Title
Parameter Inference for Stochastic Differential Equations

Author
Rajapaksha Wasala Mudiyanselage, Anusha Madushani

Publication Date
2017

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, MERCED

Parameter Inference for Stochastic Differential Equations

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Applied Mathematics

by

R. W. M. A. Madushani

Committee in charge:

Professor Harish S. Bhat, Chair
Professor Roummel F. Marcia
Professor Suzanne S. Sindi

2017

The dissertation of R. W. M. A. Madushani is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

(Professor Roummel F. Marcia)

(Professor Suzanne S. Sindi)

(Professor Harish S. Bhat, Chair)

University of California, Merced

2017

DEDICATION

To my wonderful mother, Asoka Damayanthi, and my beloved husband Lasith Adhikari.

EPIGRAPH

In the middle of every difficulty
lies opportunity.
—Albert Einstein

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor professor Harish Bhat, for his continuous support for my research, for his encouragement, patience, and immense knowledge. Without his help and guidance, this thesis would not have been possible. I would also like to express my sincere gratitude to my thesis committee members: professor Roummel Marcia and professor Suzanne Sindi for their insightful comments and suggestions.

Also I would like to thank professor Arnold Kim for helping me to solve some research questions providing great ideas and directing me in the correct path. I also want to thank professor Francois Blanchette for being willing to discuss some of my research questions when I was stuck and disappointed. At UC Merced, I learned a lot of applied mathematics concepts and techniques which helped me to complete this thesis successfully. For this, I am grateful to the professors who taught me here: Noemi Petra, Boaz Ilan, Arnold Kim, Roummel Marcia and Francois Blanchette.

I also would like to thank Dr. Nitesh Kumar, Dr. Derya ahin Biryol, Dr. Jane Hyo Jin Lee, and Dr. Garnet Vaz for helping me in the beginning of my graduate life giving me bits of advice to succeed. Last but not least, I would like to thank my beloved husband Dr. Lasith Adhikari who has always encouraged me and helped me to overcome the difficulties throughout this experience, and I can not thank you enough for the things you have done for me throughout my life ever since I met you.

VITA

2010	B. Sc. (Special) in Mathematics, University of Sri Jayewardenepura, Sri Lanka
2012-2017	Graduate Teaching Assistant, University of California, Merced
2017	Ph. D. in Applied Mathematics, University of California, Merced, USA

PUBLICATIONS

H. S. Bhat and R. W. M. A. Madushani, "Computing the density function for a nonlinear stochastic delay system", *Proceedings of the 12th IFAC Workshop on Time Delay Systems (TDS 2015), IFAC-PapersOnLine*, 48 (12), pp. 316-321, 2015.

H. S. Bhat, and R. W. M. A. Madushani, "Nonparametric adjoint-based inference for stochastic differential equations", *Proceedings of the 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 798-807, 2016.

H. S. Bhat, R. W. M. A. Madushani, and S. Rawat, "Scalable SDE filtering and inference with Apache Spark", *Journal of Machine Learning Research W&CP, 53, Proceedings of the 5th International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications (KDD BigMine '16)*, pp. 18-34, 2016.

H. S. Bhat, R. W. M. A. Madushani, and S. Rawat, "Bayesian inference of stochastic pursuit models from basketball tracking data", *Proceedings of the 3rd Bayesian Young Statisticians Meeting (BAYSM 2016)*, Accepted 2016.

H. S. Bhat and R. W. M. A. Madushani, "Density Tracking for Stochastic Differential Equations", *Statistics and Computing*, Submitted 2016.

H. S. Bhat, R. W. M. A. Madushani, and S. Rawat, 'Parameter inference for stochastic differential equations with density tracking by quadrature", *8th international workshop on simulation (IWS)*, Submitted 2016.

ABSTRACT OF THE DISSERTATION

Parameter Inference for Stochastic Differential Equations

by

R. W. M. A. Madushani
Doctor of Philosophy in Applied Mathematics
University of California Merced, 2017
Professor Harish S. Bhat, Chair

In this dissertation, we consider the problem of inferring unknown parameters of stochastic differential equations (SDE) from time-series observations. In particular, we develop and test numerical methods to perform frequentist and Bayesian inference for SDE. A key challenge in developing practical inference algorithm is the computation of the likelihood. To compute the likelihood, we propose a novel, fast method that tracks the probability density of the SDE. Our method does not rely on sampling; instead, it evolves the density in time using repeated quadrature on the Chapman-Kolmogorov equation of the Markov chain that results from a time discretization of the SDE. We name our method density tracking by quadrature (DTQ). Our method enables accurate, parallelizable computation of the likelihood when the data is collected with large inter-observation time or when the data consists of one or more time series. In this dissertation, we focus on a particular case of the DTQ method that arises from applying the Euler-Maruyama method in time and the trapezoidal quadrature rule in space. Under some regularity condition for the drift and the diffusion terms of SDE, we theoretically prove that the density computed by the DTQ method converges in L^1 to the exact density with a first-order convergence rate in temporal step size. Numerical tests show that the empirical performance of the DTQ method complies with the theoretical convergence results.

To perform inference using maximum likelihood approach, we develop methods to compute the gradient of the likelihood. We propose a direct method to compute the gradient from the DTQ likelihood and use this direct method to perform parametric

inference of the SDE. We also propose a more efficient adjoint-based method to compute the gradient information with a computational cost (in time) that does not scale with the dimension of the unknown parameter vector. Therefore, we use this adjoint-based method to perform nonparametric inference of SDE. Using the DTQ method to compute the likelihood, we also develop a Markov Chain Monte Carlo (MCMC) algorithm using a Metropolis scheme to perform Bayesian inference. We apply this Bayesian inference method for coupled SDE. In this work, we derive a coupled, nonlinear SDE to model the chasers pursuit of the runner in a basketball game. We perform Bayesian inference using NBA tracking data to show the appropriateness of the model for basketball fast break situations.

Chapter 1

Introduction

Stochastic differential equations (SDE) are commonly used as a tool to model dynamical systems that are influenced by random noise. For example, in finance, many different SDE have been developed to model financial quantities such as asset prices, interest rates, and their derivatives. To name a few, the geometric Brownian motion is used to models stock prices [47], and models of interest rates include Vasicek model [63] and the Cox-Ingersoll-Ross model [21, 20]. In population biology, SDE are used to model two interacting populations such as epidemics consisting of susceptible and infected subpopulations [28] or predator-prey systems [2]. Physicists use SDE to model the motion of particles subjected to thermal fluctuations [57]. Other applications areas of SDE include genetics [33], Social sciences [19], geophysics [42], biology[6], and geostatistics [22].

In this dissertation, we focus on the problem of estimating unknown parameters from time-series observations of an SDE. Consider the following SDE of the form

$$dX_t = f(X_t;\theta)dt + g(X_t;\theta)dW_t \tag{1.1}$$

where X_t is a scalar stochastic process, $\theta \in \mathbb{R}^N$ is a vector of unknown parameters, and W_t is the standard Brownian motion also known as the Wiener process. Here f and g are referred to, respectively, as the drift and diffusion functions. Our goal is to estimate unknown parameter vector θ from time-series observations of X_t. When the functional form of the f and g are known, the problem is called a parametric inference problem. Otherwise, it is called a nonparametric inference problem. However, the parameter in-

ference of SDE models is a very challenging problem, due to the fact that the likelihood function is generally unknown for the case where time-discrete observations are available [58, 31, 26]. The exact likelihood function for the SDE can only be computed in special cases when we can solve analytically for the SDE's transition density. Therefore, prior work has focused on approximating the exact likelihood, either through analytical methods, numerical methods, or a combination of the two. For a thorough review of past work on this problem, we refer the reader to [18, 58, 31, 8, 26]. In our work, we seek for efficient numerical methods applicable to a large class of SDE that do not rely on sampling to solve the parameter inference problem.

1.1 Key Contributions of the Dissertation

The primary goal of this research work is to develop fast, scalable and deterministic algorithms to perform efficient Bayesian or frequentist inference for SDE. Following we list our main original contributions to the area of parameter inference of SDE.

1. We propose a novel fast numerical method to compute the likelihood function of a scalar SDE. The proposed method tracks the probability density function of the SDE and hence can be used to approximate transitional densities. By virtue of how the method is derived, we refer to the method as density tracking by quadrature (DTQ). The DTQ method stems from discretizing the SDE in time using the explicit Euler-Maruyama scheme, resulting in a discrete-time Markov chain on a continuous state space. By applying quadrature in the form of the trapezoidal rule, the DTQ method solves the Chapman-Kolmogorov equation for this Markov chain at each time step. When the data consists of one or more time series, the DTQ method enables accurate, parallelizable computation of the likelihood.

2. We establish theoretical and empirical convergence results of the proposed DTQ method. Our main result establishes that the density computed by the DTQ method converges in L^1 to both the exact density of the Markov chain (with exponential convergence rate), and to the exact density of the SDE (with first-order convergence rate). We also establish a Chernoff bound that implies convergence of a

domain-truncated version of the DTQ method. The convergence requires regularity of both f and g, together with restrictions on the rates at which the temporal and spatial grid spacing tends to zero. We carry out numerical tests to show that the empirical performance of the DTQ method matches theoretical results, and also to demonstrate that the DTQ method can compute densities several times faster than a Fokker-Planck solver, for the same level of error.

3. We develop efficient algorithms to compute maximum likelihood estimates for scalar SDE through numerical optimization of the negative log likelihood computed from the DTQ method. More specifically, we develop numerical methods to compute accurate approximations to the gradient of the negative log likelihood. We first propose a direct method of gradient computation and use the method to solve the parametric inference problem. Then we proposed a more efficient adjoint-based method to compute the gradient which enables the inference to scale well as the dimensionality of the unknown parameter vector grows. We apply the adjoint-based inference method to perform nonparametric inference of the drift and diffusion functions of an SDE.

4. We develop a Metropolis algorithm to perform Bayesian inference for models given by coupled SDE. To compute the required likelihood in the Metropolis algorithm, we generalize the DTQ method for coupled SDE. We test the effectiveness of the method through simulation experiments of a stochastic van der Pol oscillator model. We also formulate a stochastic version of the classical pursuit model which consists of a set of coupled, nonlinear SDE to model the chasers pursuit of the runner in a basketball game. Then we perform Bayesian inference for this stochastic pursuit model using basketball spatial tracking data.

1.2 Organization of the Dissertation

The structure of the dissertation is as follows. In Chapter 2, we introduce the new DTQ method to compute the likelihood function. First, we consider the likelihood computation of scalar SDE starting with formulating the likelihood for scenarios with a

single time-series data, and multiple time-series data separately. Then we detail the derivation of the DTQ method for both the scenarios. Finally, we generalized the DTQ method for coupled SDE.

In Chapter 3, we provide theoretical and empirical convergence results for the proposed DTQ method. Under some regularity conditions, we show that the DTQ method converges to the exact probability density function in L^1 norm error with a linear rate of convergent in temporal step size for a class of SDE. We also describe the results of numerical tests that confirm the empirical performance of the DTQ method is parallel to theoretical convergent results. In addition, we also discuss results from comparison tests of the DTQ method against a Fokker-Planck numerical solver.

Chapter 4 consists of methods to infer parameters of scalar SDE using maximum likelihood approach. In Chapter 4, we begin with introducing the optimization problem that minimizes the negative log likelihood. We derive two methods to compute the gradient of the negative log likelihood: the direct method and the adjoint method. We detail the two methods for two scenarios: one time-series and the multiple time-series data problems. Then we discuss results from the numerical tests carried out to test the performance of the DTQ method together with a direct gradient computation to solve the parametric inference problem. Lastly, we discuss an adjoint-based nonparametric method to infer parameters of SDE from repeated time series and/or high-dimensional longitudinal data.

In Chapter 5, we describe a Bayesian inference method for coupled SDE. We detail the derivation of Markov Chain Monte Carlo (MCMC) algorithm developed using a Metropolis scheme that uses the DTQ method compute the likelihood information. We discuss the performance of the methods providing results from simulation tests on a model of stochastic van der Pol oscillator. Then we detail the derivation of a coupled SDE to model fast break situations involve one runner and one chaser in a basketball game. We also provide results that validate the performance of the model through Bayesian inference of the model parameters from NBA tracking data.

Finally, Chapter 6 concludes the dissertation with a summary of the current work and possible directions for future research.

Chapter 2

Likelihood Computation with Density Tracking by Quadrature

2.1 Introduction

A key challenge in developing practical algorithms to perform Bayesian or frequentist inference of SDE is the computation of the likelihood. We address this problem through the use of a fast, convergent method to track the transition densities of the SDE. The inference method adapts well to scenarios in which the data consists of many samples at one point in time, or when the data consists of one or more time series. Consider the computation of the transition density $p_{X_{t_{j+1}}}(x_{j+1}|X_{t_j} = x_j, \theta)$. Here X_t stands for the state of a process that evolves forward in time via an SDE with parameter vector θ. We let x_j and x_{j+1} denote the true states of the system at times t_j and $T = t_{j+1}$. Let $p(x,t)$ denote the density function of X_t. Then one approach to approximating the transition density is to numerically solve the forward Kolmogorov (or Fokker-Planck) equation with the initial condition $p(x,t_j) = \delta(x - x_j)$ up to time t_{j+1}. Then $p(x_{j+1},T)$ will be a numerical approximation of the transition density. The Kolmogorov equation is a linear partial differential equation (PDE) with spatially-dependent coefficients. Our approach is similar in that we also numerically track the density $p(x,t)$ without sampling. Instead of numerically solving a PDE, we track the density by applying quadrature to the Chapman-Kolmogorov equation associated with a time-discretization of the SDE.

Hence we name our method as density tracking by quadrature (DTQ).

Other methods similar to ours are those of [49] and [55]. In these methods, one also starts with the Chapman-Kolmogorov equation for the Euler-Maruyama scheme applied to the SDE. However, instead of evaluating the resulting integrals by deterministic quadrature, Pedersen and Santa-Clara evaluate the integrals by Monte Carlo methods. These methods involve generating numerical sample paths of the SDE at times in between the observation times. This approach is problematic unless one generates sample paths conditional on both the initial condition $X_{t_j} = x_j$ and the final condition $X_{t_{j+1}} = x_{j+1}$. The work of [1] shares our goal of computing an accurate approximation of the exact transition density and resulting likelihood function. Instead of applying quadrature, Aït-Sahalia expands the transition density in a Gram-Charlier series and then computes the expansion coefficients up to a certain order.

This chapter is structured as follows: under the Section 2.2 we first introduce the likelihood function for scalar SDE when the data consists of one or more time series. In Section 2.2.3, we detail the DTQ method by carrying out the derivations for the cases where the data consists of either one or multiple sample paths. As we show, the DTQ method enables one to break the computation of the likelihood into a sum of likelihoods involving consecutive pairs of observations (t_j, x_j) and (t_{j+1}, x_{j+1}), and for each such pair, the DTQ method computes the likelihood using iterated matrix multiplication. Finally, in Section 2.3, we generalized the DTQ method for coupled SDE.

2.2 Likelihood Computation for Scalar Stochastic Differential Equations

In this section, we consider the likelihood computation of SDE (1.1). Let us suppose that the functional forms of f, and g are known. In other words, we consider the parametric inference problem. Suppose, we observe (1.1) at times $t_j = j\Delta t$ for some fixed time-step $\Delta t > 0$, for $j = 0, \ldots, L$. At each time t_j, we collect ν samples of X_{t_j} and label these samples as $x_j \in \mathbb{R}^\nu$. We let $\mathbf{x} = x_0, x_1, \ldots, x_L$ denote all of the collected observations. Our goal is to use $\mathbf{x}$ to infer θ. Key component of the inference problem is the computation of likelihood function $p(\mathbf{x}|\theta)$. Let us formulate the likelihood.

We first specify our notation. If $A_1,\ldots,A_N$ is a collection of random variables, then $p_{A_1,\ldots,A_N}(z_1,\ldots,z_N)$ denotes the joint probability density function of $A_1,\ldots,A_N$. Conditional densities will be denoted similarly. Also, note that we have assumed equispaced temporal observations in this problem. However, we make this assumption purely for notational simplicity; the method we describe can be easily adapted for nonequispaced temporal observations.

2.2.1 Likelihood for One Time Series

Now suppose we have collected only one observation at each point in time t_j, i.e., case where $\nu = 1$. Then the likelihood we seek to compute can be more accurately written as

$$p(\mathbf{x}|\theta) = p_{X_{t_L},\ldots,X_{t_0}}(x_L,\ldots,x_0|\theta).$$

First let us use the fact that the SDE (1.1) is an Ito diffusion and therefore satisfies the strong Markov property (see [12]). This enables us to write down our first expression for the likelihood function:

$$\mathscr{L}(\theta) = p_{X_{t_L},\ldots,X_{t_0}}(x_L,\ldots,x_0|\theta) = p_{X_{t_0}}(x_0)\prod_{j=0}^{L-1} p_{X_{t_{j+1}}}(x_{j+1}|X_{t_j}=x_j,\theta).$$

For notational simplicity only, here we assume $p_{X_{t_0}}(x_0) = 1$, i.e., a deterministic initial observation x_0. Therefore, the negative log likelihood of the observed time series is given by

$$-\log\mathscr{L}(\theta) = -\sum_{j=0}^{L-1}\log p_{X_{t_{j+1}}}(x_{j+1}|X_{t_j}=x_j;\theta), \tag{2.1}$$

where $p_{X_{t_{j+1}}}(x_{j+1}|X_{t_j}=x_j;\theta)$ is the conditional density of $X_{t_{j+1}}=x_{j+1}$ given $X_{t_j}=x_j$ (Also known as the transitional density).

2.2.2 Likelihood for Many Time Series

Here we consider the case where we have many observations at each point in time ($\nu > 1$). In other words, we reinterpret $\mathbf{x} = x_0,x_1,\ldots,x_L$ as a sequence of vector-valued observations. For each $s = 1,2,\ldots,\nu$, the sequence $x_0^s,x_1^s,\ldots,x_L^s$ is a scalar time series.

With these changes, the log likelihood becomes

$$\log \mathscr{L}(\theta) = \sum_{j=0}^{L-1} \sum_{r=1}^{v} \log p_{X_{j+1}}(x_{j+1}^r | X_j = \{x_j^s\}_{s=1}^v; \theta). \tag{2.2}$$

2.2.3 Numerical Method: Density Tracking by Quadrature

In this section, we introduce a new numerical method to compute the likelihood. Because this method computes an approximation to the conditional density via iterated quadrature, we refer to the method as DTQ. In what follows, we first consider the computation of the likelihood for the case where $v = 1$, i.e., we have collected only one observation at each point in time t_j. Later, in a subsequent section, we discuss the generalization of the method for the case where we have many observations at each point in time ($v > 1$).

Density Tracking by Quadrature for One Time Series

Let us start with approximating transition densities in (2.1). Each term in the product can be interpreted as follows: we start the SDE (1.1) with the initial condition $X_{t_j} = x_j$ and fixed parameter vector θ. We then solve for the probability density function of $X_{t_{j+1}}$, and evaluate that density function at x_{j+1}. By following these steps, we have calculated $p_{X_{t_{j+1}}}(x_{j+1} | X_{t_j} = x_j; \theta)$.

We now outline the DTQ method to compute the aforementioned probability density function. The first step of the DTQ method is to discretize (1.1) in time using the Euler-Maruyama scheme. We select an internal time step h, a small fraction of Δt, and set $hF = \Delta t$ where $F \in \mathbb{Z}$ and $F \geq 2$. Then the Euler-Maruyama discretization gives

$$\widetilde{X}_{j+n/F} \;=\; \widetilde{X}_{j+(n-1)/F} \;+\; \hat{f}(\widetilde{X}_{j+(n-1)/F}; \theta)h \;+\; \hat{g}(\widetilde{X}_{j+(n-1)/F}; \theta)h^{1/2} Z_{j+n/F}, \tag{2.3}$$

for $n = 1, \ldots, F$. Here $\{Z_{j+n/F}\}$ is an i.i.d. family of Gaussian random variables with mean 0 and variance 1. The random variable $\widetilde{X}_j$ is intended to approximate X_{t_j} when the index j is an integer. When the index j is not an integer, $\widetilde{X}_j$ represents a random variable that *interpolates in time* between the random variables that have been sampled to give us our data. The idea now is to approximate $p_{X_{t_{j+1}}}(x_{j+1} | X_{t_j} = x_j; \theta)$ in (2.1) with

$p_{\widetilde{X}_{j+1}}(x_{j+1}|\widetilde{X}_j = x_j;\theta)$. The Chapman-Kolmogorov equation for the Markov chain (2.3) is:

$$p_{\widetilde{X}_{j+n/F}}(y|\widetilde{X}_j = x_j;\theta) = \int_z p_{\widetilde{X}_{j+n/F}}(y|\widetilde{X}_{j+(n-1)/F} = z;\theta) \times p_{\widetilde{X}_{j+(n-1)/F}}(z|\widetilde{X}_j = x_j;\theta)\, dz. \tag{2.4}$$

Here y and z represent any value in the state spaces of the random variables $\widetilde{X}_{j+n/F}$ and $\widetilde{X}_{j+(n-1)/F}$, respectively. Now let $G^h_\theta(y,z)$ denote the probability density function of a Gaussian random variable with mean $z+\hat{f}(z;\theta)h$ and variance $\hat{g}(z;\theta)^2 h$, evaluated at y. From (2.3) we observe that, for each $n \in \{1,\ldots,F\}$,

$$p_{\widetilde{X}_{j+n/F}}(y|\widetilde{X}_{j+(n-1)/F} = z;\theta) = G^h_\theta(y,z), \tag{2.5}$$

and (2.4) becomes

$$p_{\widetilde{X}_{j+n/F}}(y|\widetilde{X}_j = x_j;\theta) = \int_z G^h_\theta(y,z) p_{\widetilde{X}_{j+(n-1)/F}}(z|\widetilde{X}_j = x_j;\theta)\, dz. \tag{2.6}$$

When $n = 1$ on the right-hand side, we see that conditioning on $\widetilde{X}_j = x_j$ forces $p_{\widetilde{X}_j}(z|\widetilde{X}_j = x_j;\theta) = \delta(z-x_j)$. This enables us to evaluate (2.6) at $n = 1$ to obtain

$$p_{\widetilde{X}_{j+1/F}}(y|\widetilde{X}_j = x_j;\theta) = G^h_\theta(y,x_j). \tag{2.7}$$

Starting with (2.7), we can now compute (2.6) iteratively to obtain the transition density $p_{\widetilde{X}_{j+n/F}}(y|\widetilde{X}_j = x_j;\theta)$. We compute (2.6) using trapezoidal quadrature rule.

Let us first truncate the infinite domain of the integral to $(-z^M, z^M)$, and introduce the spatial grid spacing $k > 0$ such that $k = (z^M)/M$. We use superscripts to denote spatial grid locations, for example, $z^{a_1} = a_1 k$ for all integers $a_1 \in \{-M,\ldots,0,\ldots,M\}$. By applying the trapezoidal rule to the right-hand side of (2.6), we get

$$p_{\widetilde{X}_{j+n/F}}(y|\widetilde{X}_j = x_j;\theta) \approx k \sum_{a_1=-M}^{M} G^h_\theta(y,z^{a_1}) p_{\widetilde{X}_{j+(n-1)/F}}(z^{a_1}|\widetilde{X}_j = x_j;\theta). \tag{2.8}$$

Let $kG^h_\theta(y^{a_2}, z^{a_1})$ be the (a_2,a_1) element of a matrix $K_{(2M+1)\times(2M+1)}$. We also define the a_2-th element of the vector $\hat{p}_{j+n/F}$ by $\hat{p}^{a_2}_{j+n/F} = p_{\widetilde{X}_{j+n/F}}(y^{a_2}|\widetilde{X}_j = x_j;\theta)$. Now (2.8) reduces to matrix-vector multiplication:

$$\hat{p}_{j+n/F} = K\hat{p}_{j+(n-1)/F}. \tag{2.9}$$

Starting with the initial vector $\hat{p}_{j+1/F}$, given by the right-hand side of (2.7) discretized on the spatial grid, we apply (2.9) $F-1$ times to get $\hat{p}_{j+1} = K^{F-1}\hat{p}_{j+1/F}$, where $\hat{p}_{j+1}$ is the approximation of the transition density function $p_{\widetilde{X}_{j+1}}(y|\widetilde{X}_j = x_j;\theta)$ on the spatial grid. To find the value of the transition density at $y = x_j$ one could use interpolation on $\hat{p}_{j+1}$. Instead of using interpolation, in our method we first compute $\hat{p}_{j+(F-1)/F}$ using (2.9), i.e.,

$$\hat{p}_{j+(F-1)/F} = K^{F-2}\hat{p}_{j+1/F}. \tag{2.10}$$

Now define the vector Γ_{F-1} by $\Gamma_{F-1}^{a_{F-1}} = kG_{\theta_0}^h(x_{j+1}, y^{a_{F-1}})$, where a_{F_1} is any integer between $-M$ and M. Let ^{T} denote transpose. Putting everything together, we obtain, respectively, the transition density and the negative log likelihood:

$$p_{\widetilde{X}_{j+1}}(x_{j+1}|\widetilde{X}_j = x_j, \theta) \approx [\Gamma_{F-1}]^T \hat{p}_{j+(F-1)/F}, \tag{2.11}$$

$$-\log \mathscr{L}(\theta) \approx -\sum_{j=0}^{L-1} \log\left([\Gamma_{F-1}]^T K^{F-2}\hat{p}_{j+1/F}\right). \tag{2.12}$$

Density Tracking by Quadrature for Many Time Series

Recall that the negative log likelihood for the case $\nu > 1$ is given by (2.2) in Section 2.2.2. Approximating each transitional density with Euler-Maruyama approximation we get the following approximation for the log likelihood.

$$\log \mathscr{L}(\theta) \approx \sum_{j=0}^{L-1}\sum_{a_F=1}^{\nu} \log p_{\widetilde{X}_{j+1}}(x_{j+1}^{a_F}|\widetilde{X}_j = \{x_j^s\}_{s=1}^{\nu};\theta). \tag{2.13}$$

The derivation of the negative log likelihood (2.13) is identical to that given in previous section for one time series. The only real change is that when $\nu > 1$, we use the samples $\{x_j^s\}_{s=1}^{\nu}$ of the random variable $\widetilde{X}_j$ to estimate the density of $\widetilde{X}_j$ as follows:

$$p_{\widetilde{X}_j}(z) \approx \frac{1}{\nu}\sum_{q=1}^{\nu} \delta(z - x_j^q). \tag{2.14}$$

This approximation is a density estimate that corresponds to the spatial derivative of the empirical cumulative distribution function of the samples. By logic analogous to (2.7),

we can then compute the initial density function $p_{\widetilde{X}_{j+1}}(y|\widetilde{X}_j = \{x_j^s\}_{s=1}^{\nu};\theta)$ by

$$p_{\widetilde{X}_{j+1/F}}(y|\widetilde{X}_j = x_j,\theta) = \int_z G_\theta^h(y,z)p_{\widetilde{X}_j}(z)\,dy \approx \frac{1}{\nu}\sum_{q=1}^{\nu} G_\theta^h(y,x_j^q). \tag{2.15}$$

We calculate the negative log likelihood by repeatedly applying (2.4), starting with the initial density given by (2.15) for $n = 2$. We make the approximation (2.14) so that we can evolve the density along each sample path with the same initial condition. Otherwise, we would have to repeat the calculation (2.12) ν times.

Now let us redefine $\hat{p}_{j+1/F}$ such that its a_1-th element is (2.15) evaluated at the spatial grid point y^{a_1}. Then we can use (2.10) to compute $\hat{p}_{j+(F-1)/F}$. We also redefine Γ_{F-1} to be a matrix of dimension $\nu \times (2M+1)$ whose (a_F, a_{F-1}) entry is $\Gamma_{F-1}^{a_F,a_{F-1}} = kG_\theta^h(x_{j+1}^{a_F}, y^{a_{F-1}})$. Note that the superscript in $x_{j+1}^{a_F}$ is used to denote the a_F-th sample observation taken at time t_{j+1}, whereas the superscript in $y^{a_{F-1}}$ denotes the spatial grid location. With these changes, (2.12) becomes

$$-\log\mathscr{L}(\theta) \approx -\sum_{j=0}^{L-1}\sum_{a_F=1}^{\nu}\log\left(\Gamma_{F-1}\hat{p}_{j+(F-1)/F}\right)_{a_F}. \tag{2.16}$$

Note that the drift f and g in the SDE (1.1) depends only on X_t and does not explicitly depend on time t. Even though, we have discussed the likelihood computation for this specific type of SDE under this Section 2.2, the DTQ method is easily generalizable for SDE with time-dependent drift and diffusion. In Section 2.3 we extend the DTQ method for coupled SDE with time-dependent drift and diffusion functions.

2.3 Likelihood for Coupled Stochastic Differential Equations

Let $W_{1,t}$ and $W_{2,t}$ denote two independent Wiener processes with $W_{1,0} = W_{2,0} = 0$ almost surely. In this section, we deal with coupled SDE of the form:

$$\mathrm{d}X_{1,t} = f_1(t,\mathbf{X}_t,\theta)\,\mathrm{d}t + g_1(t,\mathbf{X}_t,\theta)\,\mathrm{d}W_{1,t} \tag{2.17a}$$

$$\mathrm{d}X_{2,t} = f_2(t,\mathbf{X}_t,\theta)\,\mathrm{d}t + g_2(t,\mathbf{X}_t,\theta)\,\mathrm{d}W_{2,t}. \tag{2.17b}$$

Here $\mathbf{X}_t = (X_{1,t}, X_{2,t})$ is a two-dimensional stochastic process. For $j = 1,2$, we refer to f_j and g_j as, respectively, drift and diffusion functions. Both drift and diffusion functions may depend on a parameter vector $\theta \in \mathbb{R}^N$.

Our goal is to infer θ from discrete-time observations of $\mathbf{X}_t$. Suppose that at a sequence of times $0 = t_0 < t_1 < \cdots < t_L = T$, we have observations $\mathbf{x} := \{(x_{1,m}, x_{2,m})\}_{m=0}^{L}$. Here $\mathbf{x}_m = (x_{1,m}, x_{2,m})$ is a sample of $\mathbf{X}_{t_m}$. Again we assume equispaced temporal observations here, i.e., $t_m = m\Delta t$ for fixed step size $\Delta t > 0$. We discretize the SDE (2.17) in time using the Euler-Maruyama scheme:

$$X_1^{n+1} = X_1^n + f_1(t_n, X_1^n, X_2^n, \theta)h + g_1(t_n, X_1^n, X_2^n, \theta)\sqrt{h}Z_1^{n+1} \tag{2.18a}$$

$$X_2^{n+1} = X_2^n + f_2(t_n, X_1^n, X_2^n, \theta)h + g_2(t_n, X_1^n, X_2^n, \theta)\sqrt{h}Z_2^{n+1}. \tag{2.18b}$$

Here $h > 0$ is a fixed time step, the time step of our numerical method. We shall choose h to be a fraction of Δt, i.e., $Fh = \Delta t$ for integer $F \geq 2$. The random variables X_i^n for $i = 1,2$ are approximations of $X_{i,nh}$. The Z_i^n are independent and identically distributed random variables, normally distributed with mean 0 and variance 1, i.e., $Z_i^n \sim \mathcal{N}(0,1)$.

Let $\widetilde{p}(\mathbf{x} \,|\, \theta)$ denote the likelihood under the discrete-time model (2.18), an approximation to the true likelihood $p(\mathbf{x} \,|\, \theta)$. Note that (2.18) describes a discrete-time Markov chain. By the Markov property, the likelihood $\widetilde{p}(\mathbf{x} \,|\, \theta)$ factors and we can write:

$$p(\mathbf{x} \,|\, \theta) \approx \widetilde{p}(\mathbf{x} \,|\, \theta) = \prod_{m=0}^{L-1} \widetilde{p}(\mathbf{x}_{m+1} \,|\, \mathbf{x}_m, \theta). \tag{2.19}$$

The term $\widetilde{p}(\mathbf{x}_{m+1} \,|\, \mathbf{x}_m, \theta)$ is the transition density for (2.18), from state $\mathbf{x}_m$ at time t_m to state $\mathbf{x}_{m+1}$ at time t_{m+1}. In the next subsection, we discuss how to compute this density using the DTQ method.

Density Tracking by Quadrature (DTQ)

Equation (2.18) describes a Markov chain over a continuous state space. If we let $\widetilde{p}^n(x_1, x_2 \,|\, \theta)$ denote the joint probability density function of X_1^n and X_2^n given θ, then the Chapman-Kolmogorov equation associated with (2.18) is

$$\widetilde{p}^{n+1}(x_1, x_2 \,|\, \theta) = \int_{y_1, y_2 \in \mathbb{R}^2} K(x_1, x_2, y_1, y_2, t_n; \theta)\widetilde{p}^n(y_1, y_2 \,|\, \theta)\, dy, \tag{2.20}$$

where

$$K(x_1,x_2,y_1,y_2,t_n;\theta) = \widetilde{p}^{n+1|n}(x_1,x_2|y_1,y_2,\theta)$$
$$= (2\pi\sigma_1^2)^{-1/2}\exp\left[-(x_1-\mu_1)^2/(2\sigma_1^2)\right](2\pi\sigma_2^2)^{-1/2}\exp\left[-(x_2-\mu_2)^2/(2\sigma_2^2)\right].$$

Here $\mu_1 = y_1 + f_1(t_n,y_1,y_2;\theta)h, \mu_2 = y_2 + f_2(t_n,y_1,y_2;\theta)h, \sigma_1^2 = g_1^2(t_n,y_1,y_2;\theta)h$ and $\sigma_2^2 = g_2^2(t_n,y_1,y_2;\theta)h$. That is, $K(x_1,x_2,y_1,y_2,t_n;\theta)$ is the conditional density of X_1^{n+1} and X_2^{n+1} given $X_1^n = y_1$, $X_2^n = y_2$ and a fixed θ, evaluated at the point (x_1,x_2). The fact that the conditional density is a product of normal distributions with means μ_1,μ_2 and variances σ_1^2,σ_2^2 can be shown using (2.18) together with the fact that X_1^{n+1} and X_2^{n+1} are conditionally independent given X_1^n and X_2^n. This conditional independence is a direct consequence of having two independent random variables Z_1^n and Z_2^n in (2.18).

Now let us apply quadrature to (2.20) to evolve an initial density forward in time. Consider a $(2M+1)\times(2M+1)$ spatial grid with fixed spacing $k>0$ and grid points $x_1^i = ik$, $x_2^j = jk$, $y_1^{i'} = i'k$, and $y_2^{j'} = j'k$, where $i,i',j,j' \in \{-M,M\}$. Then we apply the trapezoidal rule in both the y_1 and y_2 variables to obtain:

$$\hat{p}^{n+1}(x_1^i,x_2^j;\theta) = k^2\sum_{i'=-\infty}^{\infty}\sum_{j'=-\infty}^{\infty} K(x_1^i,x_2^j,y_1^{i'},y_2^{j'},t_n;\theta)\hat{p}^n(y_1^{i'},y_2^{j'};\theta) \tag{2.21}$$

It is unnecessary to sum over all of $\mathbb{Z}^2$. We know that a two-dimensional Gaussian decays to zero far from its mean. Since the mean (μ_1,μ_2) is approximately (y_1,y_2), we sum only from $y_1 = x_1 - \zeta k$ to $y_1 = x_1 + \zeta k$ and similarly for y_2:

$$\hat{p}^{n+1}(x_1^i,x_2^j;\theta) = k^2\sum_{i'=i-\zeta}^{i+\zeta}\sum_{j'=j-\zeta}^{j+\zeta} K(x_1^i,x_2^j,y_1^{i'},y_2^{j'},t_n;\theta)\hat{p}^n(y_1^{i'},y_2^{j'};\theta) \tag{2.22}$$

We choose ζ manually to ensure the accuracy of the computation. We now have our method to evaluate $\widetilde{p}(\mathbf{x}_{m+1}\,|\,\mathbf{x}_m,\theta)$. Let us take $n=0$ in (2.22) to correspond to the time t_m. We start with the deterministic initial condition $\mathbf{X}^0 = \mathbf{x}_m$, corresponding to the density $\widetilde{p}^0(\mathbf{x}) = \delta(\mathbf{x}-\mathbf{x}_m)$. Inserting this point mass into (2.20), we obtain a Gaussian density for $\widetilde{p}^1(\mathbf{x})$. For each $i,j \in \{-M,M\}$ on the spatial grid, we set $\hat{p}^1(x_1^i,x_2^j;\theta) = \widetilde{p}^1(x_1^i,x_2^j;\theta)$. Now that we have $\hat{p}^1$, we use (2.22) repeatedly to compute $\hat{p}^2$, $\hat{p}^3$, and so on until we reach $\hat{p}^F$. The object $\hat{p}^F$ is then a spatially discrete approximation of the transition density from time t_m to time $t_m + Fh = t_{m+1}$. For this last density, instead of

evaluating it on the spatial grid used by the trapezoidal rule, we evaluate the density at the data $\mathbf{x}_{m+1}$ to avoid interpolation. In this way, we compute a numerical approximation of $\widetilde{p}(\mathbf{x}_{m+1} \,|\, \mathbf{x}_m, \boldsymbol{\theta})$, as required for the likelihood function.

Chapter 3

Convergence Analysis of Density Tracking by Quadrature

3.1 Introduction

In Chapter 2, we discussed the role of the DTQ method in computing the likelihood via approximations of transition densities. As already mentioned in Section 2.2.3, calculating the transition density $p_{X_{t_{j+1}}}(x_{j+1}|X_{t_j} = x_j;\theta)$ for the SDE

$$dX_t = f(X_t;\theta)dt + g(X_t;\theta)dW_t \tag{3.1}$$

can be seen as follows: fixing the parameter vector θ, we solve the SDE (3.1) for the probability density function of X_t at time $t = t_{j+1}$ starting with the initial condition $X_{t_j} = x_j$. We then evaluate that density function at x_{j+1}. Therefore, the DTQ method is actually a numerical method to computing the probability density function of X_t for a given SDE. Note that when computing the likelihood for a given parameter vector θ, it is no longer required to consider the θ-dependence of the drift and the diffusion. As discussed in Chapter 2, for the inference problem, we have constant deterministic initial conditions when computing the transition densities. However, the DTQ method can be used for SDE with random initial conditions as well. In this chapter, we establish convergence properties for the DTQ method as a more general method of computing the probability density function of a class of SDE at a given time t. Therefore, this chapter is written independently from the parameter inference of SDE. More specifically, here

we consider parameter inference as an application of the DTQ method.

Consider the SDE for the scalar process X_t,

$$dX_t = f(X_t)dt + g(X_t)dW_t, \tag{3.2}$$

where W_t is the Wiener process. X_t is an Itô diffusion; neither the drift f nor the diffusion g feature explicit time-dependence. Assuming regularity of f and g, the process X_t has a probability density function $p(x,t)$ [52]. In this chapter, we consider the DTQ method as a numerical method to solve for p. Let us recall the three main steps in the derivation of the DTQ method:

1. Discretize the SDE (3.2) in time using a convergent stochastic time-stepping method.

2. Interpret the time-discretized equation as a discrete-time Markov chain on a continuous state space; let $\tilde{p}$ denote its probability density function. We can then write down a Chapman-Kolmogorov equation that enables us to evolve $\tilde{p}$ forward in time.

3. Discretize both the Chapman-Kolmogorov equation and $\tilde{p}$ in space, e.g., using a spatial grid and numerical quadrature. Let $\hat{p}$ denote the discrete-space approximation of $\tilde{p}$.

We emphasize that these steps form a framework that encompasses many possible algorithms. In this work, we use the explicit Euler-Maruyama method in step 1 and the trapezoidal rule in step 3; unless stated otherwise, this is the DTQ method analyzed here. Had we made different choices in these steps, we would have obtained a different method in the DTQ family.

In this chapter, we prove that $\hat{p}$ converges to p as the discretization parameters tend to zero. Because there are existing results on the convergence of $\tilde{p}$ to p, the main task of this work is to show that $\hat{p} \to \tilde{p}$. More specifically, the foundational work of Bally and Talay [7] established conditions under which $\tilde{p}$ converges to p, in the case where the Euler-Maruyama method is used to discretize the SDE (3.2) in time. Let $\|f\|_1$ denote the L^1 norm of a function f. Suppose we seek the density of (3.2) at time $T > 0$. Let

$h > 0$ denote the temporal step size; as we take $h \to 0$, we assume $T = Nh$ stays fixed. Then the results of [7] imply that $\|p(\cdot,T) - \tilde{p}(\cdot,T)\|_1 = O(h)$.

Our work builds on this result. The DTQ method analyzed here combines Euler-Maruyama temporal discretization with the trapezoidal rule on an equispaced grid. This results in a fast, simple method to compute an approximation $\hat{p}$ such that $\|\tilde{p}(\cdot,T) - \hat{p}(\cdot,T)\|_1 = O(h^{-1}\exp(-rh^{-\kappa}))$ for positive constants r, κ. The user of the method can control κ by adjusting the relationship between the spatial and temporal grid spacings.

As we have already seen in previous chapters, the primary application of this work is in statistical inference for diffusion processes. In particular, we have already used the DTQ method to devise both Bayesian and frequentist inference algorithms. The present work lays a theoretical foundation for these statistical applications. Additionally, note that when inference procedures for diffusions have been compared, a method that approximates the likelihood by numerically solving the Fokker-Planck (or Kolmogorov) equation achieves superior accuracy at the cost of excessive computational time [30]. The results of the present work indicate that the DTQ method achieves the same accuracy as a Fokker-Planck solver with less computational effort, further motivating the potential use of the DTQ method in many inference applications.

We now review alternative approaches and prior work related to either the general problem of computing the density of (3.2), or the particular case of the DTQ method.

3.1.1 Alternative Approaches

If the drift f and diffusion g are sufficiently smooth, then p satisfies the forward Kolmogorov (or Fokker-Planck) equation [52]:

$$\frac{\partial}{\partial t}p(x,t) = -\frac{\partial}{\partial x}[f(x,t)p(x,t)] + \frac{1}{2}\frac{\partial^2}{\partial x^2}[g^2(x,t)p(x,t)]. \tag{3.3}$$

Prescribing an initial condition $p(x,0)$, we may then solve (3.3) to obtain the density $p(x,T)$ at time $T > 0$. The solution of (3.3) must satisfy the normalization condition $\int_{x=-\infty}^{\infty} p(\cdot,t)\,dx = 1$, which implies boundary conditions of the form $\lim_{|x|\to\infty} p(x,t) = 0$.

We view the DTQ method as an alternative to numerical methods for the solution of (3.3). The primary purpose of the present work is to demonstrate intrinsic properties—both theoretical and empirical—of the DTQ method. However, in the present work, we

also compare the performance of the DTQ method against an elementary finite difference method applied to (3.3). The finite difference method we consider is first-order in time and second-order in space. For a particular test problem at the finest grid resolution we consider, the DTQ method computes a solution with L^1 error $\approx 3 \times 10^{-3}$ more than 100 times faster than our Fokker-Planck method.

Besides the numerical solution of (3.3), another method one might use to estimate the density of (3.2) involves sampling. Specifically, one can employ any convergent numerical method to step (3.2) forward in time from $t = 0$ to $t = T$, thereby generating one sample of X_T. Repeating this procedure many times, one can obtain enough samples of X_T to compute a statistical estimate of the density at time T. For instance, one could compute a histogram or a kernel density estimate. Several methods in the literature can be viewed as special cases and extensions of this approach [29, 35, 43, 27]. In such methods, the accuracy of the density will be controlled by two parameters: the temporal step size and the number of sample paths. If there are N_S samples, then a typical stochastic time-stepping method will contribute an error of $N_S^{-1/2}$ and kernel density estimation will contribute an error of, e.g., $N_S^{-4/5}$. In comparison, the DTQ method's accuracy is also controlled by two parameters, the temporal step size and the spatial grid size. However, the spatial discretization using the trapezoidal rule on the real line will contribute an error that decays exponentially in the spatial grid size [62]. For this reason, we believe the DTQ method will be a strong alternative to a sampling-based method.

Returning to the forward Kolmogorov or Fokker-Planck equation (3.3), we see that smoothness of f and g is required in order to have classical solutions. The implementation of the DTQ method itself does not utilize derivatives (whether exact or approximate) of f and g. At the same time, the reader will note that our convergence theory assumes analyticity of f and g on a strip in the complex plane that contains the real line. We give two reasons for assuming analyticity. First, many models of scientific interest involve functions f and g that do satisfy these hypotheses. Second, in order to apply exponential error estimates for the trapezoidal rule [62], it is essential that our integrand, which depends on f and g, be analytic on a strip.

Ultimately, we do expect that the hypotheses in the present convergence proof can

be relaxed. Let $\mathring{p}$ be an approximate density that is computed in exactly the same way as $\hat{p}$ except for truncation of the infinite domain/series. Our empirical results clearly show first-order convergence of $\mathring{p}$ to p, even when not all of the hypotheses of our theorem are satisfied. Suppose that, inspired by these results, we discover how to prove convergence of $\hat{p}$ to $\tilde{p}$ assuming, for instance, that both f and g possess merely 4 bounded continuous derivatives. This will not immediately improve our ability to conclude that $\hat{p}$ converges to p; the existing result on convergence of $\tilde{p}$ to p requires that both f and g are C^∞ with bounded derivatives of all orders [7]. To make true progress on the problem, we must relax the conditions of convergence for both $\hat{p} \to \tilde{p}$ and $\tilde{p} \to p$. This is outside the scope of the present work.

3.1.2 Prior Work

When we derive the DTQ method, we make use of the fact that a time-discretization of (3.2) can be viewed as a discrete-time Markov chain on a continuous state space. Suppose we were to take a different point of view, that of trying to design a discrete-time Markov chain on a discrete state space whose law or density approximates well that of the original SDE. In this case, there are extensive results going back to the work of H. J. Kushner [38]. Like a discrete-time, discrete-time Markov chain, the DTQ algorithm can be written in the form $\hat{p}(t_{n+1}) = A\hat{p}(t_n)$, where A is a matrix (possibly with an infinite number of rows and columns) and $\hat{p}(t_j)$ represents the approximate density at time t_j. However, because of the quadrature-based derivation of the DTQ algorithm, the matrix A is, in general, not a Markov transition matrix. We find it both mathematically interesting and practically useful that, in spite of this, the DTQ method's $\hat{p}$ converges exponentially to $\tilde{p}$.

The Chapman-Kolmogorov equation that is at the center of this chapter—see (3.7)—has also appeared in other papers [49, 55]. In these works, the right-hand side of the Chapman-Kolmogorov equation is interpreted as an expected value that can be computed stochastically, i.e., using Monte Carlo methods. In our approach, we use deterministic quadrature to evaluate the right-hand side of the Chapman-Kolmogorov equation. There is only one prior paper we found that features this approach, albeit in a different context, that of a nonlinear autoregressive time series model [15]. The conver-

gence results in [15] are of a different nature than ours, because they involve taking the continuum limit in space but *not* in time. In the present work, we are interested in the error made by the DTQ method as both the temporal and spatial grid spacings vanish.

3.1.3 Summary of Results and Outline

The main result of this work is a provably convergent method for computing an approximation $\hat{p}$ of the density p for the SDE (3.2). Let $h > 0$ and $k > 0$ denote, respectively, the temporal and spatial step sizes. Assume that $k \propto h^{\rho}$ for $\rho > 1/2$, and assume that f and g are sufficiently regular (more precisely, admissible in the sense of Definition 2). Under these conditions, in Sections 3.4 and 3.5, we prove that $\hat{p}$ converges to $\tilde{p}$ in L^1, and that the error decays exponentially in h. Specifically, there exists a constant $r > 0$ such that the leading order L^1 error term is proportional to $h^{-1}\exp(-rh^{1/2-\rho})$—see Theorem 2. As a consequence of this result and the results of [7], we conclude that $\hat{p}$ converges to p in L^1, and that the error decays linearly with h—see Corollary 1.

Up to and including Section 3.5, our results pertain to an idealized version of the DTQ algorithm in which we track the density $\hat{p}$ at an infinite number of discrete grid points. In Section 3.6, we study the effect of boundary truncation. Our main tool in this section is a Chernoff bound on the tail sum of $\hat{p}$ that we establish through the moment generating function. Let $\mathring{p}$ denote the approximation of $\hat{p}$ obtained by summing over precisely $2M+1$ grid points from $-y_M = -Mk$ to $y_M = Mk$. The quantity $\mathring{p}$ is what we actually compute when we implement the DTQ method. In Lemma 9, we show that if $y_M \to \infty$ at a logarithmic rate, i.e., $y_M \propto \log h^{-1}$, then the L^1 error between $\mathring{p}$ and $\hat{p}$ is $O(h)$. Combining this with our earlier results, this establishes L^1 convergence of $\mathring{p}$ to the true density p—see Corollary 2.

In Section 3.7, we study the performance of the DTQ method. For a suite of six test problems for which we have access to the exact solution, our numerical tests confirm $O(h)$ convergence of $\mathring{p}$ to p. This remains true for drift f and diffusion functions g that do not strictly satisfy the hypotheses of our convergence theory. We also present a finite difference method for solving (3.3); we compare this method against three slightly different implementations of the DTQ method. The comparison indicates that the DTQ method—which we believe is being analyzed here for the first time—is competitive with

standard numerical methods for (3.3).

Before proceeding, we give a more detailed derivation of the DTQ method in Section 3.2 and then introduce necessary assumptions and notation in Section 3.3.

3.2 Problem Setup

We begin with a more detailed derivation of the DTQ method. First, we discretize (3.2) in time using the explicit Euler-Maruyama method:

$$x_{n+1} = x_n + f(x_n)h + g(x_n)\sqrt{h}Z_{n+1}, \tag{3.4}$$

where $h > 0$ is a fixed time step and Z_{n+1} is a random variable with a standard (mean zero, variance one) Gaussian distribution. We let $\tilde{p}(x,t_n)$ denote the probability density function of x_n. Note that this differs from $p(x,t_n)$.

From (3.4), we observe that the density of x_{n+1} given $x_n = y$ is Gaussian with mean $y + f(y)h$ and variance $hg^2(y)$. Let us denote this conditional density by $p_{n+1,n}(x|y)$; then

$$p_{n+1,n}(x|y) = G(x,y) := \frac{1}{\sqrt{2\pi g^2(y)h}} \exp\left(-\frac{(x-y-f(y)h)^2}{2g^2(y)h}\right). \tag{3.5}$$

Note that, for any $y \in \mathbb{R}$,

$$\int_{x=-\infty}^{\infty} G(x,y)\,dx = 1. \tag{3.6}$$

Applying this to (3.4), we obtain the following evolution equation:

$$\tilde{p}(x,t_{n+1}) = \int_{-\infty}^{\infty} p_{n+1,n}(x|y)\tilde{p}(y,t_n)\,dy. \tag{3.7}$$

This is the Chapman-Kolmogorov equation for the discrete-time, continuous-space Markov chain given by (3.4). Similar equations are often employed in the literature on inference for diffusions—see [49], [55], [26, Chap 6.3.3], and [36].

Let us define an equispaced temporal grid by $t_n = nh$ with $h = T/N$. In principle, we can now repeatedly apply (3.7) to determine $\tilde{p}(x,T)$. This assumes we can perform the integral over the real line. To compute (3.7) in practice, we use numerical quadrature. Here we employ the trapezoidal rule, enabling us to make use of exponential error estimates [62, 59, 40]. To begin with, we apply the trapezoidal rule on the real line. Later,

we explain how to incorporate the effects of a finite, truncated integration domain. Assume the domain $\mathbb{R}$ is discretized via an equispaced grid $y_j = jk$ where $k > 0$ is fixed. Then our discrete-time, discrete-space evolution equation is

$$\hat{p}(x,t_{n+1}) = k \sum_{j=-\infty}^{\infty} G(x,y_j)\hat{p}(y_j,t_n). \tag{3.8}$$

Except for the fact that we have not yet truncated the infinite sum, this is the DTQ method.

Thus far we have avoided the discussion of initial conditions for both $\tilde{p}$ and $\hat{p}$. For the purposes of exposition, we assume a constant initial condition $X_0 = C$, which implies $p(x,0) = \tilde{p}(x,0) = \delta(x-C)$. This choice is not essential to either the use or convergence of the DTQ method. In fact, the choice of a point mass initial condition requires special handling, because we cannot discretize $\tilde{p}(x,0)$ directly. We insert $n = 0$ into (3.7), use $\tilde{p}(x,0) = \delta(x-C)$, and obtain the non-singular initial condition

$$\hat{p}(x,t_1) = \tilde{p}(x,t_1) = G(x,C). \tag{3.9}$$

This enables us to initialize and iteratively use both (3.7) and (3.8) for $n \geq 1$.

Our main task in Sections 3.4 and 3.5 is to estimate $\|\hat{p}(\cdot,T) - \tilde{p}(\cdot,T)\|_1$. Before we start the proof of Theorem 2, we introduce necessary notation and assumptions.

3.3 Notation and Assumptions

We will use the Roman i for the imaginary unit ($\mathrm{i} = \sqrt{-1}$) and reserve the Italic i for an index of summation. We denote the L^1 norm of a function $f : \mathbb{R} \to \mathbb{R}$ by

$$\|f\|_1 = \int_{-\infty}^{\infty} |f(x)|\,dx.$$

We denote the ℓ^1 norm of the sequence $\{z_j\}_{j=-\infty}^{\infty}$ by

$$\|z\|_{\ell^1} = \sum_{j=-\infty}^{\infty} |z_j|.$$

For a function $f : \mathbb{R} \to \mathbb{R}$, we understand $\|f\|_{\ell^1}$ to be the norm of the sequence obtained by applying f on the spatial grid:

$$\|f\|_{\ell^1} = \sum_{j=-\infty}^{\infty} |f(jk)|,$$

where again $k > 0$ denotes the grid spacing. We use $\lceil x \rceil$ to denote the smallest integer greater than or equal to x, and $\lfloor x \rfloor$ to denote the largest integer less than or equal to x. The following definition is from the literature [40].

Definition 1. *For $a > 0$, let S_a denote the infinite strip of width $2a$ given by*

$$S_a = \{z \in \mathbb{C} : \ |\Im(z)| < a\}.$$

Then $B(S_a)$ is the set of functions such that $\varphi \in B(S_a)$ iff φ is analytic in S_a,

$$\int_{-a}^{a} |\varphi(x+\mathrm{i}y)|\,dy = O(|x|^{\alpha}), \quad x \to \pm\infty, \quad 0 \le \alpha < 1, \tag{3.10}$$

and

$$\mathscr{N}(\varphi, S_a) \equiv \lim_{y \to a^-} \left\{ \int_{-\infty}^{\infty} |\varphi(x+\mathrm{i}y)|\,dx + \int_{-\infty}^{\infty} |\varphi(x-\mathrm{i}y)|\,dx \right\} < \infty. \tag{3.11}$$

The next definition encapsulates the constraints that the coefficient functions f and g in the original SDE (3.2) must satisfy in order for us to show exponential convergence of $\hat{p}$ to $\tilde{p}$.

Definition 2. *In this work, we say that f and g are* admissible *if they satisfy the following properties. First, there exists $d > 0$ such that f and g are analytic on the strip S_d. Additionally, there exist positive, finite, real constants M_1, M_2, M_3, and M_4 such that for all $z \in S_d$,*

$$|f'(z)| \le M_1 \tag{3.12a}$$

$$M_2 \le |g(z)| \le M_3 \tag{3.12b}$$

$$\Re(g(z)) \ne 0 \tag{3.12c}$$

$$|g'(z)| \le M_4. \tag{3.12d}$$

We now state a theorem that gives an exponential error estimate for the trapezoidal rule [40], one that we shall use to bound the error made in one step of the DTQ method. Other error estimates can be found in the literature [59, 62].

Theorem 1. *Suppose $\varphi \in B(S_d)$ and $k > 0$. Let*

$$\eta = \int_{-\infty}^{\infty} \varphi(x)\,dx - k \sum_{j=-\infty}^{\infty} \varphi(jk).$$

Then

$$|\eta| \leq \frac{\mathscr{N}(\varphi, S_d)}{2\sinh(\pi d/k)} \exp(-\pi d/k).$$

Proof. See [40, Theorem 2.20]. □

3.4 Preliminary Theory

In this section, we prove several lemmas that are essential ingredients for the convergence theorem in Section 3.5. The overall goal of these lemmas is to show that the integrand

$$\varphi(x, y, t_n) = G(x, y)\hat{p}(y, t_n), \tag{3.13}$$

considered as a function of y for the purposes of quadrature, satisfies the hypotheses of Theorem 1.

The first lemma enables us to pass from an estimate of the error made in one time step to an estimate of the error made across a non-zero interval of time, even as the number of time steps becomes infinite.

Lemma 1. *Suppose that* $\xi(h) \geq 0$ *satisfies* $\lim_{h\to 0^+} \xi(h) = 0$. *Suppose there exist* $\gamma > 1$, $\varepsilon > 0$ *and* $h_0 > 0$ *such that* $\xi(h) \leq \varepsilon h^\gamma$ *for all* $h < h_0$. *Fix* $T > 0$, $N \in \mathbb{N}^+$, *and let* $h = T/N$. *Then*

$$\lim_{N\to\infty} \left[h \sum_{j=0}^{N-1} (1 + \xi(h))^j \right] = T.$$

Proof. Take N sufficiently large so that $h < 1$ and $h < h_0$. Then we calculate

$$\sum_{j=0}^{N-1} (1 + \xi(h))^j = \xi(h)^{-1} \left[(1 + \xi(h))^N - 1 \right] = \sum_{j=1}^{N} \binom{N}{j} \xi(h)^{j-1}$$
$$\leq \frac{T}{h} + \sum_{j=2}^{N} \frac{T^j \varepsilon^{j-1}}{j!} h^{\gamma(j-1)-j}$$

Using $h < 1$, we have

$$h \sum_{j=0}^{N-1} (1 + \xi(h))^j \leq T + \sum_{j=2}^{N} \frac{T^j \varepsilon^{j-1}}{j!} h^{(\gamma-1)(j-1)} \leq T + \varepsilon^{-1} h^{\gamma-1} \exp(T\varepsilon).$$

We have shown that the limit is T, and that the correction term to the limit is $O(h^{\gamma-1})$.

□

Next, we estimate the ℓ^1 norm of the discrete Gaussian. This estimate is standard, but we include it here for the sake of completeness.

Lemma 2. *For all $y \in \mathbb{R}$ and all $h,k>0$,*

$$k\|G(\cdot,y)\|_{\ell^1} \le 1+4\exp\left(-\frac{2\pi^2 g^2(y)h}{k^2}\right). \tag{3.14}$$

Proof. Let

$$\phi(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \tag{3.15}$$

Note that for any $d>0$, on the strip S_d, ϕ satisfies the hypotheses of Theorem 1. In particular,

$$\int_{-\infty}^{\infty}\left|\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x+\mathrm{i}d-\mu)^2}{2\sigma^2}\right)\right|dx = \exp\left(\frac{d^2}{2\sigma^2}\right).$$

As the right-hand side does not change when we replace d by $-d$, we have $\mathscr{N}(\phi,S_d) = 2\exp(d^2/(2\sigma^2))$. Therefore, applying Theorem 1,

$$\left|\int_{-\infty}^{\infty}\phi(x)\,dx - k\sum_{j=-\infty}^{\infty}\phi(jk)\right| \le \frac{\exp\left(d^2/(2\sigma^2)\right)}{\sinh(\pi d/k)}\exp\left(-\frac{\pi d}{k}\right) \le 4\exp\left(\frac{d^2}{2\sigma^2}-\frac{2\pi d}{k}\right),$$

where we have used $(\sinh(\pi d/k))^{-1} \le 4\exp(-\pi d/k)$. The right-hand side is minimized at $d = 2\pi\sigma^2/k$. Also, $\int_{-\infty}^{\infty}\phi(x)\,dx = 1$. Hence

$$k\sum_{j=-\infty}^{\infty}\phi(jk) \le 1+4\exp\left(-\frac{2\pi^2\sigma^2}{k^2}\right). \tag{3.16}$$

Note that $\phi(x) = G(x,y)$ with $\mu = y+f(y)h$ and $\sigma^2 = g^2(y)h$. Then (3.16) is (3.14). □

For each t_n, we think of $\{\hat{p}(x_j,t_n)\}_{j=-\infty}^{\infty}$ as an infinite sequence. It is important to estimate the ℓ^1 norm of this sequence.

Lemma 3. *If g is admissible in the sense of Definition 2, then for all $h,k>0$,*

$$\|\hat{p}(\cdot,t_{n+1})\|_{\ell^1} \le \|\hat{p}(\cdot,t_1)\|_{\ell^1}(1+4\exp(-2\pi^2M_2^2h/k^2))^n. \tag{3.17}$$

Proof. We begin by evaluating (3.8) at $x = x_i$:

$$\hat{p}(x_i,t_{n+1}) = k\sum_{j=-\infty}^{\infty}G(x_i,y_j)\hat{p}(y_j,t_n). \tag{3.18}$$

Before proceeding, let us discuss the convergence of the infinite series on the right-hand side for fixed h and k. Using (3.12b), we have for G the elementary bound $0 \leq G(x,y) \leq (2\pi M_2^2 h)^{-1/2}$. Note that (3.9) and (3.14) together give us an ℓ^1 bound on $\{\hat{p}(jk,t_1)\}_{j=-\infty}^{\infty}$. Combining these two bounds, it is clear that (3.18) converges for $n=1$. Now, as an induction hypothesis, assume that for a particular $n \geq 1$, we have $\hat{p}(y,t_n) \geq 0$ and that $\|\hat{p}(\cdot,t_n)\|_{\ell^1} < \infty$. We will establish an ℓ^1 bound for $\hat{p}(\cdot,t_{n+1})$.

By the induction hypothesis, we know that the infinite series on the right-hand side of (3.18) converges. We see that all terms in the infinite series are nonnegative, so $\hat{p}(y,t_{n+1}) \geq 0$. Additionally, both sides of (3.18) do not change upon taking absolute values. We sum over all i and interchange the order of summation—this is justified because, again, all terms are nonnegative. We obtain

$$\|\hat{p}(\cdot,t_{n+1})\|_{\ell^1} = \sum_{j=-\infty}^{\infty} \left[k \sum_{i=-\infty}^{\infty} G(x_i,y_j) \right] \hat{p}(y_j,t_n).$$

Applying (3.14) and (3.12b), we have

$$\|\hat{p}(\cdot,t_{n+1})\|_{\ell^1} \leq (1+4\exp(-2\pi^2 M_2^2 h/k^2))\|\hat{p}(\cdot,t_n)\|_{\ell^1}. \tag{3.19}$$

This shows that $\|\hat{p}(\cdot,t_{n+1})\|_{\ell^1} < \infty$, finishing the induction step. Combining this with the elementary bound on G, it is clear that the series on the right-hand side of (3.18) converges for all $n \geq 1$. This implies the convergence of (3.8), as an infinite series, for all $n \geq 1$.

Iterating the inequality (3.19) n times, we derive (3.17). □

The importance of Lemma 3 is that it enables us to give asymptotic conditions on h and k such that $\hat{p}$ is normalized correctly.

Lemma 4. *Suppose that g is admissible in the sense of Definition 2, and that $k = r_1 h^{\rho}$ for constants $r_1 > 0$ and $\rho > 1/2$. Assume that $N = T/h$ for some fixed $T > 0$. Then for $1 \leq n \leq N+1$,*

$$\lim_{h\to 0} k\|\hat{p}(\cdot,t_n)\|_{\ell^1} = 1. \tag{3.20}$$

Proof. Applying the hypotheses to the exponential term in (3.17) with $n = N = T/h$, we have

$$\lim_{h\to 0} \left(1+4\exp\left(-2\pi^2 M_2^2 r_1^{-2} h^{-2\rho+1}\right)\right)^{T/h} = 1. \tag{3.21}$$

For any $n \in \{0,1,\dots,N\}$, we have

$$\lim_{h\to 0} \|\hat{p}(\cdot,t_{n+1})\|_{\ell^1} / \|\hat{p}(\cdot,t_1)\|_{\ell^1} = 1. \tag{3.22}$$

Next, we combine the fact that $\hat{p}(\cdot,t_1)$ is Gaussian with (3.14) to conclude that $k\|\hat{p}(\cdot,t_1)\|_{\ell^1} \to 1$ as $k \to 0$. Then (3.20) follows immediately from (3.22). □

Lemma 5. *Suppose that f and g are admissible in the sense of Definition 2, and that $a < \min\{d, M_2^2/(2M_3M_4)\}$. Then for any $x,y \in \mathbb{R}$, there exists $A_2 > 0$ such that*

$$|G(x,y+\mathrm{i}a)| = \frac{1}{\sqrt{2\pi h|g(y+\mathrm{i}a)|^2}} \exp\left(-\frac{A_2x^2 + A_1x + A_0}{4|g(y+\mathrm{i}a)|^4 h}\right), \tag{3.23}$$

and there exists $\gamma_0 \in (0,2)$ such that

$$|G(x,y+\mathrm{i}a)| \le \frac{1}{\sqrt{2\pi h M_2^2}} \exp\left(\frac{a^2(1+hM_1)^2}{h\gamma_0 M_2^2}\right).$$

Proof. We obtain (3.23) by direct calculation of $|G(x,y+\mathrm{i}a)|$. The coefficients A_2, A_1, and A_0 are defined by

$$A_2 = g^2(y-\mathrm{i}a) + \text{c.c.} \tag{3.24a}$$

$$A_1 = -2g^2(y-\mathrm{i}a)(y+\mathrm{i}a+f(y+\mathrm{i}a)h) + \text{c.c.} \tag{3.24b}$$

$$A_0 = g^2(y-\mathrm{i}a)(y^2-a^2+f^2(y+\mathrm{i}a)h^2 + 2y\mathrm{i}a + 2(y+\mathrm{i}a)f(y+\mathrm{i}a)h) + \text{c.c.} \tag{3.24c}$$

By "c.c." we mean the complex conjugate of the preceding term. We have used the fact that because f and g are analytic on S_d, and because they are real-valued when restricted to the real axis, both f and g commute with complex conjugation. That is, $\overline{f(y+\mathrm{i}a)} = f(y-\mathrm{i}a)$ and similarly for g and g^2. The upshot is that A_2, A_1, and A_0 are all real.

Let us now prove that $A_2 > 0$. Define the function

$$\theta(y,\varepsilon) = g^2(y-\mathrm{i}\varepsilon) + g^2(y+\mathrm{i}\varepsilon),$$

for $\varepsilon \in [0,d)$. For each fixed y, by the mean-value theorem, there exists ξ such that

$$\theta(y,\varepsilon) - \theta(y,0) = \varepsilon \frac{\partial\theta}{\partial\varepsilon}(y,\xi).$$

Note that ξ may depend on ε and y. Now we use (3.12) to compute

$$\sup_{y\in\mathbb{R},\varepsilon\in(-d,d)}\left|\frac{\partial\theta}{\partial\varepsilon}\right| = 4\sup_{\substack{y\in\mathbb{R}\\ \varepsilon\in(-d,d)}}\left|\Im(g(y+\mathrm{i}\varepsilon)g'(y+\mathrm{i}\varepsilon))\right| \le 4M_3M_4. \tag{3.25}$$

Then using the previous two equations together with (3.12b), we have

$$\theta(y,\varepsilon) \ge \theta(y,0) - 4\varepsilon M_3M_4 \ge 2M_2^2 - 4\varepsilon M_3M_4. \tag{3.26}$$

The right-hand side will be positive as long as $\varepsilon < \min\{d, M_2^2/(2M_3M_4)\}$. Given the hypothesis on a in the statement of the lemma, $\theta(y,a) = A_2$ will be positive.

Because $A_2 > 0$, we can maximize the right-hand side of (3.23) as a function of x—the global maximum occurs at $x = -A_1/(2A_2)$. Then we have

$$|G(x,y+\mathrm{i}a)| \le \frac{1}{\sqrt{2\pi h M_2^2}}\exp\left(\frac{(2a+\mathrm{i}h(f(y-\mathrm{i}a)-f(y+\mathrm{i}a)))^2}{4h(g^2(y+\mathrm{i}a)+g^2(y-\mathrm{i}a))}\right)$$

We suppose that $a = bM_2^2/(2M_3M_4)$ for some $b\in(0,1)$ such that $a<d$. Then the lower bound (3.26) implies $\theta(y,a) \ge 2M_2^2(1-b)$. We define $\gamma_0 = 2(1-b)\in(0,2)$ and write

$$|G(x,y+\mathrm{i}a)| \le \frac{1}{\sqrt{2\pi h M_2^2}}\exp\left(\frac{(2a+\mathrm{i}h(f(y-\mathrm{i}a)-f(y+\mathrm{i}a)))^2}{h\gamma_0 M_2^2}\right) \tag{3.27}$$

Let Γ be the segment connecting $y-\mathrm{i}a$ to $y+\mathrm{i}a$. Note that $a<d$ implies that Γ is completely contained in the strip S_d where f is analytic. Using (3.12a), we have

$$\begin{aligned}
|2a+\mathrm{i}h(f(y-\mathrm{i}a)-f(y+\mathrm{i}a))| &\le 2|a| + h|f(y+\mathrm{i}a)-f(y-\mathrm{i}a)|\\
&\le 2|a| + h\left|\oint_\Gamma f'(z)\,dz\right|\\
&\le 2|a| + h\oint_\Gamma |f'(z)|\,|dz|\\
&\le 2|a|(1+hM_1)
\end{aligned}$$

Using this estimate in (3.27) finishes the proof. □

Lemma 6. *Suppose that f and g are admissible in the sense of Definition 2, and that $a<\min\{d, M_2^2/(2M_3M_4)\}$. Then the integrand (3.13), considered as a function of y, is a member of $B(S_a)$, i.e., $\varphi(x,\cdot,t_n)\in B(S_a)$.*

Proof. There are three conditions for membership in $B(S_a)$, which we verify in turn. First, it is simple to check that φ is analytic on S_a; this follows naturally from (3.12c) and the lower bound in (3.12b).

At time step t_1, we have $\hat{p}(y,t_1) = G(y,C)$, which is analytic. The arguments made earlier regarding the convergence of (3.18) hold equally well with x_i replaced by any x. This implies that for $n \geq 1$, $\hat{p}(y,t_{n+1})$ is analytic in y on S_d, so the integrand φ is analytic on $S_a \subset S_d$.

Next, we consider

$$\Phi(x,y,t_n) = \int_{b=-a}^{a} |\varphi(x,y+\mathrm{i}b,t_n)|\,db. \tag{3.28}$$

Since

$$\hat{p}(y+\mathrm{i}a,t_{n+1}) = k \sum_{j=-\infty}^{\infty} G(y+\mathrm{i}a,z_j)\hat{p}(z_j,t_n), \tag{3.29}$$

we have

$$\begin{aligned}
\Phi(x,y,t_{n+1}) &\leq k \sum_{j=-\infty}^{\infty} \hat{p}(z_j,t_n) \int_{b=-a}^{a} |G(y+\mathrm{i}b,z_j)||G(x,y+\mathrm{i}b)|\,db \\
&= k \sum_{j=-\infty}^{\infty} \hat{p}(z_j,t_n) G(y,z_j) \int_{b=-a}^{a} \exp\left(\frac{b^2}{2g^2(z_j)h}\right) |G(x,y+\mathrm{i}b)|\,db \\
&\leq \frac{1}{\sqrt{2\pi h M_2^2}} \int_{b=-a}^{a} \exp\left(\frac{b^2}{2M_2^2 h}\right) \exp\left(\frac{b^2(1+hM_1)^2}{h\gamma_0 M_2^2}\right) db \\
&\qquad \times k \sum_{j=-\infty}^{\infty} \hat{p}(z_j,t_n) G(y,z_j).
\end{aligned}$$

To arrive at the last line, we have applied Lemma 5 and (3.12b). There is only one remaining term on the right-hand side that depends on y. As $|y| \to \infty$, we have that $G(y,z_j) \to 0$. So, as $|y| \to \infty$, we have that $\Phi(x,y,t_{n+1}) = O(|y|^{\alpha})$ for $\alpha = 0$, satisfying (3.10).

Next, we establish a bounded, real function L_n such that for each $x \in \mathbb{R}$,

$$\begin{aligned}
\mathscr{N} := \int_{y=-\infty}^{\infty} |G(x,y+\mathrm{i}a)\hat{p}(y+\mathrm{i}a,t_n)|\,dy \\
+ \int_{y=-\infty}^{\infty} |G(x,y-\mathrm{i}a)\hat{p}(y-\mathrm{i}a,t_n)|\,dy \leq L_n(x) < \infty.
\end{aligned} \tag{3.30}$$

We need this estimate in order to apply Theorem 1. For this purpose, we seek an upper bound on $\mathscr{N}$ that does not depend essentially on the spatial discretization parameter k. Starting again from (3.29), we have

$$\begin{aligned}
&\int_{y=-\infty}^{\infty} |G(x,y+\mathrm{i}a)\hat{p}(y+\mathrm{i}a,t_{n+1})|\,dy \\
&\qquad \le k\sum_{j=-\infty}^{\infty} \hat{p}(z_j,t_n)\int_{y=-\infty}^{\infty} |G(y+\mathrm{i}a,z_j)||G(x,y+\mathrm{i}a)|\,dy \\
&\qquad = k\sum_{j=-\infty}^{\infty} \hat{p}(z_j,t_n)\int_{y=-\infty}^{\infty} \exp\left(\frac{a^2}{2g^2(z_j)h}\right) G(y,z_j)|G(x,y+\mathrm{i}a)|\,dy \\
&\qquad \le \exp\left(\frac{a^2}{2M_2^2h}\right)\left[k\sum_{j=-\infty}^{\infty} \hat{p}(z_j,t_n)\int_{y=-\infty}^{\infty} G(y,z_j)|G(x,y+\mathrm{i}a)|\,dy\right] \qquad (3.31) \\
&\qquad \le k\exp\left(\frac{a^2}{2M_2^2h}\right)\|\hat{p}(\cdot,t_n)\|_{\ell^1}\sup_z\left[\int_{y=-\infty}^{\infty} G(y,z)|G(x,y+\mathrm{i}a)|\,dy\right] \\
&\qquad \le k\exp\left(\frac{a^2}{2M_2^2h}\right)\|\hat{p}(\cdot,t_n)\|_{\ell^1}\psi(x,a), \qquad (3.32)
\end{aligned}$$

where

$$\psi(x,a) = \sup_{z\in\mathbb{R}}\left[\int_{y=-\infty}^{\infty} G(y,z)|G(x,y+\mathrm{i}a)|dy\right]. \qquad (3.33)$$

Examining (3.23), we see that the right-hand side of (3.32) is invariant under the reflection $a \mapsto (-a)$. We define the real-valued function

$$L_{n+1}(x) = 2k\exp\left(\frac{a^2}{2M_2^2h}\right)\|\hat{p}(\cdot,t_n)\|_{\ell^1}\psi(x,a),$$

and note that (3.32) implies $\mathscr{N} \le L_n(x)$, as required by (3.30). Our task now is to demonstrate that L_n is finite. By Lemma 5 and (3.6), we have

$$\begin{aligned}
\psi(x,a) &\le \sup_z\left[\frac{1}{\sqrt{2\pi hM_2^2}}\exp\left(\frac{a^2(1+hM_1)^2}{h\gamma_0M_2^2}\right)\int_{y=-\infty}^{\infty} G(y,z)\,dy\right] \\
&\le \frac{1}{\sqrt{2\pi hM_2^2}}\exp\left(\frac{a^2(1+hM_1)^2}{h\gamma_0M_2^2}\right).
\end{aligned}$$

Using this estimate in (3.32), we obtain

$$L_{n+1}(x) \le 2k\exp\left(\frac{a^2}{2M_2^2h}\right)\|\hat{p}(\cdot,t_n)\|_{\ell^1}\frac{1}{\sqrt{2\pi hM_2^2}}\exp\left(\frac{a^2(1+hM_1)^2}{h\gamma_0M_2^2}\right).$$

Note that the bound on the right-hand side does not depend on x at all. The dependence on k is confined to the terms $k\|\hat{p}(\cdot,t_n)\|$. By Lemmas 2 and 3 together with (3.14),

$$k\|\hat{p}(\cdot,t_n)\| \leq \left(1+4\exp(-2\pi^2 g^2(C)h/k^2)\right)\left(1+4\exp(-2\pi^2 M_2^2 h/k^2)\right)^{n-1} \leq 5^n < \infty$$

for all $k \geq 0$. In sum, we have shown that for fixed $h > 0$, fixed $n \geq 1$, and $a < \min\{d, M_2^2/(2M_3M_4)\}$, $L_n(x)$ is bounded uniformly in x and k. We have demonstrated that (3.30) holds. We conclude that $\varphi(x,\cdot,t_n) \in B(S_a)$. □

3.5 Convergence Theorem

Let

$$E(y,t_n) = \tilde{p}(y,t_n) - \hat{p}(y,t_n). \tag{3.34}$$

In this section, we establish conditions under which $\|E(\cdot,T)\|_1$ goes to zero at an exponential rate.

Theorem 2. *Assume that f and g are admissible in the sense of Definition 2. Assume that*

$$k = r_1 h^{\rho} \tag{3.35}$$

for constants $r_1 > 0$ and $\rho > 1/2$. Choose $a < \min\{d, M_2^2/(2M_3M_4)\}$ such that

$$a = r_2 h^{1/2} \tag{3.36}$$

for some $r_2 > 0$. For fixed $T > 0$, choose

$$h \in \left(0, \min\{T, (M_2^2/(4M_3M_4r_2))^2\}\right) \tag{3.37}$$

such that $N = T/h \in \mathbb{N}^+$. To be clear, r_1 and r_2 are constants that do not depend on h. Then

$$\|E(\cdot,T)\|_1 \leq c_\star h^{-1}\exp(-2\pi r_2 r_1^{-1} h^{1/2-\rho})(1+o(h)+o(k)) \tag{3.38}$$

where $o(h)$ and $o(k)$ stand for terms that vanish as $h \to 0$ and $k \to 0$, and $c_\star > 0$ is a constant that does not depend on h.

Proof. We begin with

$$\begin{aligned}\tilde{p}(x,t_{n+1}) &= \int_{y=-\infty}^{\infty} G(x,y)\tilde{p}(y,t_n)\,dy \\ &= \int_{y=-\infty}^{\infty} G(x,y)\hat{p}(y,t_n)\,dy + \int_{y=-\infty}^{\infty} G(x,y)E(y,t_n)\,dy.\end{aligned}$$

We now apply the trapezoidal rule to the first integral. For each x and t_n, we let $\tau(x,t_n)$ denote the quadrature error incurred, i.e.,

$$\begin{aligned}\int_{y=-\infty}^{\infty} G(x,y)\hat{p}(y,t_n)\,dy &= k\sum_{j=-\infty}^{\infty} G(x,y_j)\hat{p}(y_j,t_n) + \tau(x,t_n) \\ &= \hat{p}(x,t_{n+1}) + \tau(x,t_n). \qquad (3.39)\end{aligned}$$

We use this in the previous equation to derive

$$E(x,t_{n+1}) = \int_{y=-\infty}^{\infty} G(x,y)E(y,t_n)\,dy + \tau(x,t_n).$$

Taking absolute values, we apply the triangle inequality together with $G \geq 0$ to obtain

$$|E(x,t_{n+1})| \leq \int_{y=-\infty}^{\infty} G(x,y)|E(y,t_n)|\,dy + |\tau(x,t_n)|.$$

Integrating over x and using Fubini's theorem and (3.6), we have

$$\|E(\cdot,t_{n+1})\|_1 - \|E(\cdot,t_n)\|_1 \leq \|\tau(\cdot,t_n)\|_1. \qquad (3.40)$$

Summing both sides from $n=1$ to $n=N-1$ and using (3.9), we have

$$\|E(\cdot,T)\|_1 \leq \sum_{n=1}^{N-1} \|\tau(\cdot,t_n)\|_1. \qquad (3.41)$$

We apply Lemma 6 and Theorem 1 to produce the estimate

$$|\tau(x,t_n)| \leq \frac{\mathscr{N}}{2\sinh(\pi a/k)}\exp(-\pi a/k) \qquad (3.42)$$

where τ and $\mathscr{N}$ are defined by (3.39) and (3.30), respectively. Combining (3.31) with (3.23), we have

$$\begin{aligned}\int_{y=-\infty}^{\infty} |G(x,y+\mathrm{i}a)\hat{p}(y+\mathrm{i}a,t_{n+1})|\,dy &\leq \exp\left(\frac{a^2}{2M_2^2 h}\right) \\ &\times k\sum_{j=-\infty}^{\infty} \hat{p}(z_j,t_n)\int_{y=-\infty}^{\infty} \frac{G(y,z_j)}{\sqrt{2\pi h|g(y+\mathrm{i}a)|^2}}\exp\left(-\frac{A_2x^2+A_1x+A_0}{4|g(y+\mathrm{i}a)|^4 h}\right)dy,\end{aligned}$$

where again A_2, A_1, and A_0 are defined by (3.24). We see that the right-hand side of this inequality is invariant under $a \mapsto -a$, and so we write

$$\mathscr{N} \le 2\exp\left(\frac{a^2}{2M_2^2 h}\right) k \sum_{j=-\infty}^{\infty} \hat{p}(z_j,t_n) \\ \times \int_{y=-\infty}^{\infty} \frac{G(y,z_j)}{\sqrt{2\pi h|g(y+\mathrm{i}a)|^2}} \exp\left(-\frac{A_2x^2+A_1x+A_0}{4|g(y+\mathrm{i}a)|^4 h}\right) dy.$$

For $a < \min\{d, M_2^2/(2M_3M_4)\}$, we have shown that the coefficient A_2 is positive on S_a. This enables us to integrate both sides with respect to x:

$$\int_{x=-\infty}^{\infty} \mathscr{N}\, dx \le 2\sqrt{2}\exp\left(\frac{a^2}{2M_2^2 h}\right) k \sum_{j=-\infty}^{\infty} \hat{p}(z_j,t_n) \\ \times \int_{y=-\infty}^{\infty} \frac{G(y,z_j)|g(y+\mathrm{i}a)|}{\sqrt{g^2(y+\mathrm{i}a)+g^2(y-\mathrm{i}a)}} \exp\left(\frac{(2a+\mathrm{i}h(f(y-\mathrm{i}a)-f(y+\mathrm{i}a)))^2}{4h(g^2(y+\mathrm{i}a)+g^2(y-\mathrm{i}a))}\right) dy.$$

On the right-hand side, we have carried out the x integral first; the changing of the order of summation and integration is justified by the nonnegativity of every term. Next, we apply estimates established in the proof of Lemma 5. We obtain

$$\int_{x=-\infty}^{\infty} \mathscr{N}\, dx \le 2\sqrt{2}\exp\left(\frac{a^2}{2M_2^2 h}\right) \frac{M_3}{\gamma_0^{1/2} M_2} \exp\left(\frac{a^2(1+hM_1)^2}{h\gamma_0 M_2^2}\right) k \sum_{j=-\infty}^{\infty} \hat{p}(z_j,t_n)$$

Combining this with (3.42), we have

$$\int_{x=-\infty}^{\infty} |\tau(x,t_n)|\, dx \le \\ 4\sqrt{2}\exp\left(\frac{a^2}{2M_2^2 h}\right) \frac{M_3}{\gamma_0^{1/2} M_2} \exp\left(\frac{a^2(1+hM_1)^2}{h\gamma_0 M_2^2}\right) \exp(-2\pi a/k) k \sum_{j=-\infty}^{\infty} \hat{p}(z_j,t_n).$$

Using (3.17), we obtain

$$\|\tau(\cdot,t_n)\|_1 \le 4\sqrt{2} M_3 \gamma_0^{-1/2} M_2^{-1} \exp\left(\frac{a^2}{2M_2^2 h}\right) \exp\left(\frac{a^2(1+hM_1)^2}{h\gamma_0 M_2^2}\right) \exp(-2\pi a/k) \\ \times k \|\hat{p}(\cdot,t_1)\|_{\ell^1} (1+4\exp(-2\pi^2 M_2^2 h/k^2))^{n-1}.$$

We sum both sides from $n=1$ to $n=N-1$:

$$\sum_{n=1}^{N-1} \|\tau(\cdot,t_n)\|_1 \le \sqrt{2} M_3 \gamma_0^{-1/2} M_2^{-1} \exp\left(\frac{a^2}{2M_2^2 h}\right) \exp\left(\frac{a^2(1+hM_1)^2}{h\gamma_0 M_2^2}\right) \\ \times h^{-1} \exp(-2\pi a/k) k \|\hat{p}(\cdot,t_1)\|_{\ell^1} \left[h \sum_{n=1}^{N-1} (1+4\exp(-2\pi^2 M_2^2 h/k^2))^{n-1}\right]. \quad (3.43)$$

We now use (3.41) and hypotheses (3.35) and (3.36):

$$\|E(\cdot,T)\|_1 \leq \sqrt{2}M_3\gamma_0^{-1/2}M_2^{-1}\exp\left(\frac{r_2^2}{2M_2^2}\right)\exp\left(\frac{r_2^2(1+hM_1)^2}{\gamma_0 M_2^2}\right)Th^{-1}$$
$$\times\exp(-2\pi r_2 r_1^{-1}h^{1/2-\rho})k\|\hat{p}(\cdot,t_1)\|_{\ell^1}\left[\frac{h}{T}\sum_{n=1}^{N-1}(1+4\exp(-2\pi^2M_2^2r_1^{-2}h^{1-2\rho}))^{n-1}\right]. \tag{3.44}$$

By (3.37), we have $h \leq T$. By the definition of γ_0 in Lemma 5, we have that $\gamma_0 = 2(1-b)$ where $b = 2M_3M_4a/M_2^2 = 2M_3M_4r_2h^{1/2}/M_2^2$. Assumption (3.37) now implies that $b \leq 1/2$ and $\gamma_0^{-1} \leq 1$. We write

$$c_\star = \sqrt{2}M_3M_2^{-1}\exp\left(\frac{r_2^2}{2M_2^2}\right)\exp\left(\frac{r_2^2(1+TM_1)^2}{M_2^2}\right)T.$$

Let $\xi(h) = 4\exp(-c_1h^{-c_2})$, where c_1 and c_2 are positive constants with no dependence on h. We check that ξ satisfies the hypotheses of Lemma 1; $h^{-\gamma}\xi(h)$ has a global maximum at $h_* = (c_1c_2/\gamma)^{1/c_2}$, and so we have $\xi(h) \leq \varepsilon h^\gamma$ for $\varepsilon = h_*^{-\gamma}\xi(h_*)$, any choice of $\gamma > 1$, and all $h > 0$. With $c_1 = 2\pi^2\gamma^2$ and $c_2 = 2\rho - 1$, we apply Lemma 1 to the term in square brackets on the right-hand side of (3.44). We conclude that

$$\frac{h}{T}\sum_{n=1}^{N-1}(1+4\exp(-2\pi^2M_2^2r_1^{-2}h^{1-2\rho}))^{n-1} = 1+o(h)$$

as $h \to 0$ with $N = T/h$. By Lemma 2, $k\|\hat{p}(\cdot,t_1)\|_{\ell^1} = 1+o(k)$ as $k \to 0$. Putting everything together, we are left with (3.38). □

We are now in a position to combine our result with an earlier result from the literature [7] to establish the convergence of $\hat{p}$ to p.

Corollary 1. *In addition to all of the hypotheses of Theorem 2, suppose that there exist constants $\mathscr{F}_k, \mathscr{G}_k > 0$ such that*

$$\sup_{x\in\mathbb{R}}|f^{(k)}(x)| \leq \mathscr{F}_k$$
$$\sup_{x\in\mathbb{R}}|g^{(k)}(x)| \leq \mathscr{G}_k$$

for all $k \geq 0$. Note that for $k = 1$, the first condition is redundant with (3.12a); for $k = 0$ and $k = 1$, the second condition is redundant with (3.12b) and (3.12d). Then we have

$$\|p(\cdot,T) - \hat{p}(\cdot,T)\|_1 = O(h)$$

Proof. We have

$$\|p(\cdot,T)-\hat{p}(\cdot,T)\|_1 \leq \|p(\cdot,T)-\tilde{p}(\cdot,T)\|_1 + \|\tilde{p}(\cdot,T)-\hat{p}(\cdot,T)\|_1 \tag{3.45}$$

To handle the first term, we appeal to Corollary 2.1 from [7]. Our lower bound on g in (3.12b) corresponds to Bally and Talay's uniform ellipticity hypothesis "H1"; we may then apply Equations (27-28) from [7] to derive

$$|p(x,T)-\tilde{p}(x,T)| \leq h\mathscr{K}_1 \exp\left(-\mathscr{K}_2 x^2/T\right)$$

for constants $\mathscr{K}_1, \mathscr{K}_2 > 0$ that do not depend on h. Therefore,

$$\|p(\cdot,T)-\tilde{p}(\cdot,T)\|_1 \leq h\mathscr{K}_1 \left(\frac{\pi T}{\mathscr{K}_2}\right)^{1/2} = O(h).$$

Returning to (3.45), by Theorem 2, the second term on the right-hand side goes to zero much faster than h, finishing the proof. □

3.6 Boundary Truncation

In practice, we do not evaluate (3.8) as it involves an infinite sum. In this section, we analyze a truncated version of the algorithm:

$$\mathring{p}(x,t_{n+1}) = k\sum_{j=-M}^{M} G(x,y_j)\mathring{p}(y_j,t_n) \tag{3.46}$$

This is the actual DTQ method used in practice. As in (3.9), we take $\mathring{p}(x,t_1) = G(x,C)$ and use (3.46) starting with $n=1$. Let us denote the error due to truncation by

$$r(x,t_{n+1}) = \hat{p}(x,t_{n+1}) - \mathring{p}(x,t_{n+1}) \tag{3.47}$$

By (3.9), we have $r(x,t_1) \equiv 0$. For $n \geq 1$, we have

$$r(x,t_{n+1}) = k\left(\sum_{|j|>M} G(x,y_j)\hat{p}(y_j,t_n) + \sum_{|j|\leq M} G(x,y_j)r(y_j,t_n)\right). \tag{3.48}$$

Based on the right-hand side, we see that it will be important to estimate the tail sum $\sum_{|j|>M}\hat{p}(x_j,t_n)$. We accomplish this using a Chernoff bound. To arrive at this bound,

we construct a sequence of random variables $\{Q_n\}_{n\geq 1}$. We first define a normalization constant at time n:

$$K_n = \|\hat{p}(\cdot,t_n)\|_{\ell^1} = \sum_i \hat{p}(x_i,t_n). \tag{3.49}$$

By (3.17), we know that $K_n < \infty$ for $k > 0$ and $h > 0$. Let

$$q(x_i,t_n) = \frac{\hat{p}(x_i,t_n)}{K_n}, \tag{3.50}$$

so that $\sum_i q(x_i,t_n) = 1$. For each n, we postulate a random variable Q_n with state space $\{k\mathbb{Z}\}$ and probability mass function $q(\cdot,t_n)$. In order to apply a Chernoff bound to Q_n, we must estimate its moment generating function.

Lemma 7. *Suppose f and g are admissible in the sense of Definition 2. Suppose $k = h^\rho$ for some $\rho > 1/2$. Then there exists h_* such that for all $h \in [0,h_*)$, all $s \in \mathbb{R}$, and all n satisfying $0 \leq n \leq (N-1)$,*

$$kE[e^{sQ_{n+1}}] < \frac{3}{2}\exp\left[T\left(\frac{M_3^2 s^2}{2} + f(0)s\right)\right]\left(\frac{1}{2} + \exp(Cse^{M_1 T})\right) < \infty.$$

Proof. We begin with our estimate of the moment generating function of Q_{n+1}. The calculation proceeds in two phases. The first phase is exact; note that in what follows we use the notation $y_j = jk$, $z_j = y_j + f(y_j)h$, and $g^2 = g^2(y_j)$:

$$\begin{aligned}
E[e^{sQ_{n+1}}] &= \sum_{i=-\infty}^{\infty} e^{sx_i} q(x_i,t_{n+1}) \\
&= \frac{k}{K_{n+1}} \sum_i e^{sx_i} \sum_j \frac{1}{\sqrt{2\pi g^2 h}} \exp\left(-\frac{(x_i - z_j)^2}{2g^2 h}\right) \hat{p}(y_j,t_n) \\
&= \frac{k}{K_{n+1}} \sum_j \sum_i \frac{1}{\sqrt{2\pi g^2 h}} \exp\left(-\frac{x_i^2 - 2x_i z_j + z_j^2 - 2g^2 hsx_i}{2g^2 h}\right) \hat{p}(y_j,t_n) \\
&= \frac{1}{K_{n+1}} \sum_j \zeta_s(j) \exp\left(-\frac{z_j^2 - (z_j + g^2 hs)^2}{2g^2 h}\right) \hat{p}(y_j,t_n),
\end{aligned} \tag{3.51}$$

where

$$\zeta_s(j) = k \sum_i \frac{1}{\sqrt{2\pi g^2 h}} \exp\left(-\frac{(x_i - (z_j + g^2 hs))^2}{2g^2 h}\right).$$

It is at this point that we begin to estimate. Note that the summand is in fact a discrete Gaussian $\phi(x_i)$, as in (3.15), with $\mu = z_j + g^2(y_j)hs$ and $\sigma^2 = g^2(y_j)h$. Hence we may

apply the inequalities (3.16) and (3.12b) to write

$$\zeta_s(j) \leq 1 + 4\exp\left(\frac{-2\pi^2 g^2(y_j)h}{k^2}\right) \leq 1 + 4\exp\left(\frac{-2\pi^2 M_2^2 h}{k^2}\right). \tag{3.52}$$

Next, we turn our attention to the remaining exponential in (3.51). We use (3.12b), the mean value theorem, (3.12a), and the definition of z_j to obtain:

$$\begin{aligned}
\exp\left(-\frac{z_j^2 - (z_j + g^2 hs)^2}{2g^2 h}\right) &= \exp\left(z_j s + \frac{1}{2}g^2(y_j)hs^2\right) \\
&\leq e^{M_3^2 hs^2/2} \exp(y_j s + f(y_j)hs) \\
&\leq e^{M_3^2 hs^2/2} \exp(y_j s + f(0)hs + M_1 y_j hs) \\
&\leq e^{M_3^2 hs^2/2 + f(0)hs} \exp(y_j s(1 + M_1 h)) \qquad (3.53)
\end{aligned}$$

Now we combine (3.51), (3.52), and (3.53). The result is

$$\begin{aligned}
E[e^{sQ_{n+1}}] \leq \frac{K_n}{K_{n+1}}\left(1 + 4\exp(-2\pi^2 M_2^2 h/k^2)\right) e^{M_3^2 hs^2/2 + f(0)hs} \\
\times \frac{1}{K_n}\sum_j \exp(y_j s(1 + M_1 h))\hat{p}(y_j, t_n) \qquad (3.54)
\end{aligned}$$

We recognize the expression on the second line as the moment generating function of Q_n evaluated at $s' = s(1 + M_1 h)$. Therefore,

$$\begin{aligned}
kE[e^{sQ_{n+1}}] &\leq \frac{K_n}{K_{n+1}}\left(1 + 4\exp(-2\pi^2 M_2^2 h/k^2)\right) e^{M_3^2 hs^2/2 + f(0)hs} kE[e^{s(1+M_1 h)Q_n}] \\
&\leq \underbrace{\frac{K_1}{K_{n+1}}\left(1 + 4\exp(-2\pi^2 M_2^2 h/k^2)\right)^n}_{\zeta_1(h)} e^{T(M_3^2 s^2/2 + f(0)s)} \underbrace{kE[e^{s(1+M_1 h)^n Q_1}]}_{\zeta_2(h)}.
\end{aligned}$$

The main question now is what happens as $h \to 0$ and $N \to \infty$ such that $hN = T$. We assume that $0 \leq n \leq (N-1)$. Because $k = r_1 h^\rho$ for $\rho > 1/2$, we know by Lemma 3 that $\zeta_1(h) \to 1$ as $h \to 0$. Hence there exists h_*^1 such that $h \in [0, h_*^1)$ ensures that

$|\zeta_1(h)-1|<1/2$, i.e., $\zeta_1(h)<3/2$. Next, consider

$$\begin{aligned}\zeta_2(h) &= kE[e^{s(1+M_1h)^nQ_1}]\\ &= k\sum_{i=-\infty}^{\infty} e^{s(1+M_1h)^nx_i}\hat{p}(x_i,t_1)\\ &= k\sum_{i=-\infty}^{\infty} e^{s(1+M_1h)^nx_i}G(x_i,C)\\ &= \exp\left((C+f(C)h)s(1+M_1h)^n+\frac{hg^2(C)s^2}{2}(1+M_1h)^{2n}\right)k\sum_{i=-\infty}^{\infty}\phi(x_i),\end{aligned}$$

where $\phi(x)$ is the Gaussian density defined in (3.15) with

$$\begin{aligned}\mu &= C+f(C)h+hg^2(C)s(1+M_1h)^n\\ \sigma^2 &= hg^2(C)\end{aligned}$$

Now we apply Lemma 2 and $n\le(N-1)$ to obtain

$$\begin{aligned}\zeta_2(h)\le\exp\left((C+f(C)h)s(1+M_1h)^N+\frac{hg^2(C)s^2}{2}(1+M_1h)^{2N}\right)\\ \times(1+4\exp(-2\pi^2g^2(C)h/k^2)).\end{aligned}$$

As before, $hk^{-2}=r_1^{-2}h^{1-2\rho}\to+\infty$ as $h\to0$, and the term on the second line goes to 1 as $h\to0$. Since $\lim_{h\to0^+}(1+M_1h)^N=e^{M_1T}$, we have

$$\lim_{h\to0^+}\zeta_2(h)\le\exp\left(Cse^{M_1T}\right).$$

Thus there exists h_*^2 such that $h\in[0,h_*^2)$ implies

$$\left|\zeta_2(h)-\exp\left(Cse^{M_1T}\right)\right|\le\frac{1}{2}.$$

Taking $h_*=\min\{h_*^1,h_*^2\}$ finishes the proof. □

We can now give conditions under which r, defined in (3.47), converges to zero.

Lemma 8. *Suppose f and g are admissible in the sense of Definition 2. Suppose $k=h^\rho$ for $\rho>1/2$. For $\varepsilon\ge1$, let*

$$M=\lceil(\varepsilon+\rho+1)(-\log h)/k\rceil. \tag{3.55}$$

Let h_* be defined as in Lemma 7. Then for $h<h_*$, we have $k\sum_{|i|\le M}|r(x_i,T)|=O(h)$.

Proof. We start with

$$|r(x_i,t_{n+1})| \le k \sum_{|j|>M} G(x_i,y_j)\hat{p}(y_j,t_n) + k \sum_{|j|\le M} G(x_i,y_j)|r(y_j,t_n)|.$$

Summing over i, we obtain

$$\sum_{|i|\le M} |r(x_i,t_{n+1})| \le k \sum_{|j|>M} \sum_{|i|\le M} G(x_i,y_j)\hat{p}(y_j,t_n) + k \sum_{|j|\le M} \sum_{|i|\le M} G(x_i,y_j)|r(y_j,t_n)|.$$

Using (3.14) together with (3.12b), we have

$$\sum_{|i|\le M} |r(x_i,t_{n+1})| \le (1+4\exp(-2\pi^2 M_2^2 h/k^2)) \sum_{|j|>M} \hat{p}(y_j,t_n) + (1+4\exp(-2\pi^2 M_2^2 h/k^2)) \sum_{|j|\le M} |r(y_j,t_n)|. \quad (3.56)$$

This is of the form

$$r_{n+1} \le \alpha\pi_n + \alpha r_n. \quad (3.57)$$

We derive from this the sequence of inequalities $\alpha r_n \le \alpha^2\pi_{n-1} + \alpha^2 r_{n-1}, \cdots, \alpha^{n-1} r_2 \le \alpha^n \pi_1 + \alpha^n r_1$. Summing these together with (3.57), we derive $r_{n+1} \le \sum_{i=1}^{n} \alpha^i \pi_{n-i+1} + \alpha^n r_1$. Applying this to (3.56) and using $r(\cdot,t_1) \equiv 0$, we have

$$\sum_{|i|\le M} |r(x_i,t_{n+1})| \le \sum_{i=1}^{n} (1+4\exp(-2\pi^2 M_2^2 h/k^2))^i \sum_{|j|>M} \hat{p}(y_j,t_{n-i+1}). \quad (3.58)$$

Now we use (3.50) and the Chernoff bound to derive:

$$\begin{aligned}\sum_{|j|>M} \hat{p}(y_j,t_{n-i+1}) &= K_{n-i+1} \sum_{|j|>M} q(y_j,t_{n-i+1}) \\ &\le K_{n-i+1} \left[P(Q_{n-i+1} \ge y_M) + P(Q_{n-i+1} \le -y_M)\right] \\ &\le K_{n-i+1} e^{-sy_M} \left(E[e^{sQ_{n-i+1}}] + E[e^{-sQ_{n-i+1}}]\right)\end{aligned}$$

We apply Lemma 7 to obtain

$$k \sum_{|j|>M} \hat{p}(y_j,t_{n-i+1}) \le \frac{3}{2} K_{n-i+1} e^{-sy_M} \exp\left[T\left(\frac{M_3^2 s^2}{2} + f(0)s\right)\right] \left(1+2\cosh(Cse^{M_1 T})\right) \quad (3.59)$$

Applying this result to (3.58), we have

$$k\sum_{|i|\le M}|r(x_i,t_{n+1})| \le \frac{3}{2}e^{-sy_M}\exp\left[T\left(\frac{M_3^2s^2}{2}+f(0)s\right)\right]\left(1+2\cosh(Cse^{M_1T})\right) \times\sum_{i=1}^{n}(1+4\exp(-2\pi^2M_2^2h/k^2))^iK_{n-i+1}.$$

By (3.49) and (3.17), we have

$$K_{n-i+1}\le\|\hat{p}(\cdot,t_1)\|_{\ell^1}(1+4\exp(-2\pi^2M_2^2h/k^2))^{n-i}$$

Using this and $n\le N=T/h$,

$$k\sum_{|i|\le M}|r(x_i,t_{n+1})| \le \frac{3}{2}e^{-sy_M}\exp\left[T\left(\frac{M_3^2s^2}{2}+f(0)s\right)\right]\left(1+2\cosh(Cse^{M_1T})\right) \times\|\hat{p}(\cdot,t_1)\|_{\ell^1}\frac{T}{h}(1+4\exp(-2\pi^2M_2^2h/k^2))^{T/h}. \quad (3.60)$$

Let $s=1$. Note that $\lim_{h\to0}(1+4\exp(-2\pi^2M_2^2h/k^2))^{T/h}=1$ and $\lim_{k\to0}k\|\hat{p}(\cdot,t_1)\|_{\ell^1}=1$. Thanks to (3.55), we know that $y_M\ge(\varepsilon+\rho+1)(-\log h)$. Putting things together, the right-hand side of (3.60) behaves like $h^{\varepsilon+\rho+1}k^{-1}h^{-1}=h^{\varepsilon}=O(h)$ as desired. □

So long as M remains a positive integer, we can add/subtract a constant from (3.55) and still prove Lemma 8. What is important is how M scales as a function of h; the logarithmic rate given in (3.55) is the rate at which we have to push M to $+\infty$ so that we obtain $O(h)$ convergence. If we push M to $+\infty$ at a faster rate, e.g., by replacing $(-\log h)$ with h^{-1}, then r will converge at a rate that is exponential in h.

Thus far we have considered convergence of r in a truncated and scaled version of the ℓ^1 norm. Convergence in L^1 is an easy consequence.

Lemma 9. *Suppose f and g are admissible in the sense of Definition 2. Suppose $k=h^\rho$ for $\rho>1/2$. For $\varepsilon\ge1$, let M be defined as in (3.55). Let h_* be defined as in Lemma 7. Then for $h<h_*$, we have $\|r(\cdot,T)\|_1=O(h)$.*

Proof. Note that

$$|r(x,T)|\le k\sum_{|j|>M}G(x,y_j)\hat{p}(y_j,t_{N-1})+k\sum_{|j|\le M}G(x,y_j)|r(y_j,t_{N-1})|.$$

This is similar to what we wrote above, except that the discrete variable x_i has been replaced by the continuous variable x. We now integrate both sides with respect to x to obtain

$$\|r(\cdot,T)\|_1 \le k \sum_{|j|>M} \hat{p}(y_j,t_{N-1}) + k \sum_{|j|\le M} |r(y_j,t_{N-1})|.$$

The second term is $O(h)$ by Lemma 8. For the first term, we use (3.59) to write

$$k \sum_{|j|>M} \hat{p}(y_j,t_{N-1}) \le \frac{3}{2} K_{N-1} e^{-y_M} \exp\left[T\left(\frac{M_3^2}{2} + f(0)\right)\right]\left(1+2\cosh(Ce^{M_1 T})\right). \quad (3.61)$$

Since $\lim_{k\to 0} kK_{N-1} = 1$ and $e^{-y_M} = O(h^{\varepsilon+\rho+1})$, the right-hand side of (3.61) behaves like $h^{\varepsilon+1} = O(h^2)$. □

It is now immediately clear that, under certain conditions, we have established $O(h)$ convergence of $\mathring{p}$ to the true density p in the L^1 norm.

Corollary 2. *Suppose that all of the hypotheses of Corollary 1 and Lemma 9 are satisfied. Then, combining these results, we have* $\|p(\cdot,T) - \mathring{p}(\cdot,T)\|_1 = O(h)$.

3.7 Numerical Experiments

In this section, we use R/C++ implementations of the DTQ method to study its empirical convergence behavior, and also to compare against a numerical solver for (3.3), the Fokker-Planck or Kolmogorov equation. All codes described in this section, together with instructions on how to reproduce Figures 3.1 and 3.2, are available at the following URL:

```
https://github.com/hbhat4000/sdeinference/tree/master/
DTQpaper
```

We caution the reader that, in the present work, we do not deal with all important implementation issues. Here we are primarily concerned with demonstrating properties of the DTQ method. This can be done quite well even with the assumptions on the initial condition and domain sizes given below. Relaxing these assumptions poses no conceptual difficulties, but may require changes to technical details in the codes linked above.

3.7.1 Convergence

First, we compare empirical and theoretical convergence behavior. We verify that under the conditions given by Theorem 2, we do observe convergence in practice. We also show numerical evidence that such convergence takes place when one or more of the hypotheses do not hold.

All of the SDE we consider are equations for a scalar unknown X_t. We describe here the way in which we conduct numerical tests for each SDE. We begin with the initial condition $X_0 = 0$ and solve forward in time until $T = 1$. That is, we apply the DTQ method to compute $\mathring{p}(x,1)$. We use the following values of the temporal step h:

$$\{0.5, 0.2, 0.1, 0.05, 0.02, 0.01, 0.005, 0.002, 0.001\}. \tag{3.62}$$

For $h \geq 0.01$, we find that an implementation of the DTQ method written completely in R is able to run in a reasonable amount of time. For $h = 0.005$ and below, we use an implementation where computationally intensive parts of the code are written in C++; this code is glued to our R code using the Rcpp and RcppArmadillo packages [24, 23, 25, 54].

The remaining algorithm parameters are set in the following way:

$$k = h^{3/4} \tag{3.63a}$$

$$\begin{cases} \text{Examples 1,2,4,5,6} & M = \lceil \pi/k^2 \rceil \\ \text{Example 3} & M = \lceil \pi/(2k) - 2 \rceil. \end{cases} \tag{3.63b}$$

$$x_j = jk, \text{ for } -M \leq j \leq M. \tag{3.63c}$$

For each value of h, we compare $\mathring{p}(x,T)$ computed using the DTQ method against the exact solution $p(x,T)$. Let $F(y,T) = \int_{x=-\infty}^{x=y} p(x,T)\,dx$ denote the cumulative distribution function associated with the density p. Each comparison is carried out using the following three norms:

$$\|p(\cdot,T) - \mathring{p}(\cdot,T)\|_1 \approx k \sum_{j=-M}^{j=M} |p(jk,T) - \mathring{p}(jk,T)| \tag{3.64a}$$

$$\|p(\cdot,T) - \mathring{p}(\cdot,T)\|_\infty \approx \sup_{|j| \leq M} |p(jk,T) - \mathring{p}(jk,T)| \tag{3.64b}$$

$$\|F(\cdot,T) - \mathring{F}(\cdot,T)\|_\infty \approx \sup_{|j| \leq M} |F(jk,T) - \mathring{F}(jk,T)| \tag{3.64c}$$

For our tests, we consider six SDE, all for a scalar unknown X_t:

$$\text{Example 1:} \begin{cases} dX_t = -X_t dt + dW_t \\ p(x,t) = \dfrac{\exp(-x^2/(1-\exp(-2t)))}{\sqrt{(\pi(1-\exp(-2t)))}} \end{cases} \tag{3.65a}$$

$$\text{Example 2:} \begin{cases} dX_t = -\frac{1}{2}\tanh X_t \operatorname{sech}^2 X_t dt + \operatorname{sech} X_t dW_t \\ p(x,t) = (2\pi t)^{-1/2}(\cosh x)\exp(-\sinh^2 x/(2t)) \end{cases} \tag{3.65b}$$

$$\text{Example 3:} \begin{cases} dX_t = -(\sin X_t \cos^3 X_t)dt + (\cos^2 X_t)dW_t \\ p(x,t) = (2\pi t)^{-1/2}(\sec^2 x)\exp(-\tan^2 x/(2t)) \end{cases} \tag{3.65c}$$

$$\text{Example 4:} \begin{cases} dX_t = \left(\frac{1}{2}X_t + \sqrt{1+X_t^2}\right)dt + \sqrt{1+X_t^2}dW_t \\ p(x,t) = (2\pi(1+x^2))^{-1/2}\exp(-(\sinh^{-1}x - t)^2/2) \end{cases} \tag{3.65d}$$

$$\text{Example 5:} \begin{cases} dX_t = \frac{1}{2}X_t dt + \sqrt{1+X_t^2}dW_t \\ p(x,t) = (2\pi t(1+x^2))^{-1/2}\exp(-(\sinh^{-1}x)^2/(2t)) \end{cases} \tag{3.65e}$$

$$\text{Example 6:} \begin{cases} dX_t = \left(-\sqrt{1+X_t^2}\sinh^{-1}X_t + \frac{1}{2}X_t\right)dt + \sqrt{1+X_t^2}dW_t \\ p(x,t) = \dfrac{\exp(-(\sinh^{-1}x)^2/(1-\exp(-2t)))}{\sqrt{(\pi(1-\exp(-2t))(1+x^2))}} \end{cases} \tag{3.65f}$$

Note that for each example, we have supplied an exact solution in the form of a probability density function $p(x,t)$. For each example, we compare the DTQ density with $p(x,T=1)$.

Figure 3.1 shows the convergence results for all six examples. The overall impression we gain from the plots is that the practical L^1 error between the DTQ and exact density functions scales like h. As we now explain, this first-order convergence is displayed under a variety of conditions.

Example 1 features drift and diffusion coefficients that clearly satisfy the hypotheses of our convergence theory. In this case, the computational results confirm the theory.

In Example 2, the drift and diffusion coefficients satisfy all but one of the hypotheses. Specifically, because $\operatorname{sech} x \to 0$ as $|x| \to \infty$, the diffusion coefficient is not bounded away from zero. However, as a matter of numerical practice, on any truncated domain of the form (3.63), the diffusion coefficient never equals zero. We can say, then, that on

the computational domain, the diffusion coefficient does have a global lower bound that is greater than zero. The computational results display first-order convergence.

Example 3 is similar to Example 2 in that all but one of the hypotheses are satisfied. Again, it is the diffusion coefficient $\cos^2 y$ that is not bounded away from zero. However, either an analysis of the original SDE or inspection of the exact solution reveals that the density will only be supported on the interval $(-\pi/2, \pi/2)$. For this SDE, we set $M = \lceil \pi/(2k) - 2 \rceil$ as in (3.63b), retaining (3.63a) and (3.63c). This way, the spatial grid covers the interior of $(-\pi/2, \pi/2)$ and the diffusion coefficient never reaches zero. Again, the computational results show that the L^1 error scales like h.

Moving to Examples 4 and 5, we now have instances where the diffusion coefficient is bounded from below by 1 but is unbounded above. All other hypotheses of our convergence theory are satisfied. The empirical convergence rates for both examples match what we expect from theory.

Reexamining the situation with slightly more depth, what we find from our proofs is that (3.25) is the only place where the upper bound on the diffusion coefficient is used. However, for the particular case of the diffusion coefficient $g(x) = (1+x^2)^{1/2}$ used in Examples 4 and 5, we have that

$$\left|\Im\left(g(y+\mathrm{i}\varepsilon)g'(y+\mathrm{i}\varepsilon)\right)\right| = \left|\Im(y+\mathrm{i}\varepsilon)\right| \leq d,$$

meaning that we can substitute d for $M_3 M_4$ and the convergence proof follows. This is an example of how, for specific SDE that do not satisfy the hypotheses of the general theorem, we may yet be able to prove convergence of the DTQ method.

Finally, we come to Example 6. Now we have that derivative of the drift coefficient is unbounded *and* that the diffusion coefficient is unbounded above. Still, the results in the convergence plot agree with the overall first-order convergence rate implied by theory.

For the SDE in Example 6, even if we are able to patch our proof to prove that $\hat{p}$ converges to $\tilde{p}$, we can no longer apply the result of Bally and Talay [7] to guarantee convergence of $\tilde{p}$ to p. Overall, we take the numerical results for Example 6 as evidence that $\tilde{p}$ must converge to p under more general conditions than have been established in the literature.

3.7.2 Comparison with Fokker-Planck

Now we turn to a comparison of the DTQ method with a classical approach, that of numerically solving the Fokker-Planck or Kolmogorov PDE (3.3). In what follows, we use subscripts to denote partial derivatives, so that (3.3) is written

$$p_t + (f(x)p(x,t))_x = \frac{1}{2}\left(g^2(x)p(x,t)\right)_{xx}. \tag{3.66}$$

To solve this equation, we employ a standard finite difference method. To resolve the singular initial condition $p(x,0) = \delta(x)$, we use a standard subtraction idea: we set $p = u + v$, where u solves

$$u_t = \frac{1}{2}\kappa u_{xx} \tag{3.67a}$$

$$u(x,0) = \delta(x), \tag{3.67b}$$

while v solves

$$v_t + (f(x)v(x,t))_x = \frac{1}{2}\left(g^2(x)v(x,t)\right)_{xx} + \underbrace{\frac{1}{2}\left[\left(g^2(x)-\kappa\right)u(x,t)\right]_{xx} - \left[f(x)u(x,t)\right]_x}_{F(x,t)} \tag{3.68a}$$

$$v(x,0) = 0. \tag{3.68b}$$

The point is that (3.67) can be solved analytically, i.e., for $t > 0$,

$$u(x,t) = \frac{1}{\sqrt{2\pi\kappa t}}\exp\left(-\frac{x^2}{2\kappa t}\right). \tag{3.69}$$

Here $\kappa > 0$ is a parameter that we are free to set. In our own tests, we use $\kappa = 1$. Since (3.69) is known, we substitute it into the final two terms on the right-hand side of (3.68a)—this yields a known forcing term $F(x,t)$. We then employ the following numerical scheme to solve (3.68) for $v(x,t)$:

- We discretize $v(x,t)$ on fixed spatial and temporal grids with respective spacings k and h. Let V_j^n denote our numerical approximation to $v(jk,nh)$. Here $0 \le n \le N$ with $Nh = T > 0$, the final time. We also have that $-M \le j \le M$. Implicitly, we assume that $v(x,t) = 0$ for $|x| > Mk$.

- We use a first-order approximation to v_t: $v_t(x,t) \approx (V_j^{n+1} - V_j^n)/h$.
- We treat the drift term explicitly:

$$(f(x)v(x,t))_x \approx \left(f((j+1)k)V_{j+1}^n - f((j-1)k)V_{j-1}^n\right)/(2k).$$

- We treat the diffusion term implicitly:

$$\frac{1}{2}\left(g^2(x)v(x,t)\right)_{xx} \approx \frac{1}{2k^2}\left(g^2((j-1)k)V_{j-1}^{n+1} - 2g^2(jk)V_j^{n+1} + g^2((j+1)k)V_{j+1}^{n+1}\right).$$

Let $\mathbf{V}^n$ be a vector of length $2M+1$ whose j-th entry is V_j^n. Then, combining approximations, we obtain the matrix-vector system

$$A\mathbf{V}^{n+1} = B\mathbf{V}^n + \mathbf{F}^n \tag{3.70}$$

with tridiagonal matrices A and B given by

$$A = \begin{bmatrix} 1+\frac{h}{k^2}g_{-M}^2 & -\frac{h}{2k^2}g_{-M+1}^2 & & & & & \\ -\frac{h}{2k^2}g_{-M}^2 & 1+\frac{h}{k^2}g_{-M+1}^2 & -\frac{h}{2k^2}g_{-M+2} & & & & \\ & -\frac{h}{2k^2}g_{-M+1}^2 & 1+\frac{h}{k^2}g_{-M+2}^2 & -\frac{h}{2k^2}g_{-M+3}^2 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -\frac{h}{2k^2}g_{M-2}^2 & 1+\frac{h}{k^2}g_{M-1}^2 & -\frac{h}{2k^2}g_M^2 & \\ & & & & -\frac{h}{2k^2}g_{M-1}^2 & 1+\frac{h}{k^2}g_M^2 \end{bmatrix} \tag{3.71}$$

and

$$B = \begin{bmatrix} 1 & -\frac{h}{2k}f_{-M+1} & & & & & \\ \frac{h}{2k}f_{-M} & 1 & -\frac{h}{2k}f_{-M+2} & & & & \\ & \frac{h}{2k}f_{-M+1} & 1 & -\frac{h}{2k}f_{-M+3} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \frac{h}{2k}f_{M-2} & 1 & -\frac{h}{2k}f_M & \\ & & & & \frac{h}{2k}f_{M-1} & 1 & \end{bmatrix}. \tag{3.72}$$

We also define $\mathbf{F}^n$ in (3.70) by discretizing $F(x,t)$ in (3.68a). Specifically, for $-M \le j \le M$, we define the j-th component of $\mathbf{F}^n$ by

$$\begin{aligned} F_j^n = \frac{h}{2k^2}\left[g^2((j-1)k)u((j-1)k,nh) - 2g^2(jk)u(jk,nh) + g^2((j+1)k)u((j+1)k,nh)\right] \\ - \frac{h}{2k}\left[f((j+1)k)u((j+1)k,nh) - f((j-1)k)u((j-1)k,nh)\right]. \end{aligned} \tag{3.73}$$

To solve for $\mathbf{V}^{n+1}$ given $\mathbf{V}^n$, we rewrite (3.70) as

$$\mathbf{V}^{n+1} = A^{-1}B\mathbf{V}^n + A^{-1}\mathbf{F}^n. \tag{3.74}$$

Let $\mathbf{U}^N$ denote the vector obtained by evaluating $u(jk,T)$ for $-M \le j \le M$. Let $p_{\mathrm{FP}}(\mathbf{x},T)$ denote the vector whose j-th component is $p_{\mathrm{FP}}(x_j,T)$, the approximation of $p(x_j,T)$ obtained by solving the Fokker-Planck equation numerically. With these definitions, our algorithm for computing p_{FP} is easily stated: we start with $\mathbf{V}^0 = \mathbf{0}$, iterate (3.74) N times to compute $\mathbf{V}^N$, and then compute

$$p_{\mathrm{FP}}(\mathbf{x},T) = \mathbf{U}^N + \mathbf{V}^N.$$

Note that in our implementation of the Fokker-Planck method, the matrices A and B defined by (3.71) and (3.72) are implemented as sparse tridiagonal matrices. When we use (3.74) to solve for $\mathbf{V}^{n+1}$, we use sparse numerical linear algebra to compute both $A^{-1}B$ and $A^{-1}\mathbf{F}^n$. In particular, $A^{-1}B$ is precomputed before we loop from $n = 0$ to $n = N-1$.

We are now in a position to compare the DTQ and Fokker-Planck methods. For this comparison, we exclusively use the drift and diffusion functions from Example 1 in (3.65). As described above, among the examples in (3.65), Example 1 is the only one that satisfies all of the hypotheses of our DTQ convergence theory.

As mentioned in Section 3.6, when we implement the DTQ method in practice, we start with (3.46)—with x discretized on the same spatial grid as y, i.e.,

$$\mathring{p}(x_i,t_{n+1}) = k\sum_{j=-M}^{M} G(x_i,y_j)\mathring{p}(y_j,t_n) \tag{3.75}$$

For fixed n, as j varies from $-M$ to M, the elements $\mathring{p}(y_j,t_n)$ form a $(2M+1)$-dimensional vector that we denote $\mathbf{p}^n$. With this notation, (3.75) can be written

$$\mathbf{p}^{n+1} = \mathscr{A}\mathbf{p}^n, \tag{3.76}$$

where $\mathscr{A}$ is the $(2M+1)\times(2M+1)$ matrix whose (i,j)-th element is $kG(x_i,y_j)$. In our experience, *the most computationally expensive part of the DTQ method is the assembly of* $\mathscr{A}$. For the tests presented in this subsection, we have implemented three different methods to compute $\mathscr{A}$:

1. **DTQ-Naïve**. Here we assemble $\mathscr{A}$ using dense matrix methods in R. The main advantage of this approach is ease of implementation; the code to compute $\mathscr{A}$ is only 4 lines long. Incidentally, the convergence tests in the first part of this section use the DTQ-Naïve method for $h \geq 0.01$.

2. **DTQ-CPP**. Implicitly, the DTQ-Naïve method forces R to loop over the entries of $\mathscr{A}$ serially. In the DTQ-CPP method, we use Rcpp together with OpenMP directives to compute and fill in the entries of $\mathscr{A}$ in parallel. In practice, we run this code on a machine with 12 cores, setting the number of OpenMP threads to 12.

3. **DTQ-Sparse**. Here we take advantage of the structure of $\mathscr{A}$. Specifically, we have

$$\mathscr{A}_{ij} = kG(x_i, y_j) = \frac{k}{\sqrt{2\pi g^2(y_j)h}} \exp\left(-\frac{(x_i - y_j - f(y_j)h)^2}{2g^2(y_j)h}\right).$$

   Let us set $i = j + i'$. Then we have

$$\mathscr{A}_{j+i',j} = \frac{k}{\sqrt{2\pi g^2(y_j)h}} \exp\left(-\frac{(i'k - f(y_j)h)^2}{2g^2(y_j)h}\right). \tag{3.77}$$

   We think of i' as indexing the sub-/super-diagonals of $\mathscr{A}$. For each fixed $i' = 0, 1, 2, \ldots$ we evaluate (3.77) over all j to obtain the i'-th subdiagonal of $\mathscr{A}$. For h small, as i' increases, we observe that the entire subdiagonal decays rapidly. In our implementation, we compute subdiagonals until the 1-norm of the subdiagonal drops below 2.2×10^{-16} (machine precision in R) multiplied by the 1-norm of the main $i' = 0$ diagonal of $\mathscr{A}$. We then compute the same number of superdiagonals as subdiagonals. The final $\mathscr{A}$ matrix is assembled as a sparse matrix using the CRAN Matrix package [9].

Given the tridiagonal structure of both A and B in the Fokker-Planck method, we do not believe any reasonable modern implementation would use dense matrices. Similarly, while DTQ-Naïve requires minimal programming effort, a reasonable implementation would look much more like DTQ-CPP or DTQ-Sparse. None of the DTQ methods require more programming effort to implement than the Fokker-Planck method.

Results for $O(h^{3/4})$ Domain Scaling. For each h in (3.62) that satisfies $h \geq 0.01$, we use all three DTQ methods and the Fokker-Planck method to generate numerical approximations of the density function at the final time $T = 1$. For our first set of comparisons, parameters such as k and M are set via (3.63). In particular, the computational domain is $[-y_M, y_M]$ where $y_M = Mk \propto h^{-3/4}$. We compute the L^1 errors between each numerical solution and the exact solution $p(x,T)$. We also record the wall clock time (in seconds) required to compute the solution using each method. Each measurement is repeated 100 times; we report average results.

In the left panel of Figure 3.2, we have plotted (on log-scaled axes) wall clock time as a function of L^1 error for each of the four methods. We see that if one can tolerate a relatively large L^1 error, then the fastest method is the DTQ-Naïve method (green); for L^1 errors less than 0.03, the fastest method is the DTQ-Sparse method (purple). The Fokker-Planck method is often the slowest of the four methods. For an error of 0.003, the DTQ-Sparse method is approximately 100 times faster than the Fokker-Planck method.

Results for $O(\log h^{-1})$ Domain Scaling. For our second set of comparisons, we have changed the way that y_M (effectively, the size of the computational domain) scales with h. We retain $k = h^{3/4}$ but now set $y_M = (2+3/4)(-\log h) \propto (-\log h)$ in accordance with (3.55). The spatial grid, for all four methods, is now given by $x_j = -y_M + (j+M)k$ for $-M \leq j \leq M$ with $M = \lfloor y_M/k \rfloor$. In all other respects, we make no changes and rerun the test described above for all four methods.

In the right panel of Figure 3.2, we have plotted (on log-scaled axes) wall clock time as a function of L^1 error for each of the four methods. Once again, we find that the DTQ-Naïve and DTQ-Sparse methods are the fastest for, respectively, large and small error values. For an error of 0.003, the DTQ-Sparse method is approximately $10^{3/4} \approx 5.62$ times faster than the Fokker-Planck method.

3.8 Conclusion and Future Directions

In this chapter, we have established fundamental properties of the DTQ method, including theoretical and empirical convergence results. Let us make three concluding

remarks regarding our results. First, until now we have not mentioned that the DTQ method features two properties that are not always easy to establish for numerical methods for the Fokker-Planck equation (3.3): (i) the DTQ method automatically preserves the nonnegativity of the computed density $\hat{p}$, and (ii) the DTQ density $\hat{p}$ has a normalization constant that can be estimated for finite $h,k>0$. In practice, we find that $\mathring{p}$ is very close to being correctly normalized.

Second, $p(x,T)$ and $\tilde{p}(x,t_N)$ correspond to, respectively, the random variables X_T and x_N. Convergence in L^1 of $\tilde{p}$ to p is equivalent to convergence in total variation of x_N to X_T. Note that

$$\int_{x=-\infty}^{\infty} \hat{p}(x,t_{n+1})\,dx = k\sum_{j=-\infty}^{\infty} \hat{p}(y_j,t_n) = kK_n, \tag{3.78}$$

implying that $\hat{q}(x,t_{n+1}) = \hat{p}(x,t_{n+1})/(kK_n)$ is the density function of a continuous random variable y_n. An easy consequence of our results is that $\hat{q}$ converges to $\tilde{p}$ in L^1, implying convergence of y_N to x_N in total variation.

Third, if we trace back the crux of our convergence proof, a key step is estimating the L^1 error of τ starting from the trapezoidal rule error estimate (3.42). To do this, it was essential that we have an estimate of $\mathscr{N}$ that is an L^1 function of x. It was to obtain such an estimate that we put our efforts into Lemma 5. We have tried to replicate this analysis using more conventional error estimates for the trapezoidal rule—estimates that require less regularity of the integrand than we have assumed. Thus far, these other attempts have failed because they do not yield an upper bound on τ that is itself an L^1 function of x. The approach in the present work is the only one that we have gotten to work.

Given the current framework of the DTQ method, the present work motivates following two questions.

1. When we derived the DTQ method, we used three approximations: (i) an Euler-Maruyama approximation of the original SDE, (ii) a trapezoidal quadrature rule, and (iii) a finite-dimensionalization of $\tilde{p}$ that consists of sampling the function on a truncated grid. The first question to ask is: what happens to the DTQ method if we improve upon these initial approximations?

Regarding (ii), we can say that we have written a test code in which we use Gauss-Hermite quadrature instead of the trapezoidal rule. This does not yield better convergence. Given the exponential convergence of $\hat{p}$ to $\tilde{p}$ established here, this should not be a surprise.

Regarding (iii), rather than sampling the function $\tilde{p}(x,t_n)$ on a discrete grid, we could have instead chosen to represent $\tilde{p}(x,t_n)$ as a linear combination of functions—for instance, a linear combination of Gaussian densities, where each density is centered at a grid point x_j. In a collocation scheme, we would then insert these approximations of $\tilde{p}$ into (3.7) and enforce equality at a finite number of points. We have tried this as well in a test code. While such a scheme does not yield better numerical behavior, it may be easier to analyze.

Approximation (i) is the one that would most easily yield major improvements. In the DTQ derivation, we can easily replace the Euler-Maruyama method with a higher-order method. The only change is to then replace the Gaussian kernel G with a different conditional density function. With this new G, the evolution equation (3.46) for $\mathring{p}$ remains the same. Preliminary results with the weak trapezoidal method [3] indicate that, in this way, we can obtain a version of the DTQ method that features $O(h^2)$ convergence of $\mathring{p}$ to p.

2. Can we patch the DTQ method to handle diffusion functions g that equal zero at, say, a finite number of discrete points in the computational domain? We believe there should be some way of doing this by subtracting out singularities of G inside the Chapman-Kolmogorov equation (3.8).

In addition, here we have compared DTQ method against only the finite difference numerical solver for (3.3), the Fokker-Planck equation. For an extensive comparison of the DTQ method against numerical methods for the solution of (3.3), we have to consider methods such as finite difference, finite element, meshless, and Hermite spectral methods [48, 51, 16, 41]. In the following chapters, we show how the DTQ method can be used to efficiently solve the parameter inference problem.

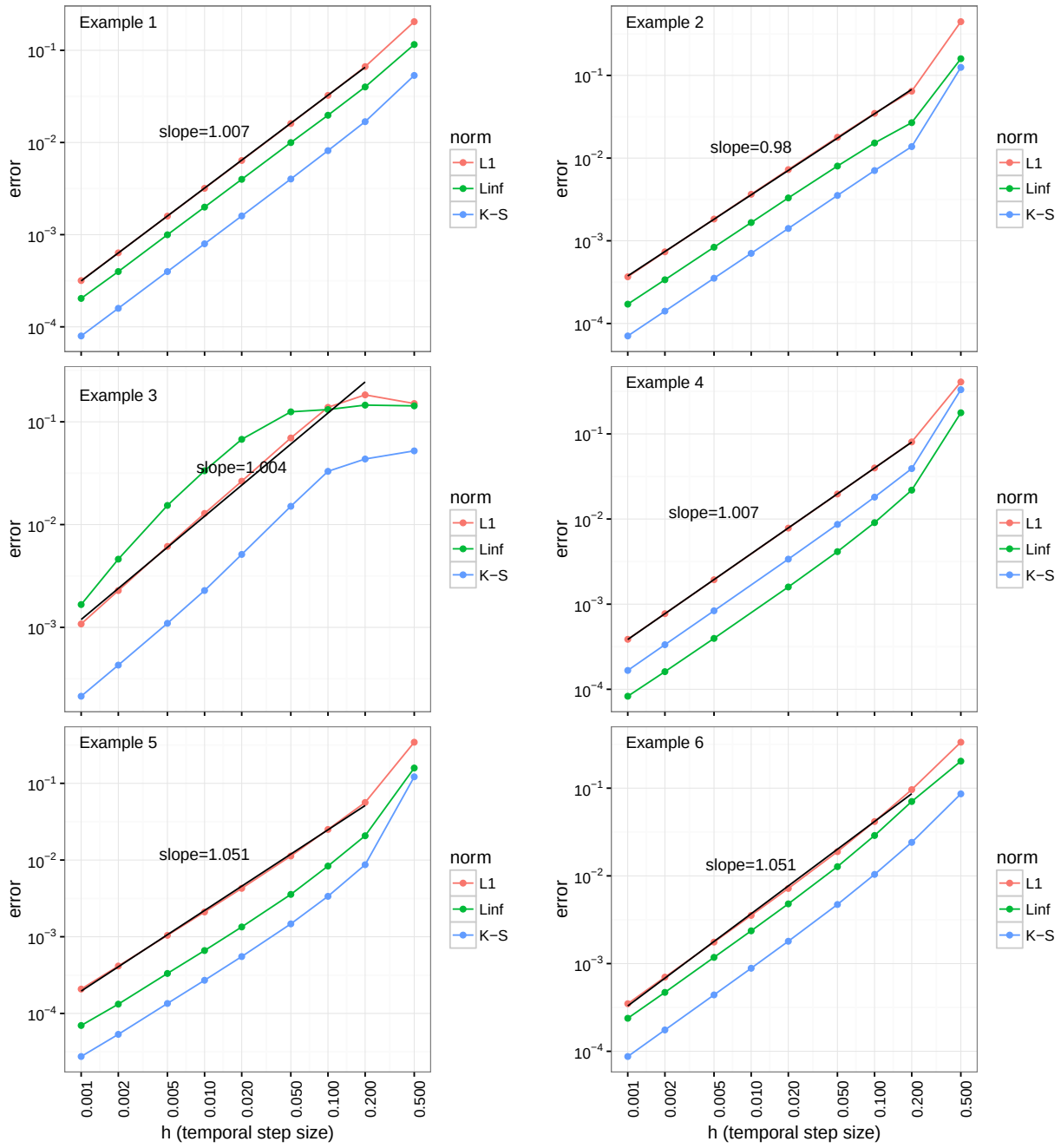

Figure 3.1: For each of the six examples in (3.65), we test the DTQ method's convergence. For each example, we plot errors between DTQ and exact solutions on log-scaled axes as a function of h, the temporal step size; all other parameters are given by (3.63). We compute errors in each of the three norms given by (3.64). The horizontal axes (labels and tick mark locations) are the same for all plots and correspond to the h values in (3.62). Least-squares fits to the L^1 error data are indicated by black lines and corresponding slope values. For all examples, we observe first-order convergence, consistent with our $O(h)$ theoretical result.

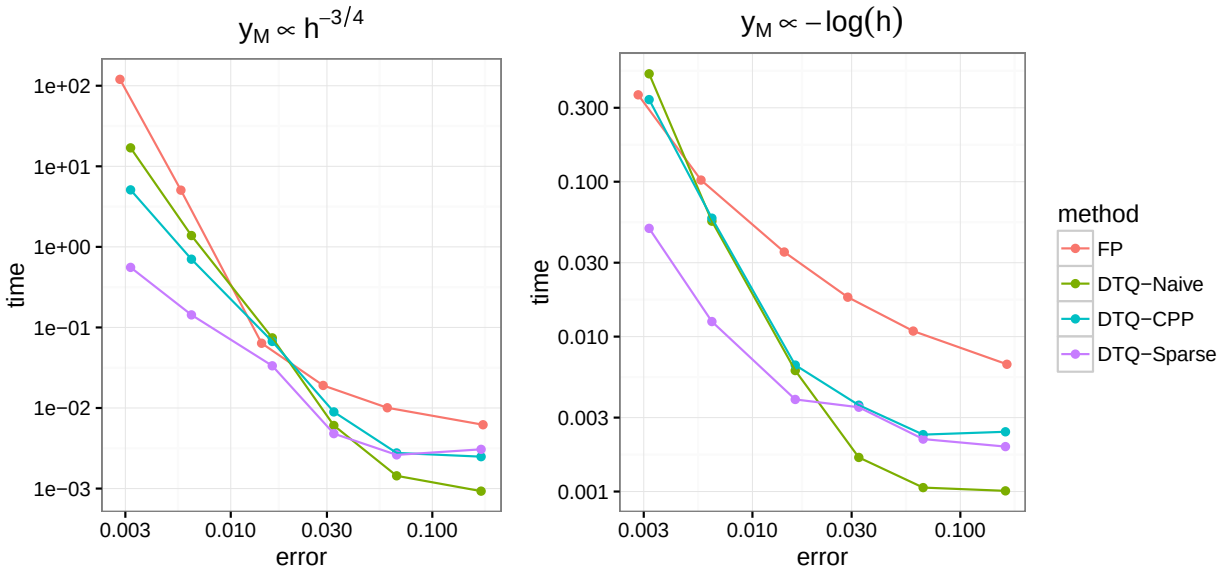

Figure 3.2: For a particular SDE, Example 1 from (3.65), suppose we are interested in computing the density $p(x,T)$ at time $T=1$. When we compute this density, we will incur some error, measured here in the L^1 norm. **The plotted results show that for a fixed value of this error, the DTQ methods require less computational time (measured in wall clock seconds) than a method for numerically solving the Fokker-Planck PDE.** In all simulations, we use a domain $[-y_M, y_M]$. For the simulations in the left (respectively, right) plot, we have scaled the domain according to $y_M \propto h^{-3/4}$ (respectively, $y_M \propto \log h^{-1}$), where $h>0$ is the time step. In both plots, we see that for smaller values of the error, the fastest method is DTQ-Sparse; for larger values of the error, the fastest method is DTQ-Naïve. In particular, for the smallest error of 0.003, the DTQ-Sparse method is over 10^2 (respectively, $10^{3/4}$) times faster than the Fokker-Planck method in the left (respectively, right) plot. Despite the fact that our Fokker-Planck solver uses the same sparse numerical linear algebra as DTQ-Sparse, it is often the slowest of the four methods. For details regarding the three implementations of the DTQ method (DTQ-Naïve, DTQ-CPP, and DTQ-Sparse) as well as the implementation of our Fokker-Planck solver, please see Section 3.7.2.

Chapter 4

Parameter Inference for Scalar SDE via Maximum Likelihood Approach

4.1 Introduction

In this chapter, we discuss how to infer the unknown parameter vector θ in (1.1) by maximizing the likelihood function. This maximum likelihood estimate (MLE) for θ can be computed by minimizing the negative log likelihood functions given in (2.1) or (2.13) for the cases of single or multiple time-series observations respectively. When computing MLE, the gradient of the likelihood function is an important ingredient for numerical optimization procedures. The gradient $\nabla_\theta p(\mathbf{x}|\theta)$ can be computed in two different ways: using the direct computation of the gradient or using an adjoint-based method. The direct method involves computing the derivatives of the likelihood with respect to each element of the parameter vector θ. On the other hand, the adjoint method enables us to compute this gradient with a computational cost (in time) that does not scale with the dimension of θ. This is important for nonparametric estimation (when the functional form of the drift f, and the diffusion g are unknown) of SDE because we do not know the dimension of θ a priori—indeed, it may be quite large. Of course, we can apply both direct and the adjoint method to parametric inference problems where the functional forms of f and g are known. Such problems are much more commonly studied in the literature—see [58, 31, 26]. Equipped with efficient algorithms to compute

the likelihood and its gradient, we can then use quasi-Newton optimization methods to compute θ from data.

The structure of this chapter is as follows. In Section 4.2, we start with introducing the optimization problem to compute MLE. Then, in Section 4.2.1 we detail the direct method to compute the gradient of the likelihood for the cases of one time-series observations and many time-series observations. Subsequently, we discuss the adjoint-based gradient computation in Section 4.2.2. In Section 4.3, we discuss how to perform parametric inference using the direct gradient computation to compute the MLE. We demonstrate the effectiveness of the method through simulation tests of different SDE. Finally, in Section 4.4, we discuss how to perform nonparametric inference using the adjoint method developed in Section 4.2.2. Through simulated data tests, we show that when the number of sample paths is large, the method does an excellent job of inferring drift functions close to the ground truth. We also show that even when the method does not infer the drift function correctly, it still yields models with good predictive power. Finally, we apply the method to real data on hourly measurements of ground level ozone, showing that it is capable of reasonable results.

4.2 The Maximum Likelihood Estimation

Let us define the maximum likelihood estimate of θ by $\hat{\theta}$, and can be computed using

$$\hat{\theta} = \arg\min_{\theta} \left(- \log \mathscr{L}(\theta)\right). \tag{4.1}$$

In Section 2.2, we discussed how to compute the likelihood $\mathscr{L}(\theta)$ for scalar SDE using DTQ method. Now we are in a position of computing $\hat{\theta}$ by solving the optimization problem (4.1). To compute the maximum likelihood estimator of θ using a numerical optimization technique, we need to compute the gradient of $-\log \mathscr{L}(\theta)$ with respect to θ. We start by explaining the direct method of computing the gradient in Section 4.2.1. Subsequently, we discuss the gradient computation via adjoint method in Section 4.2.2.

4.2.1 Gradient of the Likelihood via the Direct Method

Direct Gradient Computation for One Time Series

Let us start with computing each element of the gradient $\nabla_\theta \log \mathscr{L}(\theta)$ by taking the derivative of $\log \mathscr{L}(\theta)$ with respect to each element θ_ℓ in the parameter vector θ. This gives us

$$\frac{\partial}{\partial \theta_\ell} \log \mathscr{L}(\theta) \approx \sum_{j=0}^{L-1} \frac{1}{p_{\widetilde{X}_{j+1}}(x_{j+1}|\widetilde{X}_j = x_j, \theta)} \frac{\partial}{\partial \theta_\ell} p_{\widetilde{X}_{j+1}}(x_{j+1}|\widetilde{X}_j = x_j, \theta). \tag{4.2}$$

Now we compute $\frac{\partial}{\partial \theta_\ell} p_{\widetilde{X}_{j+1}}(x_{j+1}|\widetilde{X}_j = x_j, \theta)$ from (2.6). Taking the derivative of (2.6) with respect to θ_ℓ gives

$$\begin{aligned}\frac{\partial}{\partial \theta_\ell} p_{\widetilde{X}_{j+n/F}}(y|\widetilde{X}_j = x_j; \theta) = &\int_z \frac{\partial}{\partial \theta_\ell} G^h_\theta(y,z) p_{\widetilde{X}_{j+(n-1)/F}}(z|\widetilde{X}_j = x_j; \theta)\, dz \\ &+ \int_z G^h_\theta(y,z) \frac{\partial}{\partial \theta_\ell} p_{\widetilde{X}_{j+(n-1)/F}}(z|\widetilde{X}_j = x_j; \theta)\, dz. \end{aligned} \tag{4.3}$$

Now following the same procedure explained in the Section 2.2.3, we truncate the domain of the integral and apply the trapezoidal rule on an equispaced grid for z and get

$$\begin{aligned}\frac{\partial}{\partial \theta_\ell} p_{\widetilde{X}_{j+n/F}}(y|\widetilde{X}_j = x_j; \theta) \approx &\, k \sum_{a_1=-M}^{M} \frac{\partial}{\partial \theta_\ell} G^h_\theta(y, z^{a_1}) p_{\widetilde{X}_{j+(n-1)/F}}(z^{a_1}|\widetilde{X}_j = x_j; \theta) \\ &+ k \sum_{a_1=-M}^{M} G^h_\theta(y, z^{a_1}) \frac{\partial}{\partial \theta_\ell} p_{\widetilde{X}_{j+(n-1)/F}}(z^{a_1}|\widetilde{X}_j = x_j; \theta). \end{aligned} \tag{4.4}$$

We can write (4.4) as

$$\hat{q}_{j+n/F,\ell} = K_\ell \hat{p}_{j+(n-1)/F} + K \hat{q}_{j+(n-1)/F,\ell}, \tag{4.5}$$

where $k \frac{\partial}{\partial \theta_\ell} G^h_\theta(y^{a_2}, z^{a_1})$ is the (a_2, a_1) element of a matrix $K_{\ell(2M+1)\times(2M+1)}$, and the a_1-th element of the vector $\hat{q}_{j+n/F,\ell}$ is given by $\hat{q}^{a_1}_{j+n/F,\ell} = \frac{\partial}{\partial \theta_\ell} p_{\widetilde{X}_{j+n/F}}(y^{a_1}|\widetilde{X}_j = x_j; \theta)$. Note that the initial vector $\hat{q}_{j+1/F,\ell}$ is given by taking the derivative of the right-hand side of (2.7) with respect to θ_ℓ, i.e.,

$$\frac{\partial}{\partial \theta_\ell} p_{\widetilde{X}_{j+1/F}}(y|\widetilde{X}_j = x_j; \theta) = \frac{\partial}{\partial \theta_\ell} G^h_\theta(y, x_j), \tag{4.6}$$

and discretized on the spatial grid. Again, as in Section 2.2.3, we apply (4.5) $F-2$ times to get

$$\hat{q}_{j+(F-1)/F,\ell} = K_\ell^{F-2}\hat{p}_{j+1/F} + K^{F-2}\hat{q}_{j+1/F,\ell},$$

where $\hat{q}_{j+(F-1)/F,\ell}$ is the approximation of $\frac{\partial}{\partial\theta_\ell}p_{\widetilde{X}_{j+(F-1)/F}}(y|\widetilde{X}_j = x_j;\theta)$ on the spatial grid. We want an approximation to $\frac{\partial}{\partial\theta_\ell}p_{\widetilde{X}_{j+1}}(y|\widetilde{X}_j = x_j;\theta)$ at $y = x_j$. To find this approximation without doing any interpolation as in Section 2.2.3, let us define the a_{F-1}-th element of the vector $\Gamma_{F-1,\ell}$ by

$$\Gamma_{F-1,\ell}^{a_{F-1}} = k\frac{\partial}{\partial\theta_\ell}G_\theta^h(x_{j+1}, x_{j+(F-1)/F}^{a_{F-1}}).$$

Using this together with our old definition of Γ_{F-1}, we have

$$\frac{\partial}{\partial\theta_\ell}p_{\widetilde{X}_{j+1}}(x_{j+1}|\widetilde{X}_j = x_j,\theta) \approx \left[\Gamma_{F-1,\ell}\right]^T \hat{p}_{j+(F-1)/F} + \left[\Gamma_{F-1}\right]^T \hat{q}_{j+(F-1)/F,\ell}.$$

Following we summarize algorithm to compute the elements in the gradient using the direct method.

1. We begin with
$$\hat{q}_{j+1/F,\ell}^{a_1} = \frac{\partial}{\partial\theta_\ell}G_\theta^h(x_{j+1/F}^{a_1}, x_j).$$

2. We then iteratively define, for $n = 1,\ldots,F-2$,
$$\hat{q}_{j+(n+1)/F,\ell} = K_\ell\hat{p}_{j+n/F} + K\hat{q}_{j+n/F,\ell}. \tag{4.7}$$

3. We finish with:
$$\frac{\partial}{\partial\theta_\ell}p_{\widetilde{X}_{j+1}}(x_{j+1}|\widetilde{X}_j = x_j,\theta) \approx \left[\Gamma_{F-1,\ell}\right]^T \hat{p}_{j+(F-1)/F} + \left[\Gamma_{F-1}\right]^T \hat{q}_{j+(F-1)/F,\ell}.$$

   Combining this with (2.11) and (4.2), we obtain

$$\frac{\partial}{\partial\theta_\ell}\log\mathscr{L}(\theta) \approx \sum_{j=0}^{L-1}\frac{1}{\left[\Gamma_{F-1}\right]^T K^{F-2}\hat{p}_{j+1/F}}\left[\Gamma_{F-1,\ell}\right]^T \hat{p}_{j+(F-1)/F} + \left[\Gamma_{F-1}\right]^T \hat{q}_{j+(F-1)/F,\ell}. \tag{4.8}$$

Direct Gradient for Multiple Time Series

The derivation of the gradient of $\log \mathscr{L}(\theta)$ given for one time series now proceeds just as before with $G^h_\theta(x_{j+1/F}, x_j)$ replaced by (2.15). The only changes required in the algorithm are, first, to redefine

$$\hat{q}^{a_1}_{j+1/F,\ell} = \frac{1}{\nu}\sum_{s=1}^{\nu} \frac{\partial}{\partial \theta_\ell} G^h_\theta(x^{a_1}_{j+1/F}, x^s_j)$$

and, second, to redefine $\Gamma_{F-1,\ell}$ as a matrix whose (a_F, a_{F-1}) entry is

$$\Gamma^{a_F,a_{F-1}}_{F-1,\ell} = k\frac{\partial}{\partial \theta_\ell} G^h_\theta(x^{a_F}_{j+1}, x^{a_{F-1}}_{j+(F-1)/F}).$$

With these changes, the gradient becomes

$$\frac{\partial}{\partial \theta_\ell}\log \mathscr{L}(\theta) \approx \sum_{j=0}^{L-1}\sum_{a_F=1}^{\nu} \frac{\left(\left[\Gamma_{F-1,\ell}\right]^T \hat{p}_{j+(F-1)/F} + \left[\Gamma_{F-1}\right]^T \hat{q}_{j+(F-1)/F,\ell}\right)_{a_F}}{\left(\left[\Gamma_{F-1}\right]^T K^{F-2}\hat{p}_{j+1/F}\right)_{a_F}}, \quad (4.9)$$

where $\hat{q}$ is computed using (4.7) just as before.

4.2.2 An Adjoint-Based Gradient of the Likelihood

In this section we discuss an adjoint-based approach to compute this gradient of the negative log likelihood. Let us first discuss the case of one time-series observations.

Adjoint Method for One Time Series

Recall from the Chapter 2 that for one time-series

$$-\log \mathscr{L}(\theta) \approx -\sum_{j=0}^{L-1} \log\left(\left[\Gamma_{F-1}\right]^T \hat{p}_{j+(F-1)/F}\right), \quad (4.10)$$

where

$$\hat{p}_{j+(F-1)/F} = K^{F-2}\hat{p}_{j+1/F}. \quad (4.11)$$

Let us first consider (4.11), which can be written as a system of block matrices to compute $\bar{p}_{j+1} = (\hat{p}^T_{j+1/F}, \cdots, \hat{p}^T_{j+(F-1)/F})^T$ as follows:

$$\bar{K}\bar{p}_{j+1} = v_{j+1}, \quad (4.12)$$

where $v_{j+1} = (\hat{p}^T_{j+1/F}, \cdots, 0)^T$, and

$$\bar{K} = \begin{pmatrix} I & O & O & \cdots & O \\ -K & I & O & \cdots & O \\ O & -K & I & \cdots & O \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ O & O & \cdots & -K & I \end{pmatrix}.$$

Here 0 denotes a zero row vector of the same length as $\hat{p}_{j+1/F}$. Also, O denotes a zero matrix and I denotes an identity matrix, both with the same dimension as that of the matrix K. Therefore, the block matrix $\bar{K}$ has dimension $(2M+1)(F-1) \times (2M+1)(F-1)$. Now define $w_{j+1} = (0, 0, \cdots, 0, \Gamma^T_{F-1})^T$, a vector of the same length as $\bar{p}_{j+1}$. Then we can write the negative log likelihood in (4.10) as

$$-\mathscr{L}(\theta) \approx -\sum_{j=0}^{L-1} \log\left({w_{j+1}}^T \bar{p}_{j+1}\right).$$

Let us define the Lagrangian as follows:

$$\mathbf{L}(\bar{p}_{j+1}, \theta, \bar{u}_{j+1}) = \sum_{j=0}^{L-1} \left[-\log\left({w_{j+1}}^T \bar{p}_{j+1}\right) + {\bar{u}_{j+1}}^T \left(\bar{K}\bar{p}_{j+1} - v_{j+1}\right) \right]. \tag{4.13}$$

Here, $\bar{p}_{j+1}$ is the state solution from (4.12) and $\bar{u}_{j+1}$ is a vector of Lagrange multipliers. Now, by taking the variations of the Lagrangian with respect to $\bar{u}_{j+1}$ and $\bar{p}_{j+1}$, we obtain the state equation,

$$\mathbf{L}_{\bar{u}_{j+1}}(\bar{p}_{j+1}, \theta, \bar{u}_{j+1}) = \bar{K}\bar{p}_{j+1} - v_{j+1} = 0, \tag{4.14}$$

and the adjoint equation,

$$\mathbf{L}_{\bar{p}_{j+1}}(\bar{p}_{j+1}, \theta, \bar{u}_{j+1}) = -\frac{w_{j+1}}{{w_{j+1}}^T \bar{p}_{j+1}} + \bar{K}^T \bar{u}_{j+1} = 0. \tag{4.15}$$

In the above equations, we have used the subscript on $\mathbf{L}$ to denote the variables with respect to which we have differentiated. Recall that θ is the vector of parameters in the drift and the diffusion of (1.1). We compute the gradient of the log likelihood by taking the derivative of the Lagrangian with respect to each θ_i:

$$\mathbf{L}_{\theta_i}(\bar{p}_{j+1}, \theta, \bar{u}_{j+1}) = -\sum_{j=0}^{L-1} \frac{{\bar{p}_{j+1}}^T \frac{\partial w_{j+1}}{\partial \theta_i}}{{w_{j+1}}^T \bar{p}_{j+1}} + \sum_{j=0}^{L-1} Q^T \bar{u}_{j+1} = -\frac{\partial \mathscr{L}(\theta)}{\partial \theta_i}, \tag{4.16}$$

where

$$Q = \frac{\partial \bar{K}}{\partial \theta_i} \bar{p}_{j+1} - \frac{\partial}{\partial \theta_i} v_{j+1}. \tag{4.17}$$

We rewrite (4.16) as

$$-\frac{\partial \mathscr{L}(\theta)}{\partial \theta_i} = -\sum_{j=0}^{L-1} \frac{\hat{p}_{j+(F-1)/F}^T \frac{\partial \Gamma_{F-1}}{\partial \theta_i}}{\Gamma_{F-1}^T \hat{p}_{j+(F-1)/F}} - \sum_{j=0}^{L-1} \left[\frac{\partial}{\partial \theta_i} \hat{p}_{j+1/F} \right]^T u_{j+1/F} - \sum_{j=0}^{L-1} \sum_{\ell=1}^{F-2} \left[\frac{\partial K}{\partial \theta_i} \hat{p}_{j+\ell/F} \right]^T u_{j+(\ell+1)/F}. \tag{4.18}$$

Let $\bar{u}_{j+1} = (u_{j+1/F}^T, u_{j+2/F}^T, \cdots, u_{j+(F-1)/F}^T)^T$. Then, the adjoint equation (4.15) can be written as the following temporally evolving system of equations, which can be solved backward in time for $n = 1, \cdots, F-2$:

$$u_{j+(F-(n+1))/F} - K^T u_{j+(F-n)/F} = 0, \tag{4.19a}$$

$$u_{j+(F-1)/F} = \frac{1}{\Gamma_{F-1}^T \hat{p}_{j+(F-1)/F}} \Gamma_{F-1}, \tag{4.19b}$$

The adjoint method to compute the gradient can now be summarized. For each fixed j, the procedure is as follows.

1 Given the unknown parameter vector θ, solve the state/forward problem equation (2.9) to find $\bar{p}_{j+1}$. This is the same as solving the time evolution system in (4.11).

2 Given θ and $\bar{p}_{j+1}$, solve the adjoint equation (4.15) to find $\bar{u}_{j+1}$. This is same as solving the time evolution system in (4.19a-4.19b).

3 With $\bar{p}_{j+1}$, $\bar{u}_{j+1}$, use (4.18) to compute the gradient.

We follow this procedure for each Markovian piece of the likelihood $p(x_{j+1}|x_j, \theta)$. For each piece, we need F steps in time to solve for the state $\bar{p}$ and F steps in time to solve for the adjoint $\bar{u}$. In total, across the entire time series, we need $2LF$ steps to compute the log likelihood and its gradient. In particular, note that the number of steps in time does not depend on the dimension of θ. This is in sharp contrast to a direct (i.e., non-adjoint) method computation of the gradient, in which one would have to perform a forward evolution in time (analogous to (4.11)) for each component of θ. Such a method would require $LF(1+|\theta|)$ steps to compute the log likelihood and its gradient, where $|\theta|$ is the dimension of θ.

Adjoint Method for Multiple Time Series

To compute the gradient for the case of many time-series using the adjoint method, we write the negative log likelihood in (2.16) as

$$-\mathscr{L}(\theta) \approx -\sum_{j=0}^{L-1}\sum_{a_F=1}^{\nu} \log\left(\mathbf{e}_{\mathbf{a_F}}^{\mathbf{T}} W_{j+1}\bar{p}_{j+1}\right), \tag{4.20}$$

where $\bar{p}_{j+1}$ comes from (4.12), and $\mathbf{e}_{\mathbf{a_F}}$ is the a_F-th canonical basis unit vector. Consider a row vector $\boldsymbol{\gamma}_{\mathbf{aF}}$, the a_F-th row of the matrix Γ_{F-1}. Then, each aF-th row of the matrix W_{j+1} consists of the vector $(0,0,\cdots,0,\boldsymbol{\gamma}_{\mathbf{aF}})$, where 0 denotes a zero row vector of length $2M+1$. Note that W_{j+1} is of dimension $\nu \times (2M+1)(F-1)$. Now the Lagrangian is given by

$$\mathbf{L}(\bar{p}_{j+1},\theta,\bar{u}_{j+1}) = \sum_{j=0}^{L-1}\left[-\sum_{a_F=1}^{\nu} \log\left(\mathbf{e}_{\mathbf{a_F}}^{\mathbf{T}} W_{j+1}\bar{p}_{j+1}\right) + \bar{u}_{j+1}{}^{T}\left(\bar{K}\bar{p}_{j+1} - v_{j+1}\right)\right]. \tag{4.21}$$

As a result, (4.14) remains the same. However, (4.15) becomes

$$\mathbf{L}_{\bar{p}_{j+1}}(\bar{p}_{j+1},\theta,\bar{u}_{j+1}) = -\sum_{a_F=1}^{\nu} \frac{W_{j+1}^{T}\mathbf{e}_{\mathbf{a_F}}}{\mathbf{e}_{\mathbf{a_F}}^{\mathbf{T}} W_{j+1}\bar{p}_{j+1}} + \bar{K}^{T}\bar{u}_{j+1} = 0. \tag{4.22}$$

Let us denote the quantity $\mathbf{e}_{\mathbf{a_F}}^{\mathbf{T}} W_{j+1}\bar{p}_{j+1}$ by a constant c_{a_F} that depends on a_F. Then, $\sum_{a_F=1}^{\nu} \frac{W_{j+1}^{T}\mathbf{e}_{\mathbf{a_F}}}{\mathbf{e}_{\mathbf{a_F}}^{\mathbf{T}} W_{j+1}\bar{p}_{j+1}}$ is a vector of length $(2M+1)(F-1)$ given by $(0,0,\cdots,\mathbf{y}^{T})^{T}$, where $\mathbf{y} = (\sum_{a_F=1}^{\nu} \frac{\gamma_{a_F}^{1}}{c_{a_F}},\cdots,\sum_{a_F=1}^{\nu} \frac{\gamma_{a_F}^{(2M+1)}}{c_{a_F}})^{T}$. Here $\gamma_{a_F}^{\ell}$ denotes the ℓ-th element of the vector $\boldsymbol{\gamma}_{\mathbf{a_F}}$. Now we can write (4.22) as the system of equations

$$u_{j+(F-(n+1))/F} - K^{T}u_{j+(F-n)/F} = 0, \tag{4.23a}$$

$$u_{j+(F-1)/F} = \mathbf{y}, \tag{4.23b}$$

which can be solved backward. Equation (4.16), required to compute the gradient, becomes

$$\mathbf{L}_{\theta_i}(\bar{p}_{j+1},\theta,\bar{u}_{j+1}) = -\sum_{j=0}^{L-1}\sum_{a_F=1}^{\nu} \frac{e_{a_F}{}^{T}\frac{\partial W_{j+1}}{\partial\theta_i}\bar{p}_{j+1}}{e_{a_F}{}^{T}W_{j+1}\bar{p}_{j+1}} + \sum_{j=0}^{L-1} Q^{T}\bar{u}_{j+1} = -\frac{\partial\mathscr{L}(\theta)}{\partial\theta_i}, \tag{4.24}$$

while (4.17), required to compute Q^{T}, remains the same. As a consequence of (4.24), only the first term in the right hand-side of (4.18) changes while other two terms remain

the same. After all of these changes, the algorithm to compute the gradient of (4.20) remains the same as explained under one time series. There are only two changes to make: (1) in the second step, we compute the adjoint solution by solving the system in (4.23a-4.23b) backwards; (2) we compute the gradient using (4.18) after replacing the first term by $-\sum_{a_F=1}^{\nu} \frac{\frac{\partial \gamma_{\mathbf{aF}}}{\partial \theta_i} \hat{p}_{j+(F-1)/F}}{\gamma_{\mathbf{aF}} \hat{p}^T_{j+(F-1)/F}}$.

4.3 Parametric Inference with Maximum Likelihood Approach

As we have already stated in Chapter 2, the parametric inference problem is the case when we know the functional forms of the drift f and the diffusion g of (1.1). However, f and g include some unknown constant parameters (given by the vector θ) that we want to infer from time-series data. In this section, we explain how we perform the inference and present numerical results for a number of different parametric SDE obtained using the maximum likelihood approach.

Inference

Given our discussion in Chapter 2 and the Section 4.2, The procedure for carrying out inference of the parametric problem is now straightforward. More specifically, we use the algorithms derived in Section 2.2.3 and Section 4.2.1 (or Section 4.2.2) to compute the objective function,

$$J(\theta) = -\log \mathscr{L}(\theta),$$

and its gradient respectively. We pass the objective function and its gradient to a numerical optimization package, NLopt, developed by [32]. Starting from an initial condition, NLopt finds a minimum of the objective function using one of many possible algorithms. In our work, we primarily use the low-storage BFGS algorithm described in [45, 46]. We use $\hat{\theta}$ to denote the minimizer of $J(\theta)$—this minimizer is our maximum likelihood estimate of θ.

Even though, we can compute the gradient of the likelihood using any of the methods discussed in Section 4.2.1 and 4.2.2, the results shown in this section are obtained only

using the direct method. All of the implementations used in this section are coded in *R*; we call algorithms from NLopt using **nloptr**. For more details on this package, consult [68]. Note that we are committed to making our algorithms available to the community; all codes used to produce the results of this paper can be downloaded at `http://github.com/hbhat4000/sdeinference/`.

Results

We now present numerical tests of our algorithm in three cases. For each case, we generate multiple sample paths using a specified SDE with known parameters. We use θ to denote the true parameter vector. Using the data thus generated, we then use our method to produce $\hat{\theta}$, our maximum likelihood estimate of the parameter vector. In the first two scenarios, the SDE we use for generating data coincides with the SDE used for inference. In the third scenario, we use a generic polynomial SDE for inference—this SDE includes as a special case of the SDE used for generating data.

To test the performance of the algorithm, we generate the data using the Euler-Maruyama approximation of the SDE. We step forward in time, starting from t_0 to a final time point $T > 0$. We use a step size of ξ, where $\xi = 10^{-4}$ unless specified otherwise. We retain the samples only at times $t = m\Delta t$ from $m = 0$ to $m = L$, where $L\Delta t = T$. For consistency during comparisons, we set $t_0 = 0$, $T = 25$, and $\Delta t = 1$.

Case 1: Linear SDE (Ornstein-Uhlenbeck process)

We consider the SDE for the Ornstein-Uhlenbeck process with linear drift and constant diffusion terms.

$$dX_t = \theta_1(\theta_2 - X_t)dt + \theta_3 dW_t \tag{4.25}$$

For the first set of experiments, the true parameter vector is $\theta = (0.5, 0, 1)$. We start the optimizer with an initial condition $\theta_0 = (1, 2, 0.5)$. We study how well we are able to infer the parameters as a function of the DTQ method's internal time step h and the number of sample paths. For this set of experiments, the spatial grid spacing k is set to $k = h^{0.75}$. In Table 4.1, we summarize all of this information together with the RMS (root-mean-square) error between the estimated and true parameter values. This

is equivalent to the 2-norm error, $\|\theta - \hat{\theta}\|_2$. We also record the number of iterations required for the optimizer to converge to the minimizer of the objective function, the negative log likelihood.

Estimated $\hat{\theta}$	Iterations	h	Paths	RMS Error
(1.020, 0, 1.404)	31	0.05	300	0.6597
(1.041, 0, 1.430)	30	0.02	300	0.6916
(1.048, 0, 1.438)	34	0.01	300	0.7028
(1.052, 0, 1.443)	34	0.005	300	0.7084
(1.054, 0, 1.445)	35	0.002	300	0.7119
(0.671, 0, 1.143)	31	0.01	100	0.2238
(0.673, 0, 1.146)	28	0.005	100	0.2264
(0.674, 0, 1.147)	26	0.002	100	0.2284

Table 4.1: Results for Case 1. Using either 300 or 100 sample paths produced by Euler-Maruyama simulation with time step $\xi = 10^{-4}$, we study the effect of reducing h, the internal time step of the DTQ inference method.

The method is not as sensitive to h as one might expect. Instead, what we find is that the error decreases when we decrease the number of sample paths. When we use only 100 sample paths, we obtain a qualitatively reasonable solution for all three components of θ, with θ_2 in particular identified up to machine precision.

To explore whether the above findings were peculiar to the way we generated the data, we conducted another series of tests starting with a true parameter vector of $\theta = (0.5, 0.9, 1)$. The results are displayed in Table 4.2. This time, when we use the Euler-Maruyama method to generate data, we use an internal time step of $\xi = 10^{-6}$, retaining all other parameters described above. For the inference, we give the optimizer an initial guess of $\theta_0 = (1, 0.5, 0.5)$. We again set the spatial grid spacing to $k = h^{0.75}$ and record the RMS error.

The results from Table 4.2 show that if we increase the number of sample paths from 50 to 300, the error decreases dramatically. This leads us to our view that, for the present version of our DTQ inference algorithm, the quality of the data is important. When we decrease the Euler-Maruyama time step from $\xi = 10^{-4}$ to $\xi = 10^{-6}$, we gain roughly one extra decimal place of accuracy in the sample paths. This leads our DTQ algorithm towards higher quality estimates of the parameters in the Ornstein-Uhlenbeck model (4.25).

The performance of the DTQ method should increase as the number of sample paths increases. In this regard, we believe the results from Table 4.1 are an artifact of how the data was generated. We will see confirmation of this in the results below on a nonlinear SDE model.

Additionally, we note that Table 4.2 confirms that the DTQ method's results are relatively insensitive to decreasing h, the internal time step of the DTQ method. Note that the data set we use for the experiments is collected at intervals of $\Delta t = 1$. We have found, in practice, that the choice $h = \Delta t/20$ is sufficient for inference. This is consistent with the results of [49], who chooses $h \approx \Delta t/25$.

Estimated $\hat{\theta}$	Iterations	h	Paths	RMS Error
(0.361, 0.968, 0.836)	39	0.050	50	0.2254
(0.362, 0.968, 0.839)	46	0.020	50	0.2226
(0.362, 0.968, 0.840)	42	0.010	50	0.2219
(0.362, 0.968, 0.841)	28	0.005	50	0.2212
(0.463, 0.885, 0.966)	45	0.050	300	0.05244
(0.466, 0.886, 0.973)	22	0.020	300	0.04561
(0.467, 0.886, 0.975)	22	0.010	300	0.04370
(0.468, 0.886, 0.976)	26	0.005	300	0.04237
(0.468, 0.886, 0.976)	20	0.002	300	0.04237

Table 4.2: Results for Case 1. Using either 300 or 100 sample paths produced by Euler-Maruyama simulation with time step $\xi = 10^{-6}$, we study the effect of reducing h, the internal time step of the DTQ inference method.

Case 2: Nonlinear SDE (Double Well Potential)

As our second example, we consider the following SDE with a nonlinear drift and constant diffusion term:

$$dX_t = \theta_1 X_t(\theta_2 - X_t^2)dt + \theta_3 dW_t \tag{4.26}$$

In Table 4.3, we show the results of an initial set of tests. In these tests, we vary both the true parameter vector θ and the initial guess θ_0 that is given to the optimizer. For all of these tests, the data consists of 100 sample paths and the DTQ grid spacing is given by $k = h^{0.75}$. Note that even when θ_0 is far from θ, the estimated parameters $\hat{\theta}$ are close to

True θ	Initial θ_0	Estimated $\hat{\theta}$	Iterations	h	RMS Error
(0.2, 1, 0.5)	(1, 1, 1)	(0.162, 0.886, 0.488)	37	0.05	0.06901
(0.4, 1, 0.5)	(1, 1, 1)	(0.629, 1.023, 0.618)	24	0.05	0.14965
(1, 4, 0.5)	(0.5, 0.5, 0.5)	(0.928, 3.990, 0.467)	50	0.01	0.04568
(1, 4, 0.5)	(2, 2, 1)	(0.925, 3.990, 0.430)	48	0.01	0.05935
(1, 4, 0.5)	(8, 8, 2)	(0.928, 3.990, 0.467)	47	0.01	0.04571

Table 4.3: Results for Case 2. We study a collection of problems involving different true θ values and different initial guesses θ_0.

θ. This trend holds for different values of θ. In fact, the RMS error values produced by our method are quite low for all of the tests involving the nonlinear model 4.26.

Next, in Table 4.4, we study the effect of decreasing the internal time step of the DTQ method, h, when all other problem/algorithm parameters are kept fixed. For these tests, we set $\theta = (1,4,0.5)$, $\theta_0 = (2,2,1)$, and $k = h^{0.75}$. The data consists of 100 sample paths. The results show that it is possible to slightly reduce the RMS error by decreasing h, the internal time step. Based on these results, we see that there is no disadvantage incurred by using our method with $h = 0.05$; at this internal time step, the method runs very quickly in R.

Estimated $\hat{\theta}$	Iterations	h	RMS Error
(0.925046, 3.990012, 0.430020)	37	0.05	0.05948
(0.925311, 3.990029, 0.430068)	48	0.01	0.05935
(0.926930, 3.990418, 0.471400)	48	0.005	0.04563
(0.925808, 3.990464, 0.473724)	41	0.002	0.04577
(0.925433, 3.990480, 0.474493)	31	0.001	0.04583

Table 4.4: Results for Case 2. We study the effect of decreasing h while keeping all other parameters fixed.

In Table 4.5, we run a series of tests where each test is repeated twice, once with the spatial grid spacing set to $k = h^{0.75}$ and again with $k = h$. For all of these tests, we generate data with $\theta = (1,4,0.5)$. If we examine the first two rows of Table 4.5, what we see is that decreasing the spatial grid spacing has a significant, beneficial effect on the RMS error. What has happened here is that we have given the optimizer an initial guess where the third element of θ_0 is 0.1, a relatively small value. If we go back to the SDE (4.26), we see that this third element of θ_0 corresponds to the diffusion coefficient. When the diffusion coefficient is small, the Gaussian kernel G_θ^h becomes very narrow.

This necessitates a finer spatial grid in order to resolve the kernel well enough to perform accurate quadrature. For the final four rows of Table 4.5, the third element of θ_0 is 1 and we do not observe as significant a reduction in RMS error when we refine the spatial grid.

Initial θ_0	Estimated $\hat{\theta}$	Iterations	k	Paths	RMS Error
(0.5, 0.5, 0.1)	(0.100, 4.024, 0.100)	39	$h^{0.75}$	100	0.5688
(0.5, 0.5, 0.1)	(1.035, 3.993, 0.499)	43	h	100	0.0205
(2, 2, 1)	(0.925, 3.990, 0.430)	48	$h^{0.75}$	100	0.0593
(2, 2, 1)	(0.955, 3.995, 0.481)	35	h	100	0.0283
(2, 2, 1)	(1.035, 3.993, 0.499)	75	$h^{0.75}$	300	0.0206
(2, 2, 1)	(1.022, 4.008, 0.497)	32	h	300	0.0138

Table 4.5: Results for Case 2. We compare spatial grid laws $k = h^{0.75}$ and $k = h$.

Finally, in Table 4.6, we study the effect of increasing the number of Euler-Maruyama sample paths in the data set that we feed into the inference algorithm. We keep all other algorithm and problem parameters fixed, with $\theta = (1,4,0.5)$, $\theta_0 = (2,2,1)$, $h = 0.01$, and $k = h^{0.75}$. The results show a steady improvement in the estimated $\hat{\theta}$ as the number of sample paths increase. The last row of Table 4.6 contains our best result for this inference problem with an RMS error less than 0.01.

Estimated $\hat{\theta}$	Iterations	Paths	RMS Error
(0.776, 4.060, 0.424)	100	2	0.1408
(0.899, 3.992, 0.510)	27	10	0.0583
(0.833, 4.018, 0.440)	35	50	0.1030
(0.925, 3.990, 0.430)	48	100	0.0593
(0.901, 4.007, 0.464)	33	200	0.0609
(1.035, 3.993, 0.499)	75	300	0.0206
(1.107, 3.994, 0.513)	43	400	0.0624
(0.988, 3.999, 0.489)	33	1000	0.0094

Table 4.6: Results for Case 2. We examine the effect of increasing the number of sample paths in the data set, keeping all other parameters fixed.

Case 3: Generic Polynomial Drift and Diffusion Functions

For our third example, we consider the same nonlinear SDE (4.26) to generate the observation data for the inference problem. Even though, this section is devoted for parametric inference problems, we present our next set of results as a motivation example of a simple case of nonparametric inference problem: a topic we will discuss in Section 4.4 in detail. In other words, here we consider this case as an example of where we do not know the exact form of the drift function. Therefore, we assume the drift function has the form of a third degree polynomial in parameter θ. In other words, we consider the SDE

$$dX_t = (\theta_0 + \theta_1 X_t + \theta_2 X_t^2 + \theta_3 X_t^3)dt + \theta_4 dW_t, \tag{4.27}$$

and we infer the parameters $\theta = (\theta_0, \theta_1, \theta_2, \theta_3, \theta_4)$ in the SDE (4.27) from the observations generated using the SDE (4.26) to see if we recover the correct form of the drift function. Specifically, we should infer that the parameters θ_0 and θ_2 in (4.27) are zero.

In Table 4.7, we display our results for three values of h, the internal time step. We generate our data by simulating 100 sample paths of (4.26) with $\theta_1 = 0.2$, $\theta_2 = 4$, and $\theta_3 = 0.4$. Note that in terms of the inference model 4.27, this corresponds to $\theta = (0, 0.8, 0, -0.2, 0.4)$. For the initial guess, we use $\theta_0 = (0, 0, 0, 0, 0.5)$. In this particular set of tests, instead of the using the BFGS algorithm described above, we use the method-of-moving-asymptotes algorithm described by [61].

Overall, the results show that the inference algorithm correctly identifies the qualitative form of the model. That is, we find that the first and third components of $\hat{\theta}$ are close to zero, and the remaining components of $\hat{\theta}$ are also close to their true values.

Estimated $\hat{\theta}$	Iterations	h	RMS Error
(0.014, 0.619, -0.003, -0.154, 0.357)	69	0.005	0.0859
(0.014, 0.867, -0.003, -0.217, 0.424)	57	0.002	0.0334
(0.012, 0.766, -0.003, -0.192, 0.408)	89	0.001	0.0168

Table 4.7: Results for Case 3. We perform inference using model (4.27), which has a higher-dimensional parameter space than (4.26), the model used to generate the data.

4.4 Nonparametric Adjoint-Based Inference

Consider a noisy, time-dependent system with a scalar observable. We define a time series to be a sequence of measurements, at regularly spaced times, of one sample path (or one realization) of the system. Suppose we have access to many sample paths; we observe all of the paths at regularly spaced times, thereby obtaining many time series. Starting with this type of data, we consider two questions:

1. How can we use all of the time series data to infer a fundamental equation of motion?

2. How well does such a model perform in predicting the future distribution of the observable variable?

We frame this problem as a non-parametric inference problem for the drift and diffusion functions of a general stochastic differential equation (SDE):

$$dX_t = f(X_t)dt + g(X_t)dW_t. \tag{4.28}$$

In this SDE, the drift function f and the diffusion function g are considered to be unknown. Here also we assume we observe (4.28) at times $t_j = j\Delta t$ for some fixed time-step $\Delta t > 0$, for $j = 0, \ldots, L$. At each time t_j, we collect v samples of X_{t_j} and label these samples as $x_j \in \mathbb{R}^v$. We let $\mathbf{x} = x_0, x_1, \ldots, x_L$ denote all of the collected observations. Now, our goal is to use $\mathbf{x}$ to infer f and g from (4.28). We can then use (4.28) to predict the future distribution of X_t. The type of data we have described arises in a diverse set of practical examples: time series measurements of large numbers of neurons [60], bid-ask prices in online auctions [44], or the position of a moving robotic arm [67]. The data we describe can also be viewed as a particular case of longitudinal high-dimensional data, of importance in a variety of biomedical contexts [17, 37].

In our approach, we first expand both the drift and the diffusion using an appropriate basis function expansion. As a result, the parameter vector θ in our statistical model is a vector of basis expansion coefficients that give finite-dimensional representations of the drift and diffusion functions that we seek to infer. We then compute maximum likelihood estimates for θ using the DTQ method. While the DTQ method is by itself efficient, what is most relevant for the current nonparametric estimation problem is our ability to

combine the DTQ method with the adjoint method to compute the gradient $\nabla_\theta p(\mathbf{x}|\theta)$. As already mentioned earlier, the advantage of the adjoint method is, it enables the inference to scale well with the dimension of θ. This is important for nonparametric estimation, because the required dimension of θ in the basis expansion may be quite large.

Prior Work

Our work is related to but distinct from prior work carried out in both the machine learning and statistics communities. First, nonparametric inference of the drift function for the SDE model (4.28) has been approached using Gaussian process approximations [53, 4]; it is not immediately clear how well such an approach will carry over to the setting where we have multiple sample paths, each observed on a relatively coarse time scale. In [44], the authors seek to infer an empirical SDE from the type of data just described. However, the class of SDE considered there is rather different than the drift-diffusion SDE (4.28) considered here. The work of [39] also fits the context of using data to infer general stochastic models; here the authors propose a model whose stochastic structure is completely unspecified. In contrast, by constraining the model to be (4.28), we form a connection to statistical physics, e.g., the Fokker-Planck and Langevin equations. In this framework, our question is: given enough realizations of a stochastic system, how well can we infer its potential function? Moving outside the SDE framework, there are several approaches to use high-dimensional longitudinal data to build predictive models: a tree-based method [56], methods from functional data analysis [65], and time series methods [67], to name just a few.

Methodology

We assume that f and g are square-integrable, i.e., $f, g \in L^2(\mathbb{R})$. The Hermite functions $\{\psi_i(x)\}_{i=0}^{\infty}$ form an orthonormal basis of $L^2(\mathbb{R})$; additionally, we have that any $\phi \in L^2(\mathbb{R})$ can be represented as an expansion in Hermite functions,

$$\phi(x) = \sum_{i=0}^{\infty} c_i \psi_i(x), \tag{4.29}$$

where the coefficient c_j can be computed via $c_j = \int \phi(x)\psi_i(x)\,dx$. The j-th Hermite function is defined by

$$\psi_j(x) = (-1)^j(2^j j!\sqrt{\pi})^{-1/2}e^{x^2/2}\frac{d^j}{dx^j}e^{-x^2}. \tag{4.30}$$

For further definitions and properties of the Hermite functions, we refer the reader to [66].

To finite-dimensionalize the inference problem for (4.28), we expand the unknown functions f and g in the Hermite basis and then truncate the expansions:

$$f(x) \approx \sum_{i=0}^{N_f} \theta_i\psi_i(x) = \hat{f}(x;\theta) \tag{4.31a}$$

$$g(x) \approx \sum_{i=0}^{N_g} \theta_{N_f+1+i}\psi_i(x) = \hat{g}(x;\theta). \tag{4.31b}$$

These approximations of f and g induce an approximation of the original SDE (4.28) by the approximate SDE

$$dX_t = \hat{f}(X_t;\theta)dt + \hat{g}(X_t;\theta)dW_t. \tag{4.32}$$

Properties of the Hermite functions guarantee that both $\hat{f}$ and $\hat{g}$ and their derivatives are globally bounded. This is sufficient for the existence of a unique solution X_t of (4.32); moreover, we are guaranteed that for $t > 0$, the random variable X_t has a density function $p(x,t)$.

Let $\theta = (\theta_0,\ldots,\theta_{N_f},\theta_{N_f+1},\ldots,\theta_{N_f+N_g+1})$. Then the inference problem consists of using the data $\mathbf{x}$ to compute θ. Now, given that the SDE (4.32) is of the parametric form of (1.1), we compute the likelihood function as explained in Section 2.2.2. Similarly, we compute the gradient information required for the optimization following the explanation in Section 4.2.2.

Dirichlet Penalty Term

Thus far, we have assumed that the dimensionality of θ, which by (4.31) is $N_f + N_g + 2$, has been fixed ahead of time by the user. In any practical data science context, the danger is that by choosing N_f and N_g sufficiently large, one can infer functions $\hat{f}(x)$ and $\hat{g}(x)$ that do not yield predictive models on test data, even though the models may

fit the training data arbitrarily well. In many data science settings, we use computational methods such as cross-validation to choose optimal values of parameters such as N_f and N_g. Because our methodology already requires substantial computational effort, we seek an alternative method to choose parameters so as to avoid overfitting. Beyond computational considerations, we also seek a method that naturally constrains θ, guiding the optimizer away from large magnitude coefficients without the use of, e.g., ad hoc box constraints. We describe this method using $\hat{f}$, but all of our considerations apply to $\hat{g}$ as well.

We begin with the Dirichlet energy of $\hat{f}$:

$$E = \int_{x=-\infty}^{\infty} |\hat{f}'(x)|^2 \, dx. \tag{4.33}$$

Minimizers of E consist of constant functions $\hat{f}$. For a constant $\gamma \geq 0$, we propose to add γE to the negative log likelihood. Minimizing the resulting penalized log likelihood will yield estimates of $\hat{f}$ that are less oscillatory than would obtained by minimizing the negative log likelihood alone. This is analogous to the penalty term used in natural smoothing splines; the only difference is that we penalize the L^2 norm of $\hat{f}'$ rather than $\hat{f}''$. Due to properties of the Hermite basis, it is possible to substitute (4.31a) in (4.33) and evaluate E in closed form. The main property that we employ is

$$\psi_j'(x) = \sqrt{\frac{j}{2}}\psi_{j-1}(x) - \sqrt{\frac{j+1}{2}}\psi_{j+1}(x).$$

With this, we obtain

$$E(\theta) = \frac{1}{2}\theta_1^2 + \frac{N_f}{2}\theta_{N_f}^2 + \sum_{j=1}^{N_f-1}\left(\sqrt{\frac{j+1}{2}}\theta_{j+1} - \sqrt{\frac{j}{2}}\theta_{j-1}\right)^2 \tag{4.34}$$

The overall penalized objective function is then

$$J(\theta) = -\log \mathscr{L}(\theta) + \gamma E(\theta) \tag{4.35}$$

with the negative log likelihood defined by (2.13) and constant $\gamma \geq 0$.

Implementation

Using R, we have implemented the algorithms described under Methodology section. Specifically, we have coded functions that take as input a trial value of θ and the

data matrix **x**, and that return as output the penalized objective function (4.35) and its gradient with respect to θ. We pass these functions to a gradient-based optimizer. In this work, we make use of two R packages for optimization:

1. **nloptr** [68]: this package is an R interface to NLopt, a general-purpose optimization package [32]. From this package, we use the low-storage BFGS method labeled as "NLOPT_LD_LBFGS." Internally, this method uses a line search method to find an optimal step size once the step direction has been chosen.

2. **trustOptim** [14]: this package implements trust region methods for nonlinear optimization. Essentially, the method finds an optimal step direction once a step size (the trust region radius) has been computed. From this package, we use the SR1 (Symmetric Rank 1) method.

Both of the optimization methods are quasi-Newton methods that seek to numerically approximate the Hessian using previously computed gradients. Such methods are well-suited for our problem: the adjoint method described in Section 4.2.2 enables us to compute the gradient of the log likelihood with computational effort that does not depend on the dimensionality of θ.

Note that though the methods described under the Methodology section are valid for nonparametric inference of both f and g, for the present paper, we have implemented the nonparametric inference only for the drift function f. In the present implementation, we treat the diffusion function g as parametric; in the results that follow, we treat g as an unknown constant. Preliminary tests indicate that though simultaneous nonparametric inference of both f and g is possible, as one might expect, such inference requires more sample paths than are required for nonparametric of f only. The present code is written entirely in R; the adjoint-based computation of the gradient features a highly intuitive but relatively slow "for" loop over each time series or sample path (from 1 to v). For future work, we propose a reimplementation of this code in either C++ or Scala (in conjunction with Apache Spark), to eliminate the sample path bottleneck, and then study simultaneous nonparametric inference of the drift and diffusion functions.

Results

Hermite Inference Test

In our first set of tests, we use artificial data generated with known drift and diffusion functions f and g. In particular, for each $i \in \{0,1,2,3,4\}$, we consider (4.28) with

$$f(x) = \psi_i(x) \tag{4.36}$$

and a fixed diffusion coefficient $g = 1$. For each resulting SDE, we generate ν sample paths using the Euler-Maruyama scheme with initial condition $X_0 = 0$. When we step the solution forward in time, we use the fixed time step $h_{\mathrm{EM}} = 10^{-4}$, i.e.,

$$\widetilde{X}_{n+1} = \widetilde{X}_n + f(\widetilde{X}_n)h_{\mathrm{EM}} + g(\widetilde{X}_n)h_{\mathrm{EM}}^{1/2}Z_{n+1} \tag{4.37}$$

where Z_{n+1} is an independent sample of a standard Gaussian (mean 0 and variance 1) random variable. Here $\widetilde{X}_n$ is intended to approximate $X_{nh_{\mathrm{EM}}}$. We step the solution forward in time until $T = 4$, i.e., until $n = T/h_{\mathrm{EM}}$. However, we retain the solution only at times $t = 0,1,2,3,4$, i.e., when $n = t/h_{\mathrm{EM}}$. In this way, each sample path we generate consists of the solution of (4.28) sampled at five points in time, one of which is the initial condition. For each choice of f given by (4.36), we generate artificial data sets with $\nu = 100$, 1000, and 10000 sample paths.

For this problem, the parameter vector θ consists of coefficients $(\theta_0,\ldots,\theta_{N_f})$ in (4.31a) together with one coefficient θ_{N_f+1} that we use to parameterize the unknown (yet constant) diffusion function g. In the following set of tests, we set $N_f = 4$. We give the optimizer an initial guess consisting of $\theta_0 = (1,1,\ldots,1)$; for each of three values of the penalty parameter $\gamma \in \{0,50,500\}$, we run the optimizer to find an inferred parameter vector $\widehat{\theta}$. When we run the optimizer, we must choose internal parameters for the DTQ method. We choose an internal DTQ time step of $h = 0.02$, spatial grid spacing $k = h^{0.6}$, and grid truncation $M = 2\lceil \pi/k^{1.5} \rceil$. These parameters are chosen so that the DTQ method preserves the total probability of the density function as it evolves.

Because we generated the artificial data ourselves, we know that the ground truth parameter vector consists of $\theta_i = 1$, $\theta_5 = 1$, and all other entries of θ equal zero. In Figure 4.1, we plot the two-norm error between the inferred and ground truth parameter vectors, i.e., $\|\widehat{\theta} - \theta\|_2$, as a function of ν, the number of samples. Note that both axes

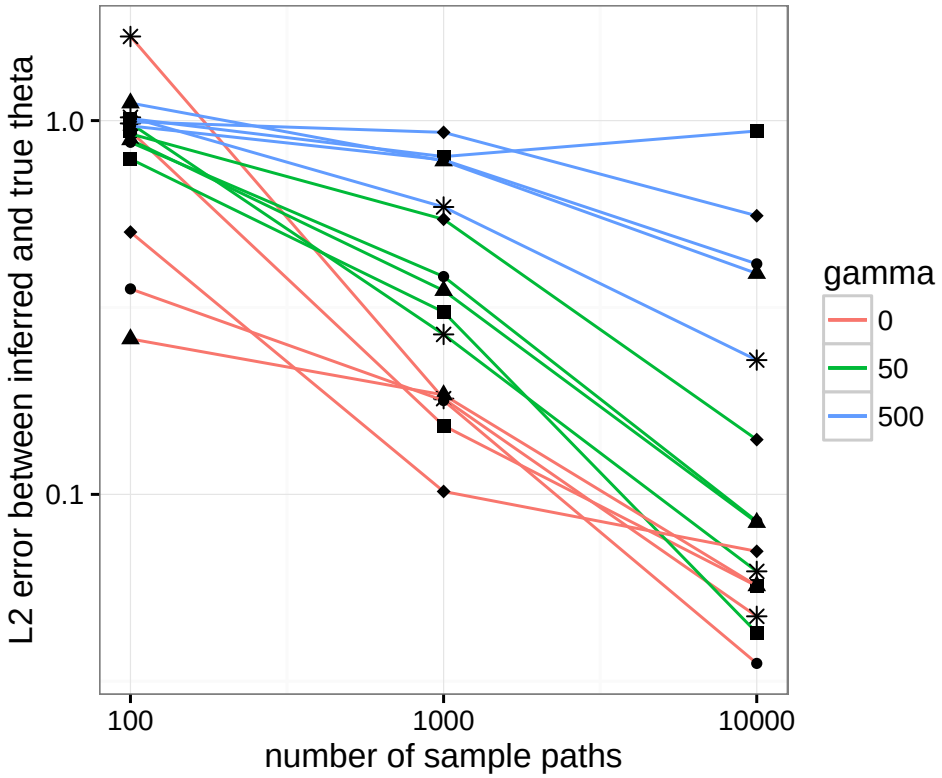

Figure 4.1: We plot the two-norm error between the true and inferred parameter vectors as a function of the number of sample paths ν. Note that both axes are log-scaled. For each value of ν, there are 15 data points corresponding to 3 possible values of γ, the penalty parameter, and 5 possible drift functions $f(x)$. In particular, the asterisk, square, triangle, diamond, and circles correspond to, respectively $i = 0,1,2,3,4$ in (4.36). Overall, we see that the error decreases as the number of sample paths increases; for the $\gamma = 0$ case, a least-squares line has slope -0.5285, consistent with an error rate of $\nu^{-1/2}$.

on this plot have been logarithmically scaled. The overall trend is clear: as we feed our algorithm a larger number of sample paths, we obtain estimates that are closer to the ground truth values. We also see that, for this simple test, the penalty term is not necessary; the best performance is achieved by setting $\gamma = 0$. If we isolate the $\gamma = 0$ data points from the plot and fit a least-squares regression line to the log-transformed data, the line has slope -0.5285, consistent with an error decay rate of $\nu^{-1/2}$. When the number of sample paths is limited, i.e., $\nu = 100$, it is possible to achieve better performance by setting $\gamma = 50$. Even in this case, the improved inference is observed for some, but not all, choices of $\psi_i(x)$ in (4.36).

In Figure 4.2, we focus our attention on the results obtained when $\gamma = 0$ and $\nu = 10000$, i.e., the best results from Figure 4.1. For each $i = 0,1,2,3,4$, we plot the true drift function $\psi_i(x)$ (in green) together with the inferred drift function $\hat{f}(x)$ (in red). Using the R function optimHess, we recover the numerical value of the Hessian matrix at the optimal solution $\widehat{\theta}$. This numerical Hessian is the observed Fisher information matrix $I(\theta)$. Let θ^{SE} denote the square root of the diagonal of $I(\theta)^{-1}$; this vector is an estimate of the standard error of $\widehat{\theta}$. Using these standard errors, we compute all $2^5 = 32$ possible curves $\hat{f}(x)$ that result from taking $\theta^* = (\theta_0 \pm \theta_0^{\mathrm{SE}}, \ldots, \theta_4 \pm \theta_4^{\mathrm{SE}})$. We plot the upper and lower envelopes of these curves as dotted lines in Figure 4.2. Overall, Figure 4.2 shows excellent agreement between the true and inferred drift functions. When there is disagreement, the true drift function almost always lies between the dotted lines, i.e., between our upper/lower band estimates for $\hat{f}$.

For this same case ($\gamma = 0$ and $\nu = 10000$), we also mention the inferred values of the constant diffusion coefficient $\hat{g}$. Recall that the true value is $g = 1$. For each i, the inferred values are 0.993,1.012,0.994,0.986, and 0.995. The standard errors for these estimates—obtained in the same way as above—are 0.00762, 0.00636, 0.00873, 0.00936, and 0.00883.

Prediction Test

In the next test, we focus our attention on the predictive power of the inferred model. For a particular data set $\mathbf{x}$, we set a window size $w \geq 2$. Let $a \vee b$ denote the maximum of the two integers a and b. For each $j \geq 1$, we then use the windowed data $\{x_{(j-w+1)\vee 1}, \ldots, x_j\}$ to train our model, i.e., to run our optimization method and infer a parameter vector $\widehat{\theta}$. We then use this inferred parameter vector in (4.32). Using the true data vector x_j as a vector of initial conditions, we step (4.32) forward in time until we reach the time corresponding to x_{j+1}. Let us label the samples thus obtained as $\widehat{x}_{j+1}$; we treat these samples as predictions. We then compute the Kolmogorov-Smirnov distance $E_j = \sup_z |F_{\widehat{x}_{j+1}}(z) - F_{x_{j+1}}(z)|$. Here F_y stands for the empirical cumulative distribution function (CDF) of the sample vector y. To see what the test error is like when we *know* the drift function perfectly, we also use $\hat{f}(x) = \sin x$ in (4.32) and compute the error just as above for each j. In what follows, we refer to this as the "control" parameter. Note

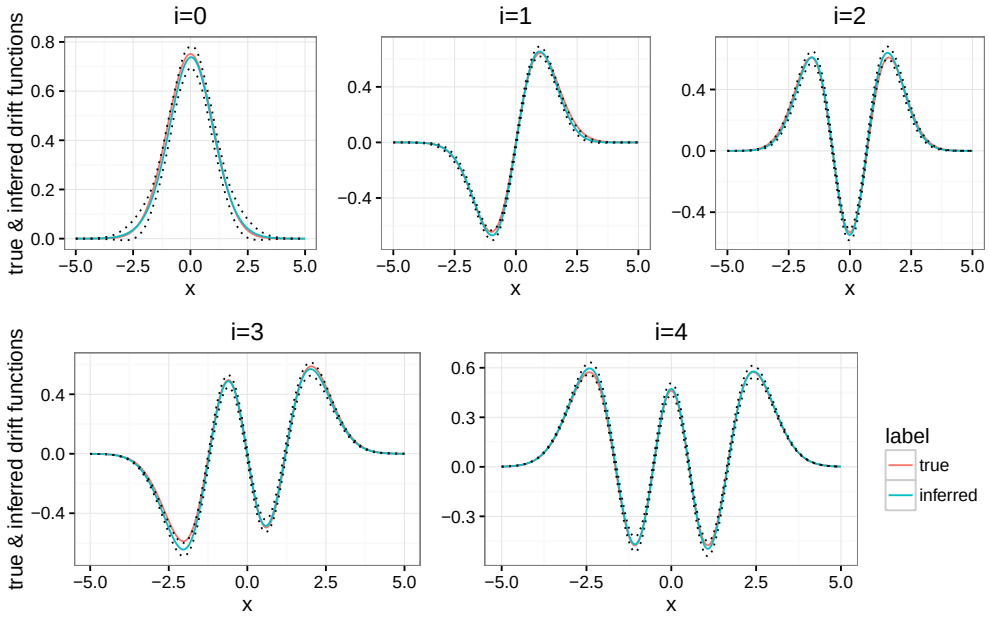

Figure 4.2: For each value of i, we plot the true drift function $\psi_i(x)$ (in green), the inferred drift function $\hat{f}(x)$ (in red), and upper/lower standard error bands about $\hat{f}(x)$ (in dotted lines). Overall, we find excellent agreement between the true and inferred drift functions. These results correspond to the $\gamma = 0$, $\nu = 10000$ results from Figure 4.1. Further details are given in the main text.

that in this test, we do not use the penalty parameter, i.e., we set $\gamma = 0$.

As we go through the data set, incrementing j by 1 and computing the test set error repeatedly, we try to avoid recomputing the parameter vector from scratch. Specifically, the parameter vector $\widehat{\theta}$ that is obtained using training data that includes x_j is used as an initial guess when we optimize with training data that includes x_{j+1}. For this test, we use the `nloptr` optimization package mentioned above. If the optimizer fails to find a minimizer, we restart the optimizer with an initial guess perturbed by uniformly distributed $U(-0.1, 0.1)$ noise and an initial diffusion coefficient of 0.7. For the very first initial guess, we take $\theta = (0, \ldots, 0, 0.7)$. As in the first test, we set $N_f = 4$.

For the internal DTQ parameters, we use the same parameters as in Hermite inference test, except for the grid truncation level M. We compare two values for M: $\tilde{M}_1 = \lceil \pi / k^{1.5} \rceil$ and $\tilde{M}_2 = 2\tilde{M}_1$. As for the artificial data generation, we again follow the same procedure as in Hermite inference test. Compared to what we described above, the only differences are that we now take $T = 50$ and $\nu = 100$; despite the small Euler-

Maruyama time step h_{EM}, we still save the solution only at integer times t. We perform the above test for window sizes $w = 2$, $w = 10$, and $w = 50$. Because $T = 50$, $w = 50$ corresponds to using all available past data to train the model.

We plot the results in Figure 4.3. The plotted curves lead to four main conclusions. First, there are several test set error curves that stay consistently below 0.15, indicating good agreement between the predicted and true CDFs. This is true in spite of the fact that the true drift function $f(x) = \sin x$ explicitly violates an assumption made in the Methodology section ; in particular, $f(x)$ is not square-integrable, i.e., $f \notin L^2$. Second, in the DTQ method, we must pay attention to the size of the spatial domain. If we set $M = \tilde{M}_1$, it turns out that the spatial domain is too small to preserve the total area under the density curve as it evolves forward in time. As we will see, this leads to unreasonably large estimates of the coefficient vector θ. Once we set $M = \tilde{M}_2$, the method performs far better. Third, the window size has an insignificant effect on the test set error. The three test set error curves corresponding to $A = A_2$ are quite close to one another, even when w is as small ($w = 2$) or as large ($w = 50$) as possible.

The fourth conclusion is perhaps the most interesting one: the seemingly ideal case in which $\hat{f}(x) = f(x) = \sin x$ leads to rather poor test set error, as we can see from the control curve. We hypothesize that because the number of sample paths is fairly small ($\nu = 100$), the artificial data we have generated may at times be more consistent with a drift function *other* than $f(x)$. In Figure 4.4, we have plotted five drift functions: the true drift $f(x)$ together with four inferred drifts corresponding to the $j = 50$ points in Figure 4.3. The $M = \tilde{M}_1$ case leads to unreasonably large coefficients, which we see immediately in the plot. The $M = \tilde{M}_2$ cases lead to perfectly reasonable drift functions, only one of which bears much resemblance to $f(x) = \sin x$. Overall, we are encouraged by the fact that our method can pick out predictive models from the data even when the inference is far from the ground truth.

Real Data Test

Our final test involves real data obtained from the California Air Resources Board [13]. Specifically, we obtained data consisting of hourly measurements of ground level ozone (in parts per million) conducted daily at numerous sites across California in the

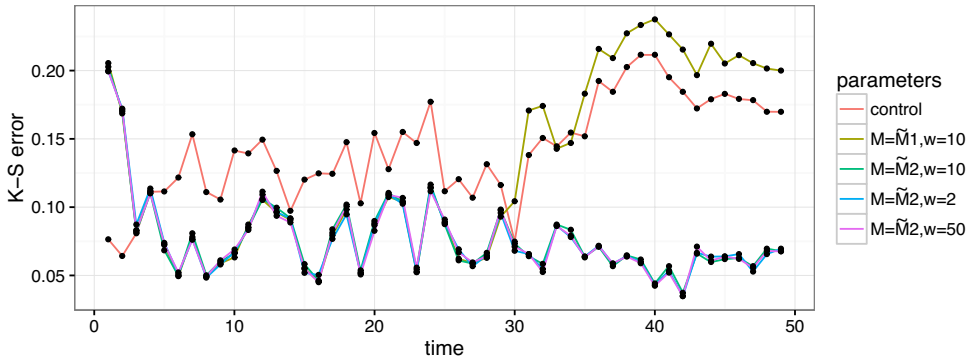

Figure 4.3: Following the procedure in the prediction test, we evaluate the predictive power of models inferred using our method. Specifically, for each point in time j, we plot the Kolmogorov-Smirnov (K-S) error between the empirical distributions of x_{j+1} (real data) and $\widehat{x}_{j+1}$ (predictions). We form the predictions by inferring the drift and diffusion functions, and then using them in a forward simulation of the SDE (4.32). The parameters A and w are, respectively the spatial grid truncation in the DTQ method and a window size, i.e., how many points along each sample path are actually used to fit the model. The $A = A_2$ results show that as long as the DTQ method's spatial grid truncation is chosen well, the inferred models yield good predictive results. These results also show insensitivity to the window size w. The control curve corresponds to predictions generated from $f(x) = \sin x$, i.e., the drift function used to create the data set.

years 2010-2011. We first narrowed our scope to 25 sites in the San Joaquin Valley. For these sites, we treated each day as a separate time series with 24 points. When fewer than 24 were available, we used linear interpolation to impute missing points; however, this procedure was only applied if the time series had at least 16 points to begin with. In this way, we formed a data set consisting of $\nu = 16675$ sample paths, each with 24 points. To allow us to use a large spatial grid spacing in the DTQ method, we multiplied each ozone level by a factor of 10. We then applied our inference method to this data set. The specific DTQ parameters used are: $h = 0.02$, $k = h$, and $M = \lceil 0.5/k^{1.5} \rceil$. As before, we take $N_f = 6$.

As one might expect, with a real data set, the objective function required penalization in order to yield reasonable results. We chose a penalty parameter of $\gamma = 50$ based on several trials. We fit the model using the first 6 hours worth of data, i.e., a window size of $w = 6$. The drift function together with upper/lower standard error bands is displayed in Figure 4.5. For this test, the inferred diffusion coefficient is $\hat{g} = 0.141$.

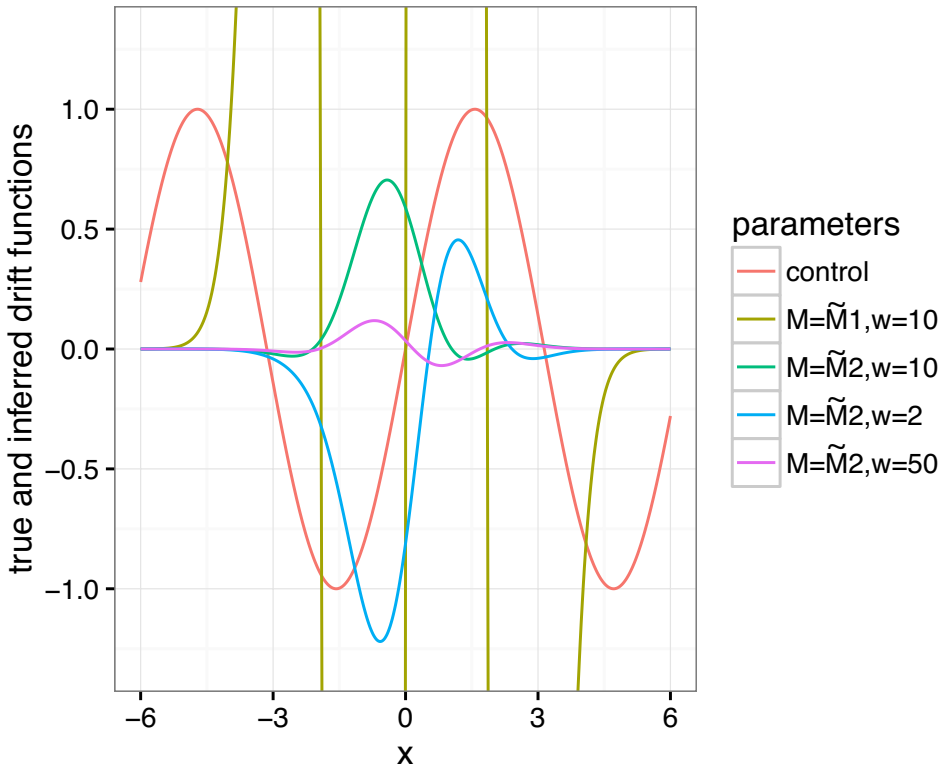

Figure 4.4: We plot the true drift function $f(x) = \sin x$ (control) together with four inferred drift functions $\hat{f}(x)$. These drift functions correspond to the $j = 50$ runs from Figure 4.3. Combining what we see here with Figure 4.3, we conclude that even though the $A = A_2$ drifts are not close to $\sin x$, they still yield good test set results.

Note that when running using the `nloptr` optimization package for this test, we have constrained the diffusion function via $\hat{g} \geq \sqrt{k}$. In the DTQ method, as the diffusion function approaches zero at a particular point, it becomes impossible to resolve the Gaussian kernel G_θ^h on a spatial grid with fixed spacing $k > 0$. In the particular inference problem we describe here, the optimizer does tend to send the diffusion constant to its lower bound—we believe this indicates the need for a spatially-dependent (rather than constant, as assumed here) diffusion function $\hat{g}$.

Following the same procedure outlined in Prediction test section, we generated predictions for the next 6 hours. The Kolmogorov-Smirnov errors between the true and predicted CDFs are, respectively, 0.148, 0.161, 0.164, 0.179, 0.190, and 0.220. The errors in the true and predicted means are, respectively, 0.0549, 0.0657, 0.0772, 0.0867,

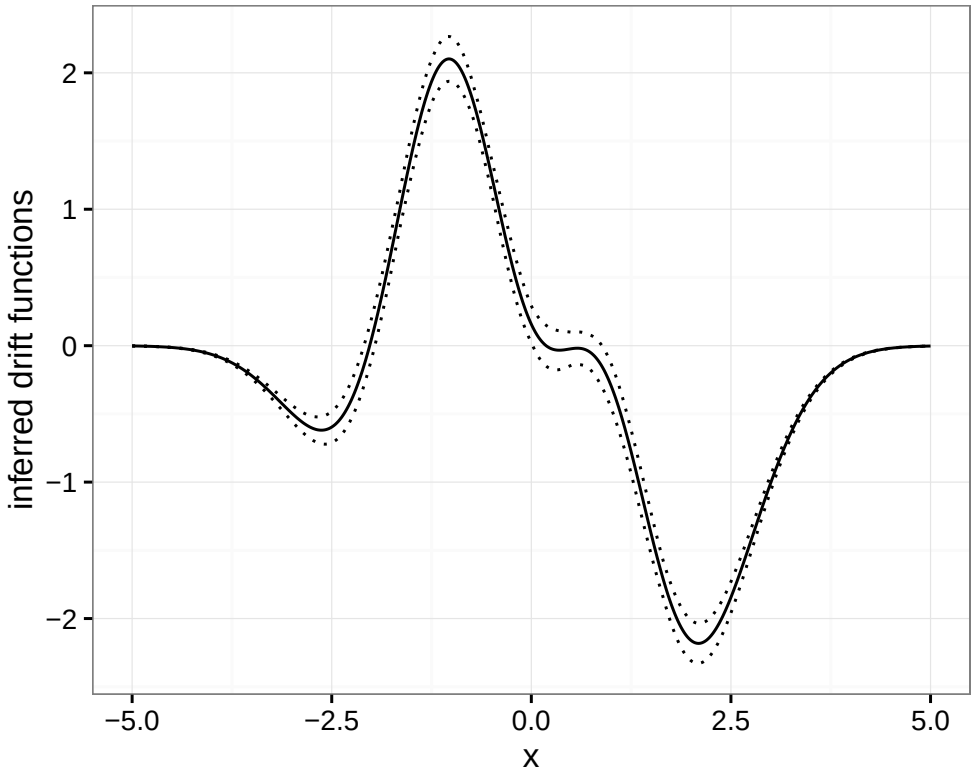

Figure 4.5: The inferred drift $\hat{f}(x)$ for hourly ozone levels in the San Joaquin Valley, together with upper/lower standard error bands.

0.1069, and 0.1333. As expected, the errors increase as we get further away from the last data point used to build the model. Please note that to convert the errors in the mean to parts per million, we must divide by 10; we believe the resulting errors indicate reasonable agreement between predictions and reality.

4.5 Discussion and Conclusion

In this chapter we discussed how to use the DTQ method to compute the maximum likelihood estimates. We presented the derivations of the gradient computation required to maximize the negative log likelihood using two different approaches: the direct method and the adjoint method. We used the direct method to perform parametric inference of SDE. We demonstrated the method's performance through simulations tests of three different cases of SDE. Specifically, we observed that with high quality data ob-

tained from Euler-Maruyama sampling, the method is capable of computing accurate estimates of θ even with initial guesses that are far from the true parameters.

Next, in this chapter, we have developed a new adjoint-based nonparametric method to infer the drift and diffusion functions for a stochastic differential equation. We have demonstrated through three tests that the method has the capability to succeed. There are three main future directions to improve the current framework. First, we have only explored the Hermite function basis in this work and we believe other bases may yield better results for particular data sets. Second, we have not yet tested the method on inference problems where we seek a non-constant diffusion function $g(x)$. Third, we plan to seek out improved methods to find optimal values of the penalization parameter γ. All of these tasks will be enabled by porting computationally intensive parts of our algorithm to a compiled language.

Chapter 5

Inference for Coupled SDE: Metropolis Algorithms via Density Tracking by Quadrature

5.1 Introduction

In this chapter, we perform Bayesian inference for the coupled SDE with time-dependent coefficients introduced in Section 2.3. In contrast to MLE approached discussed in Chapter 4, the goal of the Bayesian approach is to sample from the posterior distribution which is proportional to the product of the likelihood and the prior distribution. In this work, we develop a MCMC algorithm to perform Bayesian inference for coupled SDE. The MCMC algorithm is derived using a Metropolis scheme; our innovation is to evaluate the log likelihood efficiently using the DTQ method. We reiterate that inference of SDE models is a challenging problem, due to the fact that a closed-form likelihood function is generally unavailable [58, 31, 26]. Most existing parametric inference methods for discretely observed SDE require inter-observation times to be small. As a way to facilitate approximation of the transition density for parametric inference for large inter-observation times, Bayesian methods are used to simulate missing values of the observations to form a high-frequency data set. In situations where the likelihood function is either analytically unavailable or computationally prohibitive

to evaluate, Bayesian inference of SDE makes use of likelihood-free methods such as Approximate Bayesian Computation [50], variational methods [5, 64], and/or Gaussian processes [4, 53].

5.2 Bayesian Inference

Recall the Inference problem of coupled SDE in Section 2.3. Our goal is to Infer unknown parameters in vector θ from time-series observations $\mathbf{x}$. The posterior density of the parameter vector given the observations is $p(\theta\,|\,\mathbf{x}) \propto p(\mathbf{x}\,|\,\theta)p(\theta)$, where $p(\mathbf{x}\,|\,\theta)$ is the likelihood and $p(\theta)$ is the prior. We approximate the likelihood $p(\mathbf{x}\,|\,\theta)$ using the DTQ method as discussed in the Section 2.3.

5.2.1 Metropolis Algorithm

Here we embed the DTQ method's likelihood computation into a Metropolis algorithm to sample from the posterior. In the Metropolis algorithm, we construct an auxiliary Markov chain $\{\hat{\vec{\theta}}_N\}_{N\geq 0}$ which is designed to have an invariant distribution given by the posterior $p(\theta\,|\,\mathbf{x})$. This Markov chain is constructed as $\hat{\vec{\theta}}_{N+1} = \hat{\vec{\theta}}_N + Z_{N+1}$, where Z_{N+1} is a random vector with dimension equal to that of the parameter vector θ. In this paper, we choose all components of Z_{N+1} to be independent normal random variables with known means and variances. The Metropolis algorithm is as follows:

- Choose value q_0 for $\hat{\vec{\theta}}_0$.
- Once the values $q_0, \cdots, q_N$ of $\hat{\vec{\theta}}_0, \cdots, \hat{\vec{\theta}}_N$ have been found:
  - Generate a proposal from the auxiliary Markov chain: $q^*_{N+1} = q_N + Z_{N+1}$.
  - Calculate the ratio $\rho = \frac{p(q^*_{N+1}\,|\,\mathbf{x})}{p(q_N\,|\,\mathbf{x})}$, where $p(q^*_{N+1}\,|\,\mathbf{x}) \approx \widetilde{p}(\mathbf{x}\,|\,q^*_{N+1})p(q^*_{N+1}) = p(q^*_{N+1})\prod_{m=0}^{L-1}\widetilde{p}(\mathbf{x}_{m+1}\,|\,\mathbf{x}_m, q^*_{N+1})$. Now each term $\widetilde{p}(\mathbf{x}_{m+1}\,|\,\mathbf{x}_m, q^*_{N+1})$ can be computed using the DTQ method discussed in Section 2.3.
  - Sample $u_N \sim \mathscr{U}(0,1)$. If $\rho > u_N$ set $\hat{\vec{\theta}}_{N+1} = q^*_{N+1}$; in this case, the proposal is accepted. Else set $\hat{\vec{\theta}}_{N+1} = q_N$ and the proposal is rejected.

Once we have obtained all the samples $q_0, q_1, \cdots, q_N$ from the Metropolis algorithm, we discard a sufficient number of initial samples to ensure the Markov chain has converged to its invariant distribution.

5.3 Numerical Tests

We implement the Metropolis algorithm in R. Inside the Metropolis algorithm, we evaluate the likelihood function using the DTQ method, which is implemented in C++ as an R package.

5.3.1 Stochastic Van der Pol Oscillator

To test the method, we first consider the nonlinear SDE

$$\mathrm{d}X_{1,t} = \theta_1 X_{2,t}\,\mathrm{d}t + (0.1 + \theta_4^2\, e^{-X_{1,t}^2})\,\mathrm{d}W_{1,t}, \tag{5.1a}$$

$$\mathrm{d}X_{2,t} = (-\theta_2 X_{1,t} + \theta_3 X_{2,t}\,(1 - X_{1,t}^2))\,\mathrm{d}t + (0.1 + \theta_5^2\, e^{-X_{2,t}^2})\,\mathrm{d}W_{2,t}. \tag{5.1b}$$

This system describes a noisy van der Pol oscillator. The presence of $X_{1,t}$ and $X_{2,t}$ in the diffusion function ensures that the transition density is not Gaussian. To generate simulated data, we start with known values of the parameters: $\theta_1 = 1, \theta_2 = 1, \theta_3 = 4$ and the noise parameters $\theta_4 = \theta_5 = 0.5$. Using a fixed initial condition $(X_{1,0}, X_{2,0})$, we then use the Euler-Maruyama method to step (5.1) forward in time until a final time $T > 0$. When we carry out this time-stepping, we use a step size of 0.001 and simulate up to $T = 20$. We then retain every 100-th element to yield a data set consisting of 201 observations of X_1 and X_2 with spacing $\Delta t = 0.1$. In this way, we simulate large inter-observation times for a process that in reality operates at a finer time scale.

Using the samples $\{\mathbf{x}_m\}_{m=0}^{L}$ thus constructed, we run the Metropolis algorithm. We infer only the parameters in the drift function, i.e., θ_1, θ_2 and θ_3, keeping other parameters fixed at their known values. We initialize θ at $(0.1, 0.1, 0.1)$, far from the true θ values. We use a diffuse Gaussian prior with mean 0 and standard deviation 100. For the proposal distribution Z_{N+1} in the auxiliary Markov chain, we choose i.i.d. Gaussians with mean 0 and standard deviation 0.05.

Our goal here is to test the performance of the algorithm using simulated data and compare it against Bayesian particle filtering/inference method implemented in the R package "pomp" [34]. This method gives us an alternative, sampling-based approach to approximate the likelihood function. Note that we also compare DTQ against a purely Eulerian approximation of the transition density, i.e., a method where $\tilde{p}(\mathbf{x}_{m+1}|\mathbf{x}_m, \theta)$ is approximated by a Gaussian probability density function; this is equivalent to the DTQ method with zero internal quadrature steps, i.e., $h = \Delta t = 0.1$.

When we run the Metropolis algorithm, we discard the first 1000 samples and retain the next 20000 samples. We have settled on 20000 samples because, in this case, using the first 10000 post-burn-in samples does not yield significantly different results than what we obtained, i.e., we see no reason to continue sampling. We record the inferred parameter values, acceptance rate of the method (AR), and mean absolute percentage error (MAPE) for varying values of h for the 3 methods, Euler, DTQ and Pomp.

Parameters	θ_1	θ_2	θ_3	AR	MAPE	Method
$\Delta t = 0.1; h = 0.1/1$	0.747	0.906	3.070	0.296	0.193	Euler
$\Delta t = 0.1; h = 0.1/2$	0.866	1.300	4.260	0.285	0.168	
$\Delta t = 0.1; h = 0.1/4$	0.892	1.160	4.430	0.254	0.124	DTQ
$\Delta t = 0.1; h = 0.1/8$	0.980	1.170	4.210	0.239	0.081	
$\Delta t = 0.1; h = 0.1/2$	1.250	0.257	4.340	0.0003	0.361	
$\Delta t = 0.1; h = 0.1/4$	1.110	0.647	4.060	0.001	0.158	Pomp
$\Delta t = 0.1; h = 0.1/8$	1.040	0.752	3.940	0.0004	0.102	

Table 5.1: Simulation Test Results for Van der Pol Oscillator. We compare the performance of the DTQ method against Pomp and the Eulerian methods.

The first four rows of the Table 5.1 show that using the DTQ method to compute the likelihood yields more accurate posteriors than using a purely Gaussian likelihood (Eulerian method). In comparison to Pomp, our method does slightly better in terms of the means of the posteriors. If we look at the Metropolis samples generated by the two methods, the DTQ method has radically higher acceptance rates than Pomp. The non-adaptive version of the Metropolis sampling for Pomp does not explore the posterior adequately, rejecting thousands of consecutive proposals. Results in Table 5.2 shows that a carefully executed adaptive Metropolis algorithm for Pomp does generate better

results than the non-adaptive version.

Parameters	θ_1	θ_2	θ_3	AR	MAPE	Method
$\Delta t = 0.1; h = 0.1/2$	0.960	0.809	4.010	0.110	0.078	
$\Delta t = 0.1; h = 0.1/4$	1.000	0.954	3.990	0.164	0.017	Pomp-adaptive
$\Delta t = 0.1; h = 0.1/8$	1.010	1.010	4.020	0.171	0.009	

Table 5.2: Simulation Test Results for Van der Pol Oscillator from Pomp using an Adaptive Metropolis Algorithm. We study the performance with decreasing h.

One should take care to interpret the results in Table 5.2: we have invested a great deal of time to tune parameters in the adaptive MCMC scheme for pomp with full knowledge of the "true" parameter vector θ. Overall, what we have learned from this exercise is that there are two main investments of effort that one can make. In the DTQ method, we have invested effort into making the likelihood calculation more accurate, efficient and stable to initial choice of parameters. This allows us to use the DTQ method with a vanilla Metropolis algorithm and obtain reasonable results. One could instead have chosen to improve the vanilla Metropolis algorithm in various ways: adaptive MCMC, sequential MC, Hamiltonian MC, etc. This is the strategy pursued by Pomp. While both strategies have their merits, it seems that the likelihood computed by Pomp is not accurate enough to enable a vanilla Metropolis method to work well.

To understand this point in more detail, we have computed log likelihood surfaces in (θ_2, θ_3) space using both the Pomp and DTQ methods. If we rescale the log likelihood values from both methods so that they achieve a maximum of 0 and then exponentiate, we find that the DTQ likelihood has more reasonable gradients than the Pomp likelihood, which varies over 3 orders of magnitude. The accept/reject ratio depends on the actual density, i.e., the exponential of the log likelihood plus the log prior. Therefore, the sharp dropoff in the likelihood function at points very close to the maximum—seen in Pomp—will cause thousands of consecutive rejected steps. The more gradual dropoff in the DTQ likelihood function leads to a reasonable fraction of accepted steps in a vanilla Metropolis algorithm.

5.3.2 Stochastic Pursuit Models from Basketball Tracking Data

In 2010, the National Basketball Association (NBA) began to install a camera system to track the positions of the players and the ball as a function of time. For the ball and for each of the 10 players on the court, the system records an (x,y) position 25 times per second. Ultimately, this wealth of data should enable us to answer a number of questions regarding basketball strategy that would have seemed intractable just a few years ago. To bring this vision to reality, we must develop new algorithms that can efficiently use the data for inference of appropriate models.

In this work, we focus on so-called "fast break" situations where an offensive player races towards the basket in an attempt to score before the defensive team has time to set up their defense. In many such situations, it is relatively easy to identify from the data a runner and a chaser. This motivates the following question: using the NBA's spatial tracking data, how can we infer a stochastic model for the chaser's pursuit of the runner? To answer this question, we first formulate a stochastic version of the classical pursuit model. Our model consists of a set of coupled, nonlinear stochastic differential equations with time-dependent coefficients.

Derivation of the Model

Let the *runner* be the player (on offense) who has the ball and is running toward the basket. Let the *chaser* be the player (on defense) who is trying to prevent the runner from scoring. Let the current spatial coordinates of the runner and chaser be, respectively, $(x^r(t), y^r(t))$ and $(x^c(t), y^c(t))$.

Since the chaser is moving towards the runner, the velocity vector of the chaser points toward the runner's current position. Let $\vec{\phi} = (x^r - x^c, y^r - y^c)$. Then the unit vector that points toward the runner from the chaser is $\phi/\|\phi\|$. The velocity of the chaser, $(\dot{x}^c, \dot{y}^c)$, can thus be given as

$$(\dot{x}^c, \dot{y}^c) = \gamma(t)\phi/\|\phi\|, \tag{5.2}$$

where $\gamma(t) = \|(\dot{x}^c, \dot{y}^c)\|$, the instantaneous speed of the chaser. Note that (5.2) is a coupled system of nonlinear ordinary differential equations known as the pursuit model—classically, one assumes that $\gamma(t)$ and $(x^r(t), y^r(t))$ are given, in which case one typically

solves an initial-value problem for $(x^c(t), y^c(t))$. To generalize the classical model to the real data context considered here, we multiply both sides of (5.2) by dt and then add noise to each component:

$$\mathrm{d}(x^c, y^c) = \gamma(t)\left[\vec{\phi}/\|\vec{\phi}\|\right]\mathrm{d}t + (\nu_1\,\mathrm{d}W_t^1, \nu_2\,\mathrm{d}W_t^2) \tag{5.3}$$

Here $W_{1,t}$ and $W_{2,t}$ denote two independent Wiener processes with $W_{1,0} = W_{2,0} = 0$ almost surely. We refer to this model as the stochastic pursuit model. The question we want to answer is, given time-discrete observations of (x^c, y^c) and (x^r, y^r), how do we infer $\gamma(t)$ together with ν_1 and ν_2? For this stochastic pursuit model (5.3), we take $\mathbf{X}_t = (x^c(t), y^c(t))$. We treat $\gamma(t)$ as piecewise constant. Each constant value of $\gamma(t)$ is one component of the parameter vector θ; the final two components of θ are ν_1 and ν_2. If we treat $(x^r(t), y^r(t))$ as given, then we can identify the time-dependent drift functions f_1 and f_2 as the two components of $\gamma(t)\phi/\|\phi\|$. Next, we do a simulation test for our inference method using the pursuit SDE (5.3).

Simulation Test Results

We set the runner's trajectory equal to a sinusoidal curve $y = \sin(\pi x)$ from $x = -1$ to $x = 1$. We assume the runner covers this trajectory over the time period $0 \le t \le 8$. The chaser's trajectory is simulated using the Euler-Maruyama method to step (5.3) forward in time from a fixed initial condition $\mathbf{X}_0 = (x_0^c, y_0^c)$. During the generation of the data, we use a step size of 10^{-4}. By downsampling this single time series, we generate time series with spacings $\Delta t = 0.4, 0.2, 0.1$.

We set $\nu_1 = 0.15$, $\nu_2 = 0.1$, $\gamma(t) = \gamma_1 = 0.4$ for $0 \le t < 4$, and $\gamma(t) = \gamma_2 = 1.0$ for $4 \le t \le 8$. Because we want all speeds and diffusion constants to be positive, we take $\gamma_i = e^{\theta_i}$ and $\nu_i = e^{\theta_{i+2}}$ for $i = 1, 2$. The priors for θ_1 and θ_2 are normal with variance one and mean equal to the log of the mean speed of the chaser computed over the chaser's entire trajectory. The priors for θ_3 and θ_4 are normal with mean $\log(0.4)$ and variance 1. We use mean zero Gaussian proposals for all components of θ. We choose the variances of these proposals so that the acceptance rate for all runs is near 30%.

Using the samples $\{\mathbf{x}_m\}_{m=0}^L$ thus constructed, we run the Metropolis algorithm with $h = \Delta t/i$ with $i = 1, 2, 3, 4$. For each choice of parameters Δt and h, we compute 10100

samples and discard the first 100. To compute the runner's trajectory at intermediate points, we use linear interpolation between times t_m and t_{m+1}. We tabulate the results in Table 5.3; each value of γ_1 represents the mean of e^{θ_1} over all Metropolis samples of θ_1.

Parameters	γ_1	γ_2	ν_1	ν_2	RMSE
$\Delta t = 0.1; h = 0.1/1$	0.301	0.748	0.124	0.088	0.136
$\Delta t = 0.1; h = 0.1/2$	0.311	0.956	0.124	0.085	0.051
$\Delta t = 0.1; h = 0.1/3$	0.307	1.011	0.117	0.080	0.050
$\Delta t = 0.1; h = 0.1/4$	0.308	1.025	0.120	0.082	0.050
$\Delta t = 0.2; h = 0.2/1$	0.306	0.650	0.142	0.114	0.181
$\Delta t = 0.2; h = 0.2/2$	0.310	0.877	0.137	0.119	0.077
$\Delta t = 0.2; h = 0.2/3$	0.309	1.015	0.112	0.084	0.050
$\Delta t = 0.2; h = 0.2/4$	0.304	1.019	0.111	0.085	0.053
$\Delta t = 0.4; h = 0.4/1$	0.292	0.514	0.188	0.201	0.254
$\Delta t = 0.4; h = 0.4/2$	0.312	0.960	0.177	0.177	0.063
$\Delta t = 0.4; h = 0.4/3$	0.307	0.987	0.124	0.144	0.053
$\Delta t = 0.4; h = 0.4/4$	0.303	1.014	0.145	0.113	0.049

Table 5.3: Simulation Test Results for Stochastic Pursuit Model. We study the effect of decreasing h for simulated data generated with different inter-observation times Δt.

Overall, the results show that our algorithm produces mean posterior estimates that are reasonably close to the ground truth values. When the spacing of the data Δt is large, we see dramatic improvement when we use the DTQ method and more internal time steps. For instance, when $\Delta t = 0.4$, the RMS error improves dramatically from 0.254 to 0.049 as we decrease h, i.e., as we take more internal DTQ steps. Similar trends can be seen for the mean estimates of γ_2, ν_1 and ν_2.

NBA Tracking Data

We now turn to real tracking data taken from the game played between the Golden State Warriors and the Sacramento Kings on October 29, 2014. Reviewing this game, we found a fast break where Stephen Curry (of the Warriors) was the runner and Ramon Sessions (of the Kings) was the chaser. The entire fast break lasts 4.12 seconds. The spatial tracking data is recorded at intervals of 0.04 seconds, for a total of 104 observations. The tracking data uses the position on a court of dimension 94×50. We have rescaled the data to lie in a square with center $(0,0)$ and side length equal to one.

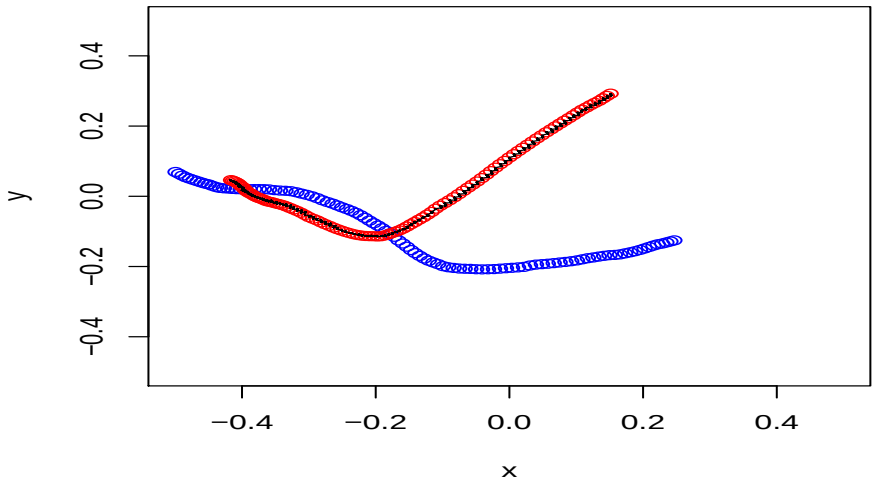

Figure 5.1: The agreement between the black curve (mean of simulated stochastic pursuit trajectories using MAP estimated parameters) and the red curve (chaser's trajectory) shows that the stochastic pursuit model is appropriate. The runner's trajectory is given in blue.

To parameterize the chaser's speed $\gamma(t)$, we have used a piecewise constant approximation with 8 equispaced pieces. Combined with the diffusion constants ν_1 and ν_2, this yields a 10-dimensional parameter vector θ. As in the previous simulated data test, we set the true parameters γ_i and ν_i to be the exponentials of the corresponding elements of the θ vector.

For the Metropolis sampler, the priors and proposals are higher-dimensional versions of those described in the simulated data test above. The main difference is that we now generate only 1000 post-burnin samples.

Using the Metropolis samples, we compute a kernel density estimate of each parameter. We then treat the mode of each computed density as the MAP (maximum a posteriori) estimate of the corresponding parameter. We then use the MAP estimates of the parameters in the pursuit SDE (5.3). We generate 100 sample paths of this SDE using the Euler-Maruyama method with time step 10^{-4}. As shown in Figure 5.1, the mean of these sample paths (plotted in black) agrees very well with the chaser's trajectory (plotted in red). This gives evidence that our stochastic pursuit system is an appropriate model for NBA fast breaks involving one runner and one chaser.

To visualize the insight provided by the model, we plot in Figure 5.2 the MAP estimated $\gamma(t)$ function over the time period of the fast break, $0 \leq t \leq 4.12$. The speed

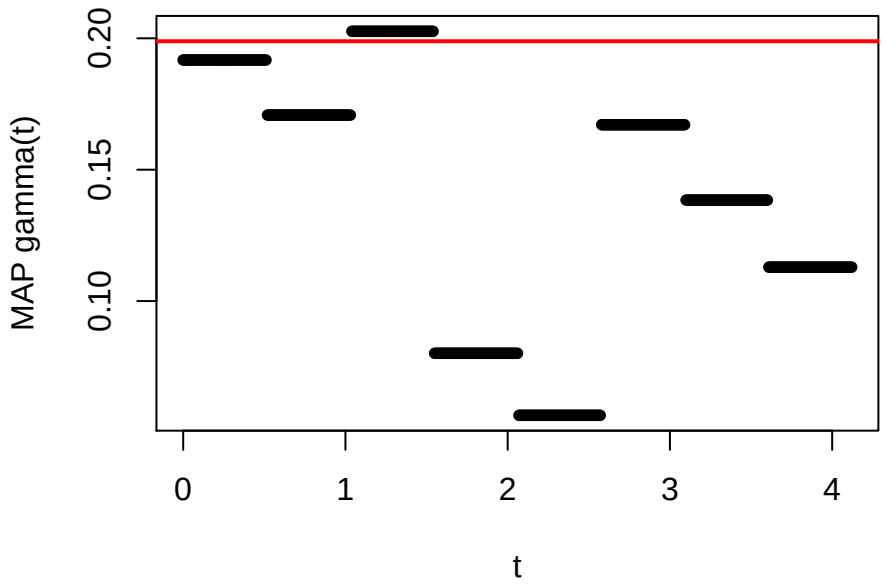

Figure 5.2: For the fast break tracking data described in the text, we plot the MAP estimate of the chaser's $\gamma(t)$ in black. This gives an estimate of how well the chaser is pursuing the runner at various points in time. The times at which γ dips below 0.1 correspond well to the times where Sessions was not able to keep up with Curry.

$\gamma(t)$ is the piecewise constant function plotted in black, while the mean speed computed directly from the data is given by a red horizontal line. The inferred speed shows that the chaser slows down dramatically approximately 1.5 seconds into the fast break. If one reviews the video footage of the play, this corresponds to the runner confusing the chaser and evading him.

Given our stochastic pursuit model's success in fitting the real data, in future work, we seek to apply the same methodology to a much larger sample of fast breaks. In this way, we can quantify a runner's ability to evade a chaser and/or a chaser's ability to stay near a runner who is actively trying to score. Note that all codes and data used in this work are available online at: `https://github.com/hbhat4000/sdeinference`.

5.4 Discussion and Conclusion

In this chapter, we developed an MCMC algorithm to perform Bayesian inference for coupled SDE. The MCMC algorithm is derived using a Metropolis scheme that uses likelihood information computed from the DTQ method. Using simulated tests of a stochastic van der Pol oscillator, we showed that the method is capable of computing accurate Bayesian estimates. We also compared the model performance against Bayesian filtering/inference method implemented in the R package "pomp", and the Eulerian method. Next, we derived a stochastic pursuit model for fast break situations involve one runner and one chaser in a basketball game. Then we demonstrated the performance of our Bayesian inference method using simulated tests of the derived stochastic model. More importantly, we demonstrated that the derived stochastic system is appropriate to model fast breaks through the Bayesian inference from NBA tracking data.

Chapter 6

Conclusions and Future Work

6.1 Summary

Given the widespread use of SDE to model dynamical systems in a diverse range of fields, solving the parameter inference problem is of greater importance. Over the course of this dissertation, we made a number of original contributions to the existing literature on parameter inference of SDE. A key bottleneck in the inference problem for SDE's, identified by many authors, is the evaluation of the conditional density given the parameters, i.e., the likelihood function for the SDE. We addressed this problem through the use of a novel, fast method to track the probability density function of the SDE. The method does not rely on sampling; instead, it uses repeated quadrature to track the time evolution of the transition density, hence the name DTQ. We also established theoretical convergence results for the DTQ method. In particular, for a class of SDE, we showed that the DTQ method converges to the true probability density function in L^1 norm error with a linear rate of convergent in temporal step size. Through a number of numerical tests, we confirmed that the empirical performance of the DTQ method complies with the theoretical convergent result. We then used the DTQ method to compute MLE for scalar SDE. We compute the MLE using derivative information computed using two different methods: the direct method and adjoint method. The direct method was used to solve the parametric inference problem of SDE. We used the adjoint method to solve the nonparametric inference problem as it enables the inference to scale well as the dimensionality of the parameter vector grows. We also generalized the DTQ method for

coupled SDE with time-dependent coefficients. Together with the likelihood computation from the DTQ method, we developed a Metropolis algorithm to perform Bayesian inference for models given by coupled SDE. Through a number of simulated tests and real data tests, we showed that the proposed methods are capable of solving the parameter inference problem of SDE quite successfully.

6.2 Future Work

The results of this thesis lead to several interesting directions for future work:

- The performance of the DTQ method can be improved to achieve a higher order of convergence. This can be accomplished by replacing the Euler-Maruyama method with a higher-order method. The only change is to then replace the Gaussian kernel G with a different conditional density function. Preliminary results with the weak trapezoidal method [3] indicate that we can obtain an $O(h^2)$ convergence with this new version of the DTQ method. In our work, we observed that the performance of the DTQ method becomes more sensitive to the internal time step h when solving the parameter inference problem of 2-D SDE. Therefore, it is very important to have a version of the DTQ method with a higher-order convergence rate to infer parameters of higher dimensional SDE. In addition, we also propose to generalize the current DTQ framework for higher dimensional SDE using sparse-grid quadrature.

- We can generalize the DTQ method for SDE driven by stochastic processes other than the Wiener process. In preliminary work, we have studied how to derive such methods to solve for the density in the case when we replace dW_t by a process whose increments follow a Lévy α-stable distribution. For such an SDE, current methods for computing the density involve numerical solution of a fractional Fokker-Planck equation. We expect DTQ-like methods to be highly competitive for such problems. Increments of Lévy α-stable processes do not have a closed-form density function. Therefore, we propose to generalize the DTQ method to track the characteristic function of the SDE and then use the Fourier transform to compute the density function.

- Current DTQ method can also be generalized to do inference on stochastic delay differential equations. In one of our work, we have generalized the DTQ method to compute the density of a specific nonlinear stochastic delay system [10]. This system arises as a switch-type control model for human balance. In this work, we capture the covariance of the solution at the present and delayed times through the time-evolution of a Gaussian approximation of the joint density at the present and delayed times. We propose to increase the accuracy of the method by replacing the Gaussian approximation with a sum of two-dimensional Gaussians, each centered at a different point.

- Throughout the thesis, we discussed doing inference for SDE from direct observations of the process X_t. In one of our work, we have considered the problem of Bayesian filtering and inference for time series data modeled as noisy, discrete-time observations of an SDE with undetermined parameters [11]. In this work, we developed a Metropolis algorithm to sample from the high-dimensional joint posterior density of all SDE parameters and state time series. In the Metropolis algorithm, we use the DTQ method to compute the likelihood of the SDE, the part of the posterior that requires the most computational effort to evaluate. We also showed that the DTQ method lends itself to a natural implementation using Scala and Apache Spark, an open source framework for scalable data mining. Then we studied the performance and scalability of our algorithm on filtering and inference problems for both regularly and irregularly spaced time series. We believe that we can derive large gains in performance by adapting our current algorithm to work in a streaming fashion. Specifically, instead of inferring the entire state series at once, as we currently do, we can proceed one step a time through the temporal sequence of observations.

Further improvements of and generalizations of the DTQ method described above will yield improved algorithms to solve parameter inference problem for a large class of SDE efficiently and effectively. In addition to the future work mentioned above, it is also remaining to conduct an extensive comparison of the DTQ method against the existing parameter inference methods in the literature.

Bibliography

[1] Yacine Aït-Sahalia. Maximum likelihood estimation of discretely sampled diffusions: A closed-form approximation approach. *Econometrica*, 70(1):223–262, 2002.

[2] Edward Allen. *Modeling with Itô stochastic differential equations*, volume 22. Springer Science & Business Media, 2007.

[3] D. F. Anderson and J. C. Mattingly. A weak trapezoidal method for a class of stochastic differential equations. *arXiv preprint arXiv:0906.3475*, 9(1):301–318, 2010.

[4] C. Archambeau, D. Cornford, M. Opper, and J. Shawe-Taylor. Gaussian process approximations of stochastic differential equations. *Journal of Machine Learning Research: Workshop and Conference Proceedings*, 1:1–16, 2007.

[5] C. Archambeau, M. Opper, Y. Shen, D. Cornford, and J. Shawe-Taylor. Variational inference for diffusion processes. In *Advances in Neural Information Processing Systems 20*, pages 17–24, 2007.

[6] Mostafa Bachar, Jerry J Batzel, and Susanne Ditlevsen. *Stochastic biomathematical models: with applications to neuronal modeling*, volume 2058. Springer, 2012.

[7] V. Bally and D. Talay. The law of the Euler scheme for stochastic differential equations. II. Convergence rate of the density. *Monte Carlo Methods and Applications*, 2(2):93–128, 1996.

[8] D. Barber, A. T. Cemgil, and S. Chiappa. *Bayesian Time Series Models*. Cambridge University Press, 2011.

[9] D. Bates and M. Maechler. *Matrix: Sparse and Dense Matrix Classes and Methods*, 2016. R package version 1.2-4.

[10] Harish S. Bhat and R. W. M. A. Madushani. Computing the density function for a nonlinear stochastic delay system. *IFAC-PapersOnLine*, 48(12):316–321, 2015. 12th IFAC Workshop on Time Delay Systems (TDS 2015), Ann Arbor, Michigan, USA, 28-30 June 2015.

[11] Harish S Bhat, RWMA Madushani, and Shagun Rawat. Scalable sde filtering and inference with apache spark. *Journal of Machine Learning Research W&CP*, 2016.

[12] R. N. Bhattacharya and E. C. Waymire. *Stochastic Processes with Applications*, volume 61 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2009.

[13] California Air Resources Board. Database: California Air Quality Data. http://www.arb.ca.gov/aqd/aqdcd/aqdcddld.htm, 2016.

[14] M. Braun. trustoptim: Trust region optimization for nonlinear functions with sparse hessians. http://cran.r-project.org/web/packages/trustOptim/, 2015.

[15] Y. Cai. Convergence theory of a numerical method for solving the Chapman-Kolmogorov equation. *SIAM Journal on Numerical Analysis*, 40(6):2337–2351, 2003.

[16] T. Canor and V. Denoël. Transient Fokker-Planck-Kolmogorov equation solved with smoothed particle hydrodynamics method. *International Journal for Numerical Methods in Engineering*, 94:535–553, 2013.

[17] S. Chen, E. Grant, T. T. Wu, and F. D. Bowman. Statistical learning methods for longitudinal high-dimensional data. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(1):10–18, 2014.

[18] Zhe Chen. Bayesian filtering: From Kalman filters to particle filters, and beyond. Technical report, Adaptive Systems Lab, McMaster University, Hamilton, ON, Canada, 2003.

[19] Loren Cobb. Stochastic differential equations for the social sciences. *Mathematical frontiers of the social and policy sciences*, (54):37, 1981.

[20] John C Cox, Jonathan E Ingersoll Jr, and Stephen A Ross. An intertemporal general equilibrium model of asset prices. *Econometrica: Journal of the Econometric Society*, pages 363–384, 1985.

[21] John C Cox, Jonathan E Ingersoll Jr, and Stephen A Ross. A theory of the term structure of interest rates. *Econometrica: Journal of the Econometric Society*, pages 385–407, 1985.

[22] Jason A Duan, Alan E Gelfand, CF Sirmans, et al. Modeling space-time data using stochastic differential equations. *Bayesian Analysis*, 4(4):733–758, 2009.

[23] D. Eddelbuettel. *Seamless R and C++ Integration with Rcpp*. Springer, New York, 2013.

[24] D. Eddelbuettel and R. François. Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18, 2011.

[25] D. Eddelbuettel and C. Sanderson. RcppArmadillo: Accelerating R with high-performance C++ linear algebra. *Computational Statistics and Data Analysis*, 71:1054–1063, March 2014.

[26] C. Fuchs. *Inference for Diffusion Processes: With Applications in Life Sciences*. Springer, Berlin, 2013.

[27] M. B. Giles, T. Nagapetyan, and K. Ritter. Multilevel Monte Carlo approximation of distribution functions and densities. *SIAM/ASA J. Uncertainty Quantification*, 3(1):267–295, 2015.

[28] Alison Gray, David Greenhalgh, L Hu, Xuerong Mao, and Jiafeng Pan. A stochastic differential equation sis epidemic model. *SIAM Journal on Applied Mathematics*, 71(3):876–902, 2011.

[29] Y. Hu and S. Watanabe. Donsker delta functions and approximations of heat kernels by the time discretization method. *J. Math. Kyoto Univ.*, 36:494–518, 1996.

[30] A. S. Hurn, J. I. Jeisman, and K. A. Lindsay. Seeing the wood for the trees: A critical evaluation of methods to estimate the parameters of stochastic differential equations. *Journal of Financial Econometrics*, 5(3):390–455, 2007.

[31] S. M. Iacus. *Simulation and Inference for Stochastic Differential Equations: With R Examples*. Springer Series in Statistics. Springer, New York, 2009.

[32] S. G. Johnson. The nlopt nonlinear-optimization package. http://ab-initio.mit.edu/nlopt, 2016.

[33] Motoo Kimura. Diffusion models in population genetics. *Journal of Applied Probability*, 1(2):177–232, 1964.

[34] A. A. King, D. Nguyen, E. L. Ionides, et al. Statistical inference for partially observed Markov processes via the R package pomp. *Journal of Statistical Software*, 69, 2016.

[35] A. Kohatsu-Higa. High order Ito-Taylor approximations to heat kernels. *J. Math. Kyoto Univ.*, 37:129–150, 1997.

[36] S. C. Kou, B. P. Olding, M. Lysy, and J. S. Liu. A multiresolution method for parameter estimation of diffusion processes. *Journal of the American Statistical Association*, 107(500):1558–1574, 2012.

[37] R. Küffner, N. Zach, R. Norel, J. Hawe, D. Schoenfeld, L. Wang, G. Li, L. Fang, L. Mackey, O. Hardiman, M. Cudkowicz, A. Sherman, G. Ertaylan, M. Grosse-Wentrup, T. Hothorn, J. van Ligtenberg, J. H. Macke, T. Meyer, B. Schoelkopf, L. Tran, R. Vaughan, G. Stolovitzky, and M. L. Leitner. Crowdsourced analysis of clinical trial data to predict amyotrophic lateral sclerosis progression. *Nature Biotechnology*, 33(1):51–U292, 2015.

[38] H. J. Kushner. On the weak convergence of interpolated Markov chains to a diffusion. *Ann. Probability*, 2:40–50, 1974.

[39] D. Y. Lin and Z. Ying. Semiparametric and nonparametric regression analysis of longitudinal data. *Journal of the American Statistical Association*, 96(453):103–13, 2001.

[40] J. Lund and K. L. Bowers. *Sinc methods for quadrature and differential equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.

[41] X. Luo and S. S.-T. Yau. Hermite spectral method to 1D forward Kolmogorov equation and its application to nonlinear filtering problems. *IEEE Transactions on Automatic Control*, 58(10):2495–2507, 2013.

[42] Maria C Mariani and Osei K Tweneboah. Stochastic differential equations applied to the study of geophysical and financial time series. *Physica A: Statistical Mechanics and its Applications*, 443:170–178, 2016.

[43] G. N. Milstein, J. G. M. Schoenmakers, and V. Spokoiny. Transition density estimation for stochastic differential equations via forward-reverse representations. *Bernoulli*, 10(2):281–312, 2004.

[44] H. Müller and F. Yao. Empirical dynamics for longitudinal data. *The Annals of Statistics*, 38(6):3458–3486, 2010.

[45] J. Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35:773–782, 1980.

[46] J. Nocedal and D. C. Liu. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[47] M. F. M. Osborne. Brownian motion in the stock market. *Operations Research*, 7(2):145–173, 1959.

[48] M. Di Paola and A. Sofi. Approximate solution of the Fokker-Planck-Kolmogorov equation. *Probabilistic Engineering Mechanics*, 17:369–384, 2002.

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

[50] U. Picchini. Inference for SDE models via approximate Bayesian computation. *Journal of Computational and Graphical Statistics*, 23(4):1080–1100, 2014.

[51] L. Pichler, A. Masud, and L. A. Bergman. Numerical solution of the Fokker-Planck equation by finite differences and finite element methods–a comparative study. In *Computational Methods in Stochastic Dynamics*, volume 2, pages 69–85. Springer, 2013.

[52] L. C. G. Rogers. Smooth transition densities for one-dimensional diffusions. *Bull. London Math. Soc.*, 17(2):157–161, 1985.

[53] A. Ruttor, P. Batz, and M. Opper. Approximate Gaussian process inference for the drift function in stochastic differential equations. In *Advances in Neural Information Processing Systems 26*, pages 2040–2048, 2013.

[54] C. Sanderson and R. Curtin. Armadillo: a template-based C++ library for linear algebra. *Journal of Open Source Software*, 1:26, 2016.

[55] P. Santa-Clara. Simulated likelihood estimation of diffusions with an application to the short term interest rate. Technical Report 12-97, Anderson School of Management, UCLA, Los Angeles, California, 1997.

[56] R. J. Sela and J. S. Simonoff. RE-EM trees: a data mining approach for longitudinal and clustered data. *Machine Learning*, 86:169–207, 2012.

[57] Kazimierz Sobczyk. *Stochastic differential equations: with applications to physics and engineering*, volume 40. Springer Science & Business Media, 2013.

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

[59] F. Stenger. *Numerical Methods Based on Sinc and Analytic Functions*. Springer Series in Computational Mathematics. Springer,New York, 2012.

[60] I. H. Stevenson and K. P. Kording. How advances in neural recording affect data analysis. *Nature Neuroscience*, 14(2):139–142, 201.

[61] K. Svanberg. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal on Optimization*, 12(2):555–573, 2002.

[62] L. N. Trefethen and J. A. C. Weideman. The exponentially convergent trapezoidal rule. *SIAM Review*, 56(3):385–458, 2014.

[63] Oldrich Vasicek. An equilibrium characterization of the term structure. *Journal of financial economics*, 5(2):177–188, 1977.

[64] M. D. Vrettas, M. Opper, and D. Cornford. Variational mean-field algorithm for efficient inference in large systems of stochastic differential equations. *Physical Review E*, 91(012148), 2015.

[65] J.-L. Wang. Nonparametric regression analysis of longitudinal data. In *Encyclopedia of Biostatistics*. Wiley, New York, NY, 2005.

[66] N. Wiener. *The Fourier Integral and Certain of Its Applications*. Cambridge Mathematical Library. Cambridge University Press, Cambridge, UK, 1988.

[67] W-K. Wong and R. B. Miller. Repeated time series analysis of ARIMA-noise models. *Journal of Business and Economic Statistics*, 8(2):243–250, 1990.

[68] J. Ypma. nloptr: R interface to nlopt. http://cran.r-project.org/web/packages/nloptr/, 2014.