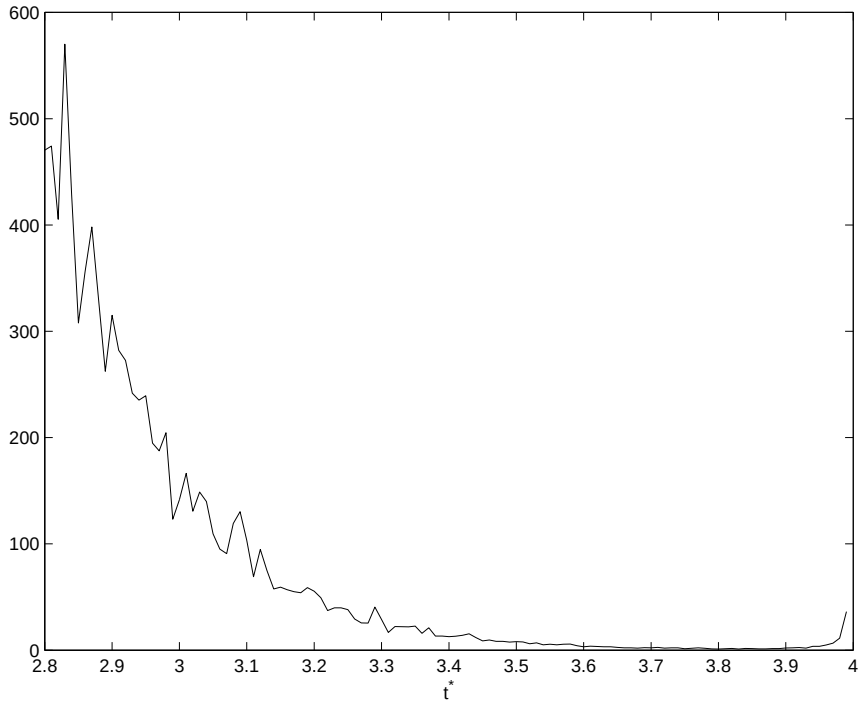


Figure 12: Factor by which number of forward and reverse tracks should at least be multiplied in order to obtain a density estimation with a variance equal to the one indicated by the dotted line in Figure 10.

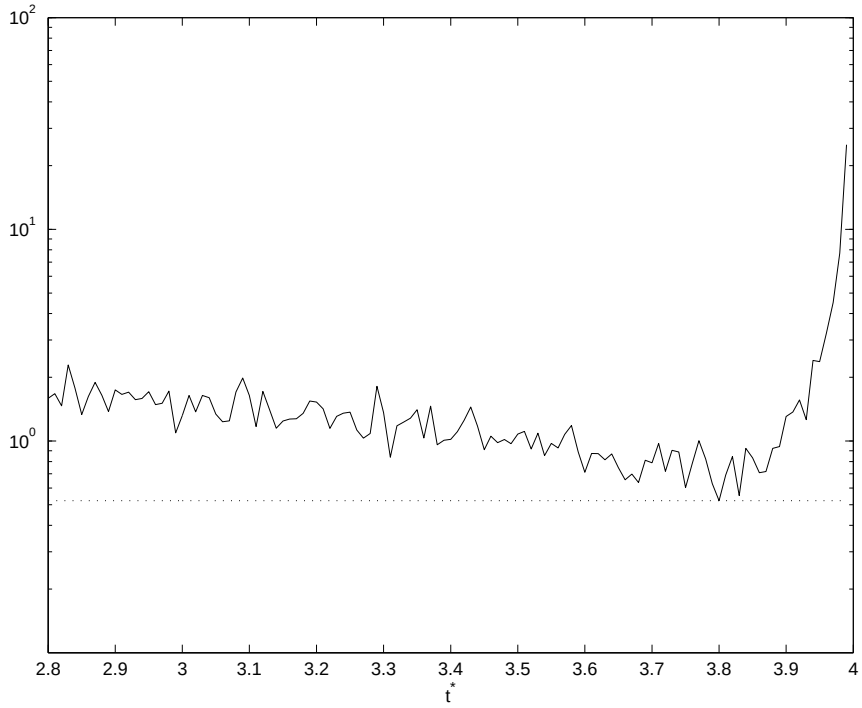


Figure 13: Idealistic time required to complete kernel estimation in case the number of realizations is chosen such that the variance in density estimation is the same for all $t^* \in [2.84)$.

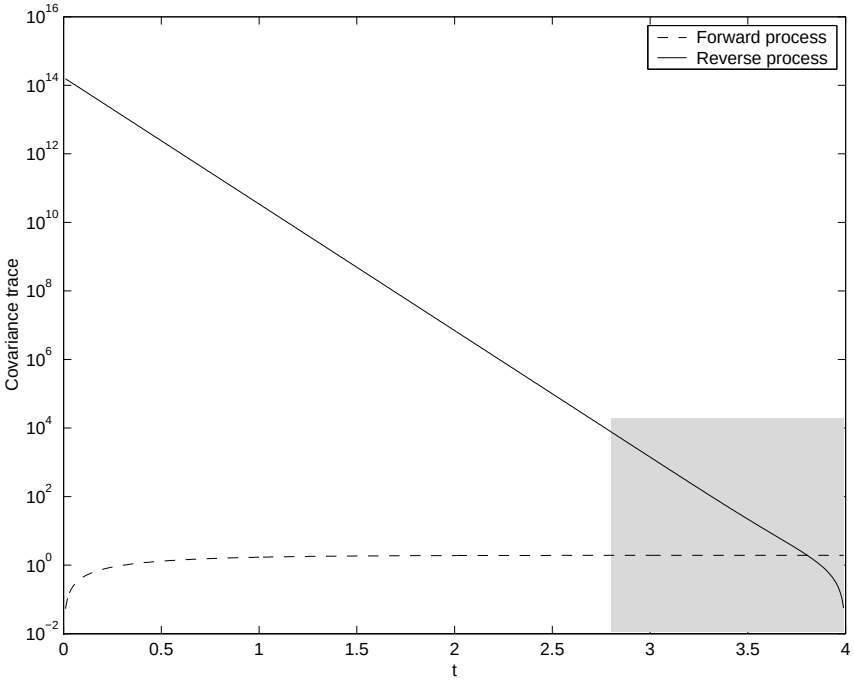


Figure 14: Trace of the covariance matrices of the forward and reverse processes, plotted to time. Time $t = 4$ is the end time of the simulation and hence the start time of the reverse process.

5 Performance evaluation of FRE versus FE

In this section, the FRE and FE are compared to each other both in terms of order of convergence of the error in the estimation, and in terms of accuracy to computing time.

The order of convergence of the accuracy of the estimated probability density can be approximated by running a number of estimations for each of a set of values for N. The standard deviation and bias of the estimation are then determined for each N, and the results plotted on a log-log scaled figure. From this figure, the order of convergence corresponds directly to the slope of the plotted line and can hence be easily determined.

We did a large number of experiments with different choices for t^* and location of the evaluation point y, in $p(t, x, T, y)$. Because of the similarity of the results, we only show the results for the BOD model with parameters as given by Table 3. For the FRE, we used $N = M = 2^i$, with $i = 5, \ldots, 15$, and $t^* = 3.803800$, while for the FE, $n = 2^j$, with $j = 5, \ldots, 18$ was used. In both estimators, the transition density was estimated $10,000$ times to get accurate values for the standard deviation and bias. In addition, the numerical integration was replaced by sampling the known distributions. Figures 15 and 16 respectively show the standard deviation and squared bias in the forward and forward-reverse estimations, along with dotted lines indicating the theoretical orders of convergence, plotted to the number of realizations.

The results match the theoretical results very closely, with the exception of the squared bias for the forward-reverse estimator which shows a convergence of $\mathcal{O}(N^{-1.1})$, more than the $\mathcal{O}(N^{-1})$ plotted. It is known (Milstein, Schoenmakers, and Spokoiny 2002) that the squared bias for problems with $d < 4$, is $o(N^{-1})$. Therefore, the results still match the theory, although the convergence rate found may be overestimated because of the relatively small number of realizations used.

Needless to say, the values of t^* and y, did not affect the order of convergence of the standard deviation and bias, although, they did influence the absolute accuracy of the estimation. The choice of numerical scheme used to generate realizations of the forward and reverse processes with, cannot improve the order of convergence. It can however prevent improvement in the accuracy, regardless of the number of realizations used, when the stepsize is taken too large.

For a realistic comparison between the two estimators, the numerical integration scheme has to be taken into account since it accounts for a major part of the total computational cost. For example, the run-time of the forward estimation can be expressed as follows

$$\text{runtime}_{FE} = \tau_{\text{integration step}} \frac{N \cdot (T - t)}{\Delta} + \tau_{\text{kernel eval}} \cdot N \tag{20}$$

where the constants $\tau_{\text{integration step}}$ and $\tau_{\text{kernel eval}}$ represent the execution time of a numerical integration step and a kernel evaluation respectively, and Δ is the timestep used in the integration scheme. The constants will differ slightly for each implementation and computer platform, but are easily measured. The value for $\tau_{\text{integration step}}$ was found to be several times larger than $\tau_{\text{kernel eval}}$, mostly because of the cost of

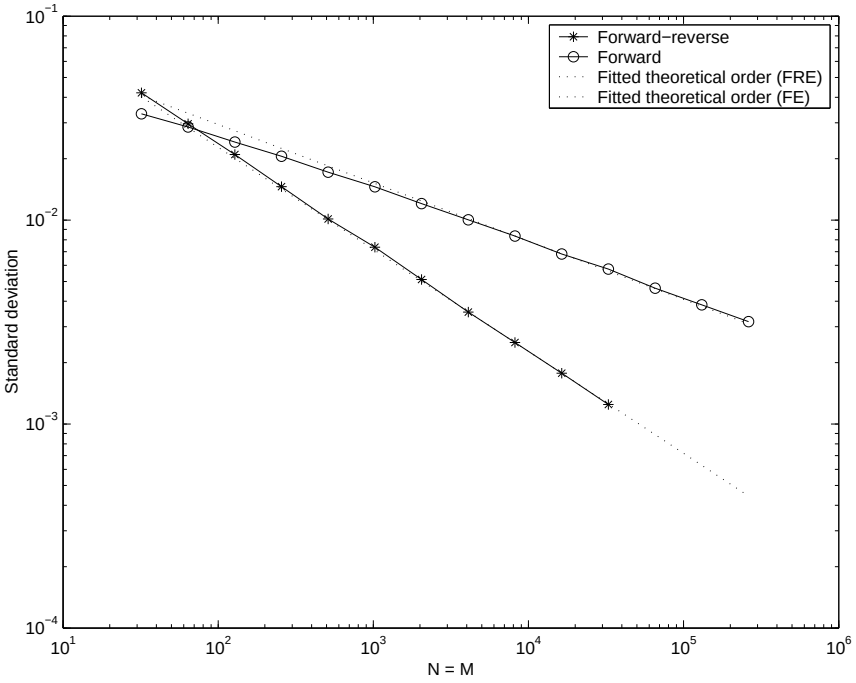


Figure 15: Convergence of the standard deviation in the estimated transition density.

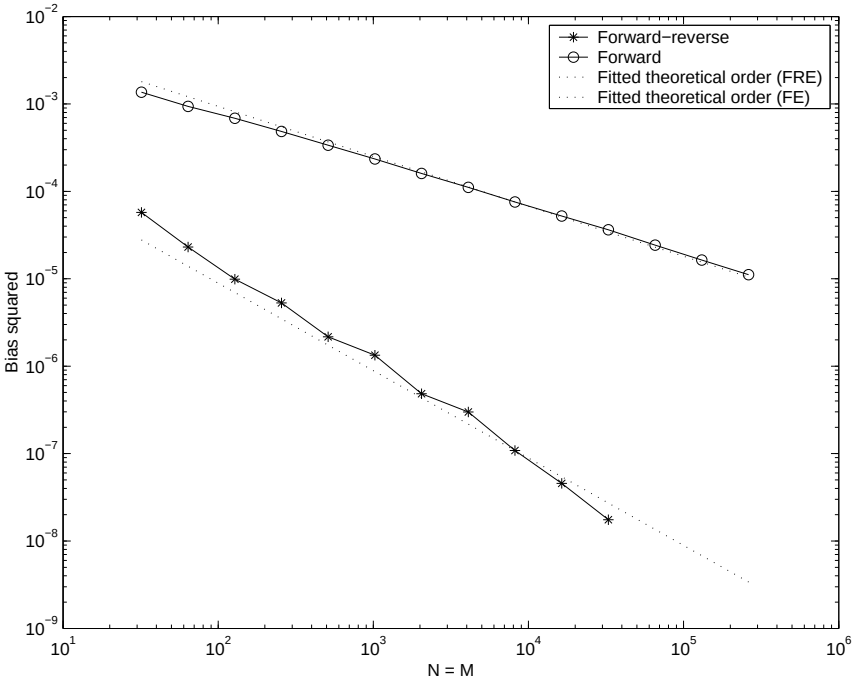


Figure 16: Convergence of squared bias in the estimated transition density.

Stepsize	0.1	0.01	0.001
N	574,950	293,389	266,052

Table 4: Number of realizations that can be used in the FE to obtain a run time similar to the FRE, when using N = M = 262,144, and a stepsize as indicated.

generating random numbers. As a result, even at a moderate timespan $(T-t)$ and large Δ, most of the computation time will be spent on generating realizations.

The run-time for the forward-reverse estimator is more difficult to express because it is not linear to the number of realizations used. Especially the kernel estimation time is hard to predict because it depends on the forward and reverse distributions at the combination point, which in turn depend on the x and y in the density function. Despite this, we still want to compare the FE and FRE in terms of computation time. This can be done by recording the simulation time of a forward-reverse estimator. This time can then be substituted in (20) to determine the matching number of realizations in the FE simulation. After running an equivalent FE simulation, the results of the two estimators be compared.

The above approach was applied to the BOD model with parameters as given by Table 3. For numerical integration, the Euler scheme as given by (2) was used with stepsize, Δ, of 0.1, 0.01 and 0.001. The number of realizations used in the FRE was $N = M = 2^i$, with $i = 7, \ldots, 18$. The value of t^* was related to the stepsize, resulting in value of 3.8, 3.80, and 3.804, respectively. As an example, Table 4 gives the equivalent number of realizations to be used in the FE, in case $i = 18$, or similarly, $N = M = 262,144$; The results of the experiments are given by Figures 17 and 18. Curves with smaller timesteps will have larger associated run-times and will hence start more to the right in the plot. Both the standard deviation and squared bias of the FRE are consistently smaller than those of the FE, while obtained at a similar computational effort. Therefore we can conclude that the FRE is indeed superior to FE, even for small numbers of realizations.

6 Conclusions

We have applied the recently introduced forward-reverse model (Milstein *et al.* 2002) to a stochastic environmental BOD model. After the derivation of the forward reverse representation of the model, we investigated a number of issues that required special attention in order to make application of the forward-reverse method efficient in a practical sense. We argued that the calculation of the kernel estimator must be implemented with care and presented an $\mathcal{O}(M \log N)$ scheme. In Section 4.3, the relation between combination point t^* and the computational efficiency has been analyzed. We presented an efficient practical approach for determining the appropriate value for t^*. It was then shown that the forward-reverse estimator indeed achieves an $\mathcal{O}(N^{-1/2})$ accuracy when applied to the BOD model, compared to an $\mathcal{O}(N^{-2/7})$ accuracy for the forward estimator. The comparison of the estimation error to the run-time clearly shows the advantage of using the forward-reverse estimator. Espe-

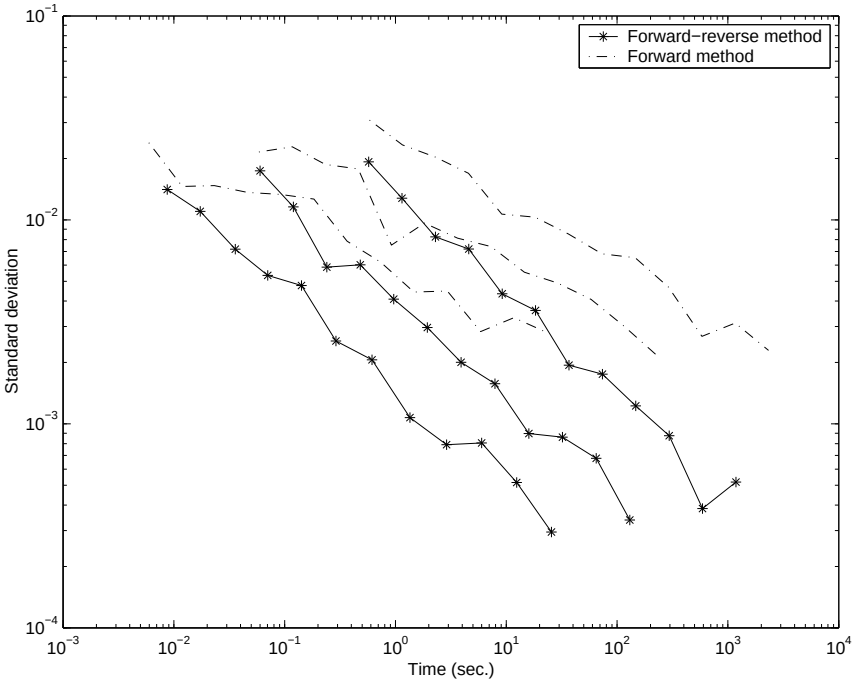


Figure 17: Standard deviation in transition density estimation for the forward and forward-reverse estimators, plotted to simulation time.

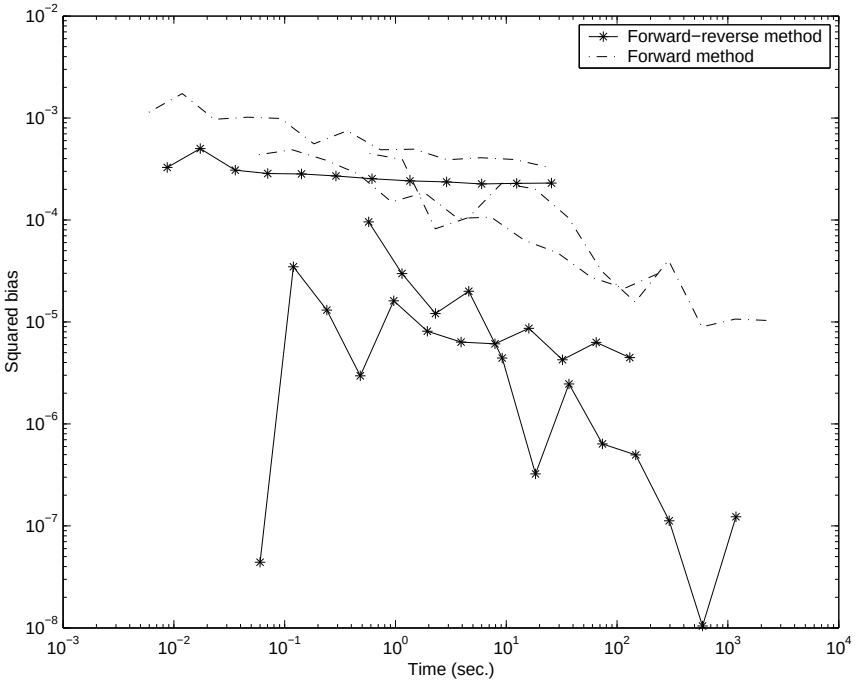


Figure 18: Squared bias in transition density estimation for the forward and forward-reverse estimators, plotted to simulation time.

cially for problems where a high accuracy is required, the FRE will be substantially faster than the FE, as seen in Figure 17. These results are encouraging and show the practical use of the forward-reverse approach for density estimation in environmental risk analyses.

References

Bally, V. and D. Talay (1996a). The law of the Euler scheme for stochastic differential equations I: convergence rate of the density. *Monte Carlo Methods Appl.* **2**, 93–128.

Bally, V. and D. Talay (1996b). The law of the Euler scheme for stochastic differential equations I: convergence rate of the distribution function. *Probab. Theory Related Fields* **104**, 43–60.

Devroye, L. and L. Györfi (1985). *Nonparametric Density Estimation: The L_1 View*. Wiley.

Greengard, L. and J. Strain (1991). The fast Gauss transform. *SIAM J. Sci. Stat. Comp.* **12**, 79–94.

Hammersley, J.M. and D.C. Handscomb (1964). *Monte Carlo Methods*. London: John Wiley & Sons, Inc.

Jazwinsky, A.W. (1970). *Stochastic processes and filtering theory*. New York: Academic Press.

Kloeden, P.E. and E. Platen (1992). *Numerical Solution of Stochastic Differential Equations*. New York: Springer Verlag.

Loucks, D.P., J.R. Stedinger, and D.A. Haith (1981). *Water Resources Systems Planning and Analysis*. New York: Prentice-Hall.

Milstein, G.N. (1995). *Numerical Integration of Stochastic Differential Equations*. Dordrecht: Kluwer Academic Publishers.

Milstein, G.N. and J.G.M. Schoenmakers (2002). Monte Carlo construction of hedging strategies against multi-asset European claims. *Stochastics and Stochastics Reports* **1-2**, 125–157.

Milstein, G.N., J.G.M. Schoenmakers, and V. Spokoiny (2002). Transition density estimation for stochastic differential equations via forward reverse representations. *Bernoulli*. (submitted).

Schoenmakers, J.G.M. and A.W. Heemink (1997). Fast valuation of financial derivatives. *Journal of Computational Finance 1*(1), 47–67.

Schoenmakers, J.G.M., A.W. Heemink, K. Ponnambalam, and P.E. Kloeden (2002). Variance reduction for Monte Carlo simulation of stochastic environmental models. *Applied Mathematical Modelling* **26**, 785–795.

Silverman, B.W. (1986). *Density Estimation for Statistics and Data Analysis*. 29 West 35th Street, New York NY 10001: Chapman and Hall Ltd.

Stijnen, J.W. (2002). *Numerical Methods for Stochastic Environmental Models*. Ph.d. thesis, Delft University of Technology.

Thomson, D.J. (1987). Criteria for the selection of stochastic models of particle trajectories in turbulent flows. *J. Fluid Mech.* **180**, 529–556.