April 2023

DATA-DRIVEN COMPUTATIONAL METHODS FOR QUASI-STATIONARY DISTRIBUTION AND SENSITIVITY ANALYSIS

Yaping Yuan

DATA-DRIVEN COMPUTATIONAL METHODS FOR QUASI-STATIONARY DISTRIBUTION AND SENSITIVITY ANALYSIS

A Dissertation Presented

by

YAPING YUAN

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

February 2023

Mathematics & Statistics

DATA-DRIVEN COMPUTATIONAL METHODS FOR QUASI-STATIONARY DISTRIBUTION AND SENSITIVITY ANALYSIS

A Dissertation Presented

by

YAPING YUAN

Approved as to style and content by:

Yao Li, Chair

Matthew Dobson, Member

Brian Van Koten, Member

Jianhan Chen, Member

Tom Braden, Graduate Program Director
Mathematics & Statistics

DEDICATION

To my family.

ACKNOWLEDGMENTS

I would like to express my most sincere gratitude to my advisor, Professor Yao Li, who has taught me an enormous amount of mathematics and gave me invaluable guidance for my thesis work. This thesis would would be impossible without his consistent support and inspiring guidance. It was an honor to be his student. I am also thankful to Professor Hongkun Zhang, who has provided tremendous help me during my PhD study.

My thankfulness is also to my parents for their endless support and encouragement through my study. My special thanks go to my husband, Bintian. Throughout the ups and downs of life, you never left my side, thank you for being my rock. Last but not least, I want to thank my daughter, Yuyan, who makes my life colorful and meaningful.

ABSTRACT

DATA-DRIVEN COMPUTATIONAL METHODS FOR QUASI-STATIONARY DISTRIBUTION AND SENSITIVITY ANALYSIS

FEBRUARY 2023

YAPING YUAN

B.Sc., INNER MONGOLIA UNIVERSITY

M.Sc., INNER MONGOLIA UNIVERSITY

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Yao Li

The goal of the dissertation is to develop the computational methods for quasi-stationary-distributions(QSDs) and the sensitivity analysis of a QSD against the modification of the boundary conditions and against the diffusion approximation.

Many models in various applications are described by Markov chains with absorbing states. For example, any models with mass-action kinetics, such as ecological models, epidemic models, and chemical reaction models, are subject to the population-level randomness called the demographic stochasticity, which may lead to extinction in finite time. There are also many dynamical systems that have interesting short term dynamics but trivial long term dynamics, such as dynamical systems with transient chaos [28]. A common way of capturing asymptotical properties of these transient dynamics is the quasi-stationary distribution (QSD), which is the conditional limiting distribution conditioning on not hitting the absorbing set yet. However, most QSDs do not have a closed form. So numerical solutions are necessary in various applications.

This dissertation studies computational methods for quasi-stationary distributions (QSDs). We first proposed a data-driven solver that solves Fokker-Planck equations for QSDs. Motivated by the case of Fokker-Planck equations for invariant probability measures, we set up an optimization problem that minimizes the distance from a low-accuracy reference solution, under the constraint of satisfying the linear relation given by the discretized Fokker-Planck operator. Then we use coupling method to study the sensitivity of a QSD against either the change of boundary condition or the diffusion coefficient. The 1-Wasserstein distance between a QSD and the corresponding invariant probability measure can be quantitatively estimated. Some numerical results about both computation of QSDs and their sensitivity analysis are provided.

This dissertation also studies the sensitivity analysis of mass-action systems against their diffusion approximations, particularly the dependence on population sizes. As a continuous time Markov chain, a mass-action system can be described by a equation driven by finite many Poisson processes, which has a diffusion approximation that can be pathwisely matched. The magnitude of noise in mass-action systems is proportional to the square root of the molecular count/population, which makes a large class of mass-action systems have quasi-stationary distributions (QSDs) instead of invariant probability measures. In this thesis we modify the coupling based technique developed in [15] to estimate an upper bound of the 1-Wasserstein distance between two QSDs. Some numerical results for sensitivity with different population sizes are provided.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

Many models in various applications are described by Markov chains with absorbing states. For example, any models with mass-action kinetics, such as ecological models, epidemic models, and chemical reaction models, are subject to the population-level randomness called the demographic stochasticity, which may lead to extinction in finite time. There are also many dynamical systems that have interesting short term dynamics but trivial long term dynamics, such as dynamical systems with transient chaos [28]. A common way of capturing asymptotical properties of these transient dynamics is the quasi-stationary distribution (QSD), which is the conditional limiting distribution conditioning on not hitting the absorbing set yet. However, most QSDs do not have a closed form. So numerical solutions are necessary in various applications.

Computational methods for QSDs are not very well developed. Although the relation between QSD and the Fokker-Planck equation is well known, it is not easy to use classical PDE solver to solve QSDs because of the following two reasons. First a QSD is the eigenfunction of the Fokker-Planck operator whose eigenvalue is unknown. The cost of solving eigenfunction of a discretized Fokker-Planck operator is considerably high. Secondly the boundary condition of the Fokker-Planck equation is unknown. We usually have a mixture of unbounded domain and unknown boundary value at the absorbing set. There are some literatures about Monte Carlo sampling from a QSD, which usually include regenerating samples from an empirical distribution once hitting the boundary [5, 6, 8, 9, 43]. However the efficiency of the Monte Carlo simulation is known to be low. To get the probability density function, one needs to deal with undesired noise associated to the Monte Carlo simulation.

Methods like the kernel density estimator and the variable kernel density estimation can smooth the solution but also introduce undesired diffusions to the solution, especially when a QSD is highly concentrated at the vicinity of some low-dimensional sets.

We are particularly interested in studying the dynamics of mass-action networks, as many of them admits QSDs instead of steady states. A mass-action network is a system of finite many species and reactions whose rule of update satisfies the mass-action law. Mass-action network covers a large number of chemical reaction network, epidemiology models, and population models. At the molecular level, reactions in the mass-action network are random events that modify the state of the network according to the stoichiometric equations. The time of these random events satisfy mass-action laws. Therefore, a mass-action network can be mathematically described by a continuous-time Markov process, which is driven by finite many Poisson processes.

The randomness in updating the network is called the demographic noise in population and epidemiology models. It is well known that demographic noise leads to finite time extinction in a very large class of population models (see for example the discussion in Section 5.1). This is because the magnitude of the demographic noise is proportional to the population size. As a result, when the population is small, in many mass-action systems, the noise could become the dominate term and leads to finite time extinction with strictly positive probability. Therefore, the asymptotic property of the mass-action network with finite time extinction is usually described by the quasi-stationary distribution (QSD), which is the conditional limiting distribution conditioning on not hitting the absorbing set yet. As discussed in [32], when the extinction rate is low, the quasi-stationary distribution can be well approximated by the invariant probability measure of a modified process that artificially ”pushes” the trajectory away from the extinction.

It has been known for decades that when the population size is large, the continuous-time Markov process converges to the mass-action ordinary differential equations (ODEs). In addition, by setting up a martingale problem, one can show that the re-scaled difference between the continuous-time Markov process and the mass-action ODE converges to a stochastic differential equation. Therefore, at any finite time, the continuous-time Markov process of a mass-action network is approximated by a stochastic differential equation. This is called the diffusion approximation of a mass-action network. We refer [2, 18] for further details.

The first goal of this dissertation is to extend the data-driven Fokker-Planck solver [14] to the case of QSDs . Similar to [14], we need a reference solution $\mathbf{v}$ generated by the Monte Carlo simulation. Then we discretize the Fokker-Planck operator in a numerical domain D without the boundary condition. Because of the lack of boundary conditions, the discretization only gives an underdetermined linear system, denoted by $\mathbf{A}\mathbf{u} = 0$. Then we minimize $\|\mathbf{v} - \mathbf{u}\|$ in the null space of $\mathbf{A}$. As shown in [15], this optimization problem projects the error terms of $\mathbf{v}$ to a low dimensional linear subspace, which significantly reduces its norm. Our numerical simulations show that this data-driven Fokker-Planck solver can tolerate very high level of spatially uncorrelated error, so the accuracy of $\mathbf{v}$ does not have to be high. The main difference between QSD solver and the Fokker-Planck solver is that we need a killing rate to find the QSD, which is obtained by a Monte Carlo simulation. We find that the QSD is not very sensitive against small error in the estimation of the killing rate.

The second goal of this dissertation is to study the sensitivity of QSDs. We are interested in two different kinds of sensitivities of QSDs, the one against modifying boundary condition and the one against diffusion approximation. Some modifications of either the boundary condition or the model parameters can prevent the Markov process from hitting the absorbing state in finite time. So the modified process would admit an invariant probability measure instead of a QSD. It is important to understand the difference between the QSD of a Markov process and the invariant probability measure of its modification. For example, many ecological models do not consider demographic noise because the population size is large and the QSD is much harder to study. But would the demographic noise completely change the asymptotical dynamics? More generally, a QSD captures the transient dynamics of a stochastic differential equation. If we separate a domain from the global dynamics by imposing reflecting boundary condition, how would the local dynamics be different from the corresponding transient dynamics? All these require some sensitivity analysis with quantitative bounds. Our approach is to regenerate from QSD after exiting. The process with regeneration admits the QSD as its invariant probability measure. So the approach of sensitivity analysis for invariant measure can be applied.

The last goal of this dissertation is to study the sensitivity of QSDs against the diffusion approximation. We are interested in how the QSD of the Markov process and its diffusion approximation differs from each other. The motivation is that an exact simulation at the molecular level is usually computationally expensive even if the stochastic simulation algorithm (SSA) is implemented optimally [19, 30, 40]. It is even harder to numerically compute the QSD when the number of molecules is large. On the other hand, the simulation of a diffusion process is much easier. The technique of computing the invariant probability measure or QSD of a stochastic differential equation is also well developed [14, 45]. Hence it is important to have a quantitative upper bound of the difference between the QSD of a mass-action system and that of its diffusion approximation. The way of sensitivity analysis is developed from on the coupling-based method in [15]. We need both finite time truncation error and the rate of contraction of the transition kernel of the diffusion process. The finite time error is given by the KMT algorithm in [36]. With the explicit construction of coupled trajectories of the Poisson process and the diffusion process, the finite time error up to fixed time T can be computed. The rate of contraction is modified from the data-driven method proposed in [31]. We design a suitable coupling scheme for the modified diffusion process that regenerates from the QSD right after hitting the absorbing set. Because of the coupling inequality, the exponential tail of the coupling time can be used to estimate the rate of contraction. The sensitivity analysis is demonstrated by several numerical examples. Generally speaking, the distance between two processes is much larger for smaller volume (i.e., molecular count).

The organization of this dissertation is as follows. Beside a preliminary about Quasi-stationary-distribution(QSD), stochastic differential equations, connection to the Fokker-Planck equation, we also provide basic settings and properties about the mass-action systems, Poisson processes and diffusion processes in Chapter 2. Chapter 3 is about the first goal of this dissertation. We introduce the data-driven solver for QSD and some numerical results are provided as well. The sensitivity analysis of Quasi-stationary-distribution(QSD) against modification of boundary condition or diffusion coefficients are presented in Chapter 4. Chapter 5 is about applications of Quasi-stationary-distribution(QSD) in mass-action systems and the introduction to the algorithms for computing the finite time error and the rate of contraction in two different cases. All explicit expressions of Poisson process and Wiener process are shown in the appendix.

CHAPTER 2

PRELIMINARY

2.1 Quasi-stationary-distribution(QSD)

We first give definition of the QSD and the exponential killing rate λ of a Markov process with an absorbing state. Let $X = (X_t : t \geq 0)$ be a continuous-time Markov process taking values in a measurable space $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$. Let $P^t(x, \cdot)$ be the transition kernel of X such that $P^t(x, A) = \mathbb{P}[X_t \in A \,|\, X_0 = x]$ for all $A \in \mathcal{B}$. Now assume that there exists an absorbing set $\partial\mathcal{X} \subset \mathcal{X}$. The complement $\mathcal{X}^a := \mathcal{X} \backslash \partial\mathcal{X}$ is the set of allowed states.

The process X_t is killed when it hits the absorbing set, implying that $X_t \in \partial\mathcal{X}$ for all $t > \tau$, where $\tau = \inf\{t > 0 : X_t \in \partial\mathcal{X}\}$ is the hitting time of set $\partial\mathcal{X}$. Throughout this thesis, we assume that the process is almost surely killed in finite time, i.e. $\mathbb{P}[\tau < \infty] = 1$.

For the sake of simplicity let $\mathbb{P}_x$ (resp. $\mathbb{P}_\mu$) be the probability conditioning on the initial condition $x \in \mathcal{X}$ (resp. the initial distribution μ).

Definition 2.1.1. *A probability measure μ on $\mathcal{X}^a$ is called a quasi-stationary distribution(QSD) for the Markov process X_t with an absorbing set $\partial\mathcal{X}$, if for every measurable set $C \subset \mathcal{X}^a$*

$$\mathbb{P}_\mu[X_t \in C | \tau > t] = \mu(C), \ t \geq 0, \tag{2.1.1}$$

If there is a probability measure μ exists such that

$$\lim_{t \to \infty} \mathbb{P}_x[X_t \in C | \tau > t] = \mu(C), \ \forall x \in \mathcal{X}^a. \tag{2.1.2}$$

in which case we also say that μ is a quasi-limiting distribution(QLD).

Remark 2.1.1. *When μ satisfies (2.1.2), it is called a quasi-limiting distribution (QLD), or a Yaglom limit. A QLD must be a QSD. Under some mild ergodicity assumptions, a QSD is also a QLD [11].*

In the analysis of QSD, we are particularly interested in a parameter λ, called the killing rate of the Markov process. If the distribution of the killing time $\mathbb{P}_x(\tau > t)$ has an exponential tail, then λ is the rate of this exponential tail. The following theorem shows that the killing time is exponentially distributed when the process starts from a QSD [10].

Theorem 2.1.1. *Let μ be a QSD of stochastic process X. Then*

$$\exists\ \lambda = \lambda(\mu) \textit{ such that } \mathbb{P}_\mu[\tau > t] = e^{-\lambda t},\ \forall t \geq 0,$$

where λ is called the killing rate of X.

Throughout this thesis, we assume that X admits a QSD denoted by μ with a strictly positive killing rate λ.

2.2 Stochastic differential equations

2.2.1 Stochastic differential equations

Consider a stochastic differential equation (SDE)

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \sigma(X_t)\mathrm{d}W_t, \tag{2.2.1}$$

where $X_t \in \mathbb{R}^d$ and X_t is killed when it hits the absorbing set $\partial\mathcal{X} \subset \mathbb{R}^d$; $f : \mathbb{R}^d \to \mathbb{R}^d$ is a continuous vector field; σ is an $d \times d$ matrix-valued function; and $\mathrm{d}W_t$ is the white noise in $\mathbb{R}^d$. The following well known theorem shows the existence and the uniqueness of the solution of equation (2.2.1).

Theorem 2.2.1. *Assume that there are two positive constants C_1 and C_2 such that the two functions f and σ in (2.2.1) satisfy*
(1) (Lipschitz condition) for all $x, y \in \mathbb{R}^d$ and t

$$|f(x) - f(y)|^2 + |\sigma(x) - \sigma(y)|^2 \leq C_1|x - y|^2;$$

(2) (Linear growth condition) for all $x, y \in \mathbb{R}^d$ and t

$$|f(x)|^2 + |\sigma(x)|^2 \leq C_2(1 + |x|^2).$$

Then there exists a unique solution $X(t)$ to equation (2.2.1).

There are quite a few recent results about the existence and convergence of QSD. Since the theme of this thesis is numerical computations, in this thesis we directly assume that X_t admits a unique QSD μ on set $\mathcal{X}^a$ that is also the quasi-limit distribution. The detailed conditions are referred in [17, 24, 38, 41, 42].

2.2.2 Connection with Fokker-Planck equation

The Fokker-Planck equation is the equation governing the time evolution of the probability density of a stochastic process. For a stochastic differential equation(SDE)

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \sigma(X_t)\mathrm{d}W_t \ ,$$

the probability density u of the random variable X_t satisfies the Fokker-Planck equation

$$u_t = \mathcal{L}u = -\sum_{i=1}^{d}(f_i u)_{x_i} + \frac{1}{2}\sum_{i,j=1}^{d}(D_{ij}u)_{x_i x_j},$$

with diffusion tensor $D = \sigma\sigma^T$. $u(x, t)$ denotes the probability density at time t, and subscripts t and x_i denote the partial derivatives with respect to time t an location x. In this dissertation, we focus on the quasi-stationary-distribution (QSD), whose density function u satisfies [3]

$$-\lambda u = \mathcal{L}u = -\sum_{i=1}^{d}(f_i u)_{x_i} + \frac{1}{2}\sum_{i,j=1}^{d}(D_{ij}u)_{x_i x_j}, \tag{2.2.2}$$

where $D = \sigma^T\sigma$, and λ is the killing rate. The operator $\mathcal{L}$ is the infinitesimal generator.

For simplicity, we only consider cases when D_{ij} are constants in this dissertation.

2.2.3 Numerical Scheme

2.2.4 Basic settings

For simplicity, we introduce the algorithm for $n = 2$, specifically, we solve u in equation (2.2.2) numerically on a 2D domain $D = [a_0, b_0] \times [a_1, b_1]$. Firstly, we construct an $N \times M$ grid on D with grid size $h = \frac{b_0 - a_0}{N} = \frac{b_1 - a_1}{M}$. Each small box in the mesh is denoted by $O_{i,j} = [a_0 + (i-1)h, a_0 + ih] \times [a_1 + (j-1)h, a_1 + jh]$. Let $\mathbf{u} = \{u_{i,j}\}_{i=1,j=1}^{i=N,j=M}$ be the numerical solution on D that we are interested in, then $\mathbf{u}$ can be considered as a vector in $\mathbb{R}^{N\times M}$. Each element $u_{i,j}$ approximates the density function u at the center of each $O_{i,j}$, with coordinate $(ih + a_0 - h/2, jh + a_1 - h/2)$. We consider u as the solution to the boundary-free partial differential equation (2.2.2) and discretize the operator $\mathcal{L}$ on D with respect to all $(N-2)(M-2)$ interior boxes. The discretization of the Fokker-Planck equation with respect to each center point is given by the following Finite Difference Methods.

2.2.4.1 Finite difference method

In numerical analysis, finite difference method is a numerical technique for solving differential equations by approximating derivatives with finite difference. Finite difference representations of derivatives are derived from Taylor series expansions. The basic philosophy of finite difference methods is to replace the derivatives of the governing equations with algebraic difference quotients. This will result in a system of algebraic equations which can be solved for the dependant variables at the discrete grid points. Let $u_{i,j}$ approximates the density function u at the center of each $O_{i,j}$. Note that we assume $u_{i,j}$ represent a smooth function, meaning that we can differentiate the function several times and each derivative is well-defined bounded function over $O_{i,j}$. It is not hard to see that $u_{i,j}$ is immediately related to the function value at its right side $u_{i+1,j}$ and to the function value at its left side $u_{i-1,j}$. By Taylor series expansion, we have

$$u_{i+1,j} = u_{i,j} + \left(\frac{\partial u}{\partial x}\right)_{i,j} h + \left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} \frac{h^2}{2} + \left(\frac{\partial^3 u}{\partial x^3}\right)_{i,j} \frac{h^3}{6} + \cdots \tag{2.2.3}$$

If terms of order h^2 and higher are neglected, then equation (2.2.3) is reduced to

$$u_{i+1,j} \approx u_{i,j} + \left(\frac{\partial u}{\partial x}\right)_{i,j} h,$$

therefore, the **first-order forward** difference is defined as

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{u_{i+1,j} - u_{i,j}}{h} + O(h).$$

Similarly, the **first-order backward** difference is defined as

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{u_{i,j} - u_{i-1,j}}{h} + O(h),$$

And the **second-order central** difference for the derivative $\left(\frac{\partial u}{\partial x}\right)$ at the gird point (i, j) is defined as

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2h} + O(h^2).$$

For second-order partial derivatives, the **central difference of second derivative** is

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + O(h^2).$$

With all derivatives approximations above, we have the discretization of the Fokker-Planck operator $\mathcal{L}$, which is an $(N-2)(M-2) \times (NM)$ matrix. More precisely, row $(i-1) * (N-2) + j - 1$ is the discretization of $\mathcal{L}$ at the center of box (i, j), which is given by

$$-\frac{1}{2h}\left(f_{i+1,j}u_{i+1,j}-f_{i-1,j}u_{i-1,j}+f_{i,j+1}u_{i,j+1}-f_{i,j-1}u_{i,j-1}\right)+$$
$$\frac{1}{2h^2}\left(u_{i+1,j}+u_{i-1,j}+u_{i,j+1}+u_{i,j-1}-4u_{i,j}\right)=-\lambda u_{i,j}$$

2.2.4.2 Sampling

In order to compute the QSD, one needs to numerically simulate a long trajectory of X. Once X_t hits the absorbing state, a new initial value is sampled from the empirical QSD. The re-sampling step can be done in two different ways. We can either use many independent trajectories that form an empirical distribution [4] or re-sample from the history of a long trajectory [6]. In this dissertation we use the latter approach.

Let $\hat{X} = \{\hat{X}^\delta_n,\ n \in \mathbb{Z}_+\}$ be a stochastic process that samples the "numerical QSD". When $\hat{X}^\delta_n \in \mathcal{X}^a$, $\hat{X}$ simply approximates the time-δ sample chain of X_t, i.e., $X_{n\delta}$. The approximation uses either Euler-Maruyama scheme or the Milstein scheme. The Euler-Maruyama numerical scheme is given by

$$\hat{X}^\delta_{n+1} = \hat{X}^\delta_n + f(\hat{X}^\delta_n)\delta + \sigma(\hat{X}^\delta_n)(W_{(n+1)\delta} - W_{n\delta}), \tag{2.2.4}$$

where $\hat{X}^\delta_0 = X_0,\ W_{(n+1)\delta} - W_{n\delta} \sim \mathcal{N}(0, \delta \mathrm{Id}_d),\ n \in \mathbb{Z}_+$ is a d-dimensional normal random variable. A more accurate scheme is the Milstein scheme, which reads

$$\hat{X}^\delta_{n+1} = \hat{X}^\delta_n + f(\hat{X}^\delta_n)\delta + \sigma(\hat{X}^\delta_n)(W_{(n+1)\delta} - W_{n\delta}) + \sigma(\hat{X}^\delta_n)IL,$$

where I is a $d \times d$ matrix with its (i,j)-th component being the double It$\hat{o}$ integral

$$I_{i,j} = \int_{n\delta}^{(n+1)\delta}\int_{n\delta}^{s_2} dW^i(s_1)dW^j(s_2),$$

and $L \in \mathbb{R}^d$ is a vector of left operators with i-th component

$$u\mathcal{L}_i = \sum_{i=1}^{d} \sigma_{i,j}(\hat{X}_n^\delta)\frac{\partial u}{\partial x_i}.$$

Under suitable assumptions of Lipschitz continuity and linear growth conditions for f and σ, the Euler-Maruyama approximation provides a convergence rate of order 1/2, while the Milstein scheme is an order 1 strong approximation [25].

To deal with the situation when $\hat{X}$ hits the absorbing set, in addition to $\hat{X}_n^\delta$, we also need to update a temporal occupation measure

$$\mu_n = \frac{1}{n}\sum_{k=0}^{n-1} \delta_{\hat{X}_n^\delta}\,.$$

If the numerical scheme gives $\hat{X}_{n+1}^\delta \in \partial\mathcal{X}$, we immediately resample $\hat{X}_{n+1}^\delta$ from μ_n. More precisely, let the transition kernel of the numerical scheme of X (without resampling) be $\hat{Q}$. Then $\hat{Q}$ has an absorbing set, i.e., $\hat{Q}(\partial\mathcal{X}, \partial\mathcal{X}) = 1$. The transition kernel of $\hat{X}_n^\delta$ is modified from $\hat{Q}$ such that

$$\mathbb{P}[\hat{X}_{n+1}^\delta \in A \,|\, \hat{X}_n^\delta = x] = \hat{Q}(x, A) + \hat{Q}(x, \partial\mathcal{X})\mu_n(A)\,.$$

We have the following convergence result from [6].

Proposition 2.2.1 ((Theorem 2.5 in [6]))**.** *Under suitable assumptions about $\hat{X}$, the occupation measure μ_n converges to the QSD μ as $n \to \infty$.*

The assumption in Proposition 2.2.1 about $\hat{X}$ is that $\hat{X}$ must be Feller, the absorbing set is accessible, $\hat{X}$ admits a “weak small set”, and that the killing rates from different initial values are uniformly controlled. It is easy to check that these assumptions are satisfied by the numerical scheme of most ergodic SDEs. Since this thesis focuses on numerical algorithm, throughout this thesis, we assume that all assumptions in Proposition 2.2.1 are satisfied.

2.3 Monte-Carlo method for QSD

In line with the basic settings, we solve u in equation (2.2.2) numerically on a 2D domain $D = [a_0, b_0] \times [a_1, b_1]$. Firstly, we construct an $N \times M$ grid on D with grid size $h = \frac{b_0 - a_0}{N} = \frac{b_1 - a_1}{M}$. Each small box in the mesh is denoted by $O_{i,j} = [a_0 + (i-1)h, a_0 + ih] \times [a_1 + (j-1)h, a_1 + jh]$. Let $\mathbf{u} = \{u_{i,j}\}_{i=1,j=1}^{i=N,j=M}$ be the numerical solution on D that we are interested in, then $\mathbf{u}$ can be considered as a vector in $\mathbb{R}^{N \times M}$. Each element $u_{i,j}$ approximates the density function u at the center of each $O_{i,j}$, with coordinate $(ih + a_0 - h/2, jh + a_1 - h/2)$. Generally speaking, we count the number of $\hat{X}$ falling into each box and set the normalized value as the approximate probability density at $O_{i,j}$. The detail of the simulation is shown in Algorithm 1 below.

Algorithm 1 Monte Carlo for QSD

```
Input: Equation (2.2.4) and the grid
Output: A Monte Carlo approximation $\mathbf{u} = \{u_{i,j}\}$. Sample size $N_s$.
    Pick any initial value $X_0 \notin \partial\mathcal{X}$ in $D$
    for $n = 1$ to $N_s$ do
        Use $\hat{X}^\delta_n$ and equation (2.2.4) to compute $\hat{X}^\delta_{n+1}$
        Record the coordinates of the small box $O_{i,j}$ where $\hat{X}^\delta_{n+1}$ stands, say $i^*, j^*$
        if $\hat{X}^\delta_{n+1} \notin \partial\mathcal{X}$ then
            $u_{i^*,j^*} \leftarrow u_{i^*,j^*} + 1$
        else
            $\hat{X}^\delta_{n+1} = \hat{X}^\delta_{\lfloor U*n \rfloor}$, where $U$ is a uniformly distributed random variable
        end if
    end for
    Return $u_{i,j}/N_s h^2$ for all $i, j$ as the approximation solution.
```

Sometimes the Euler-Maruyama method underestimates the probability that X moves to the absorbing set, especially when X_t is close to $\partial\mathcal{X}$. This problem can be fixed by introducing the Brownian bridge correction. We refer to [6] for details. For a sample falling into a small box which are closed to $\partial\mathcal{X}$, the probability of them falling into the trap $\partial\mathcal{X}$ is relatively high. In fact, this probability is exponentially distributed and the rate is related to the distance from $\partial\mathcal{X}$. Let $B^T_t = W_t - \frac{t}{T} W_T$ be the Brownian Bridge on the interval $[0, T]$. In the 1D case, the law of the infimum and the supremum of the Brownian Bridge can be computed as follows: for every $z \geq \max(x, y)$

$$\mathbb{P}[\sup_{t\in[0,T]}(x + (y-x)\frac{t}{T} + \phi B_t^T) \leq z] = 1 - \exp(-\frac{2}{T\phi^2}(z-x)(z-y)), \tag{2.3.1}$$

where $x = \hat{X}_n^\delta \in \mathcal{X}^a, y = \hat{X}_{n+1}^\delta \in \mathcal{X}^a$, and $\phi = \sigma(\hat{X}_n^\delta)$ is the strength coefficient of Brownian Bridge. This means that at each step n, if $\hat{X}_{n+1}^\delta \in \mathcal{X}^a$, one can compute, with the help of the above properties, a Bernoulli random variable G with the parameter

$$p = \mathbb{P}[\exists t \in (n\delta, (n+1)\delta), \hat{X}_t \in \partial\mathcal{X} | x = \hat{X}_n^\delta, y = \hat{X}_{n+1}^\delta] \quad \text{(If } G = 1, \text{the process is killed).}$$

2.4 Coupling

2.4.1 Coupling Method

The coupling method is used for the sensitivity analysis of QSDs.

Definition 2.4.1. ***(Coupling of probability measures)*** *Let μ and ν be two probability measures on a probability space $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$. A probability measure γ on $(\mathcal{X}\times\mathcal{X}, \mathcal{B}(\mathcal{X})\times\mathcal{B}(\mathcal{X}))$ is called a coupling of μ and ν, if two marginals of γ coincide with μ and ν respectively.*

The definition of coupling can be extended to any two random variables that take value in the same state space. Now consider two Markov processes $X = (X_t : t \geq 0)$ and $Y = (Y_t : t \geq 0)$ with the same transition kernel P. A coupling of X and Y is a stochastic process (X^m, Y^m) on the product state space $\mathcal{X} \times \mathcal{X}$ such that

(i) The marginal processes X^m and Y^m are Markov processes with the transition kernel P;

(ii) If $X_s^m = Y_s^m$, we have $X_t^m = Y_t^m$ for all $t > s$.

The first meeting time of X_t^m and Y_t^m is denoted as $\tau^c := \inf_{t\geq 0}\{X_t^m = Y_t^m\}$, which is called the coupling time. The coupling (X^m, Y^m) is said to be successful if the coupling time is almost surely finite, i.e. $\mathbb{P}[\tau^c < \infty] = 1$. Here the super index m stands for the marginal distribution, which is dropped when it causes no confusion.

In order to give an estimate of the sensitivity of the QSD, we need the following two metrics.

Definition 2.4.2. *(Wasserstein distance) Let d be a metric on the metric state space V equipped with distance $d(\cdot,\cdot)$. For probability measures μ and ν on V, the Wasserstein distance between μ and ν for d is given by*

$$\begin{aligned} d_w(\mu,\nu) &= \inf\{\mathbb{E}_\gamma[d(x,y)] : \ \gamma \text{ is a coupling of } \mu \text{ and } \nu.\} \\ &= \inf\{\int d(x,y)\gamma(dx,dy) : \gamma \text{ is a coupling of } \mu \text{ and } \nu.\} \end{aligned} \tag{2.4.1}$$

In this thesis, without further specification, we assume that the 1-Wasserstein distance is induced by $d(x,y) = \min\{1, \|x-y\|\}$, where $\|x-y\|$ is the Euclidean norm.

Definition 2.4.3. *(Total variation distance) Let μ and ν be probability measures on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$. The total variation distance of μ and ν is*

$$\|\mu-\nu\|_{TV} := \sup_{C\in\mathcal{B}(\mathcal{X})} |\mu(C)-\nu(C)|.$$

Lemma 2.4.1. *(Coupling inequality) For the coupling given above and the Wasserstein distance induced by the distance given in (2.4.1), we have*

$$\mathbb{P}[\tau^c > t] = \mathbb{P}[X_t^m \neq Y_t^m] \geq d_w(P^t(x,\cdot), P^t(y,\cdot)).$$

Proof. By the definition of Wasserstein distance,

$$\begin{aligned} d_w(P^t(x,\cdot), P^t(y,\cdot)) &\leq \int d(x,y)\mathbb{P}[(X_t^m, Y_t^m) \in (dx,dy)] \\ &= \int_{x\neq y} d(x,y)\mathbb{P}[(X_t^m, Y_t^m) \in (dx,dy)] \\ &\leq \int_{x\neq y} \mathbb{P}[(X_t^m, Y_t^m) \in (dx,dy)] \\ &= \mathbb{P}[X_t^m \neq Y_t^m]. \end{aligned}$$

The lemma follows because $\mathbb{P}[\tau^c > t] = \mathbb{P}[X_t^m \neq Y_t^m]$ by definition. $\square$

Consider a Markov coupling $(\hat{X}, \hat{Y})$, where $\hat{X} = \{\hat{X}_n^\delta : n \in \mathbb{N}\}$ and $\hat{Y} = \{\hat{Y}_n^\delta : n \in \mathbb{N}\}$ are two numerical trajectories of the stochastic differential equation described in (2.2.4). Theoretically, there are many ways to make stochastic differential equations couple. But since numerical computation always has errors, two numerical trajectories may miss each other when the true trajectories couple. Hence we need to apply a mixture of the following coupling methods in practice.

Independent coupling. Independent coupling means the noise term in the two marginal processes $\hat{X}$ and $\hat{Y}$ are independent when running the coupling process $(\hat{X}, \hat{Y})$. That is

$$\begin{aligned} \hat{X}_{n+1}^\delta &= \hat{X}_n^\delta + f(\hat{X}_n^\delta)\delta + (W_{(n+1)\delta}^{(1)} - W_{n\delta}^{(1)}) \\ \hat{Y}_{n+1}^\delta &= \hat{Y}_n^\delta + f(\hat{Y}_n^\delta)\delta + (W_{(n+1)\delta}^{(2)} - W_{n\delta}^{(2)}), \end{aligned}$$

where $(W_{(n+1)\delta}^{(1)} - W_{n\delta}^{(1)})$ and $(W_{(n+1)\delta}^{(2)} - W_{n\delta}^{(2)})$ are independent random variables for each n.

Reflection coupling Two Wiener processes meet less often than the 1D case when the state space has higher dimensions. This fact makes the independent coupling less effective. The reflection coupling is introduced to avoid this case. Take the Euler-Maruyama scheme of the SDE

$$dX_t = f(X_t)dt + \sigma dW_t$$

as an example, where σ is a constant matrix. The Euler-Maruyama scheme of X_t reads as

$$\hat{X}_{n+1}^\delta = \hat{X}_n^\delta + f(\hat{X}_n^\delta)\delta + \sigma(W_{(n+1)\delta} - W_{n\delta}),$$

where W is a standard Wiener process. The reflection coupling means that we run $\hat{X}_n^\delta$ as

$$\hat{X}_{n+1}^\delta = \hat{X}_n^\delta + f(\hat{X}_n^\delta)\delta + \sigma(W_{(n+1)\delta} - W_{n\delta}),$$

while run $\hat{Y}_n^\delta$ as

$$\hat{Y}_{n+1}^\delta = \hat{Y}_n^\delta + f(\hat{Y}_n^\delta)\delta + \sigma P(W_{(n+1)\delta} - W_{n\delta}),$$

where $P = I - 2e_n e_n^T$ is a projection matrix with

$$e_n = \frac{\sigma^{-1}(\hat{X}_n^\delta - \hat{Y}_n^\delta)}{\|\sigma^{-1}(\hat{X}_n^\delta - \hat{Y}_n^\delta)\|}.$$

Nontechnically, reflecting coupling means that the noise term is reflected against the hyperplane that orthogonally passes the midpoint of the line segment connecting $\hat{X}_n^\delta$ and $\hat{Y}_n^\delta$. In particular, $e_n = -1$ when the state space is 1D.

Maximal coupling Above coupling schemes can bring $\hat{X}_n^\delta$ moves close to $\hat{Y}_n^\delta$ when running numerical simulations. However, a mechanism is required to make $\hat{X}_{n+1}^\delta = \hat{Y}_{n+1}^\delta$ with certain positive probability. That's why the maximal coupling is involved. One can couple two trajectories whenever the probability distributions of their next step have enough overlap. Denote $p^{(x)}(z)$ and $p^{(y)}(z)$ as the probability density functions of $\hat{X}_{n+1}^\delta$ and $\hat{Y}_{n+1}^\delta$ respectively. The implementation of the maximal coupling is described in the following algorithm.

Algorithm 2 Maximal coupling

```
Input: $\hat{X}_n^\delta$ and $\hat{Y}_n^\delta$
Output: $\hat{X}_{n+1}^\delta$ and $\hat{Y}_{n+1}^\delta$, and $\tau^c$ if coupled
  Compute probability density functions $p^{(x)}(z)$ and $p^{(y)}(z)$
  Sample $\hat{X}_{n+1}^\delta$ and calculate $r = Up^{(x)}(\hat{X}_{n+1}^\delta)$, where $U$ is uniformly distributed on [0,1]
  if $r < p^{(y)}(\hat{X}_{n+1}^\delta)$ then
    $\hat{Y}_{n+1}^\delta = \hat{X}_{n+1}^\delta, \tau^c = (n+1)\delta$
  else
    Sample $\hat{Y}_{n+1}^\delta$ and calculate $r' = Vp^{(y)}(\hat{Y}_{n+1}^\delta)$, where $V$ is uniformly distributed on [0,1]
    while $r' < p^{(x)}(\hat{Y}_{n+1}^\delta)$ do
      Resample $\hat{Y}_{n+1}^\delta$ and $V$. Recalculate $r' = Vp^{(y)}(\hat{Y}_{n+1}^\delta)$
    end while
    $\tau^c$ is still undetermined
  end if
```

For discrete-time numerical schemes of SDEs, we use reflection coupling when $\hat{X}_n^\delta$ and $\hat{Y}_n^\delta$ are far away from each other, and maximal coupling when they are sufficiently close. The threshold of changing coupling method is $2\sqrt{\delta}\|\sigma\|$ in our simulation, that is, the maximal coupling is applied when the distance between $\hat{X}^{\delta}_n$ and $\hat{Y}^{\delta}_n$ is smaller than the threshold.

2.5 Mass-action network

2.5.1 Stochastic mass reaction networks

We consider a mass action network of K reactions involving d distinct species, $S_1, \cdots, S_d$,

$$\sum_{i=1}^{d} c_{ki} S_i \to \sum_{i=1}^{d} c'_{ki} S_i, \ k = 1, \cdots, K \tag{2.5.1}$$

where c_{ki} and c'_{ki} are non-negative integers that denote the number of moleculars of species S_i consumed and produced by reaction k, respectively. Let V be the volume of the reaction system. Let $X(t) = (x_1(t), \cdots, x_d(t)) \in \mathbb{R}^d$ be the state of the mass action system at time t, such that the i-th entry of $X(t)$ represents the concentration of species S_i, $i = 1, \cdots, d$. In other words the number of moleculars of S_i is $Vx_i := N_i$. Let λ_k be the rate at which the kth reaction occurs, that is, it gives the propensity of the k-th reaction as a function of the concentrations of moleculars of the chemical species.

2.5.2 Rates for the law of mass action

The law of mass action means the rate of a reaction should be proportional to the number of distinct subsets of the participating moleculars. More precisely, the rate of reaction k reads

$$\lambda_k = \kappa_k V \prod_{i=1}^{d} (\frac{N_i}{V})^{c_{ki}} := V f_k(X),$$

where κ_k is a rate constant, and N_i be the number of molecular of the ith species in the system. Let $\Delta t \ll 1$ be a very short time period. More precisely, given all information of the system up to time t, we have

$$\mathbb{P}[\text{ reaction } k \text{ occurs in } [t, t + \Delta t)] = \lambda_k \Delta t + O(\Delta t^2)\,.$$

2.5.3 Poisson process

We use Poisson counting process to represent $X(t)$, because $X(t)$ is a continuous time discrete state Markov chain. Let $X_i(t)$ be i-th entry of $X(t)$, then

$$X_i(t) = X_i(0) + \frac{1}{V}\sum_k R_k(t)(c'_{ki} - c_{ki}),$$

where $R_k(t)$ is the number of times the reaction k has occurred by time t and $R_k(0) = 0$. Because the number of moleculars of species changes with time, $R_k(t)$ is an inhomogeneous Poisson process that is given by

$$R_k(t) = P_k(V\int_0^t f_k(X(s))ds), \tag{2.5.2}$$

where $P_k(\cdot)$ is a unit-rate Poisson point process. It is well known that $P_k(\cdot)$ satisfies the following three properties: (1) $P_k(0) = 0$, (2) $P_k(\cdot)$ has independent increments, and (3) $P_k(s+t) - P_k(s)$ is a Poisson random variable with parameter t. And the whole system is given by

$$X(t) = X(0) + \sum_k \frac{l_k}{V} P_k(V\int_0^t f_k(X(s))ds) \tag{2.5.3}$$

where $P_k(t),\ k = \{1, \cdots, K\}$ are independent unit-rate Poisson processes, and $l_k = c'_k - c_k \in \mathbb{R}^d$ denotes the coefficient change of moleculars at reaction k.

2.5.4 Diffusion process

When V is large, a Poisson process can be approximated by a diffusion process. The follow lemma in [26, 27] gives the strong approximation theorem for Poisson processes.

Lemma 2.5.1. *A unit Poisson process $P(\cdot)$ and a Wiener process $B(\cdot)$ can be constructed so that*

$$\left|\frac{P(Vt) - Vt}{\sqrt{V}} - \frac{1}{\sqrt{V}}B(Vt)\right| \le \frac{\log(Vt \vee 2)}{\sqrt{V}}\Gamma,$$

where Γ is a random variable such that $\mathbb{E}(e^{c\Gamma}) < \infty$ for some constant $c > 0$.

Remark 2.5.1. *By the scaling property of Wiener process,* $\frac{1}{\sqrt{V}}B(Vt)$ *is also a standard Wiener process.*

With the lemma above and Ito's formula, we have the diffusion approximation

$$\begin{aligned} P_k\left(V\int_0^t f_k(X(s))ds\right) &\approx V\int_0^t f_k(X(s))ds + \int_0^t \sqrt{Vf_k(X(s))}dB(s) \\ &= V\int_0^t f_k(X(s))ds + B_k\left(V\int_0^t f_k(X(s))ds\right) \end{aligned}$$

This gives the diffusion approximation of the mass action system $X(t)$:

$$Y(t) = Y(0) + \sum_k \frac{l_k}{V}\left[V\int_0^t f_k(Y(s))ds + B_k\left(V\int_0^t f_k(X(s))ds\right)\right].$$

In the chemical literature, Y is known as the Langevin approximation for the continuous time Markov chain model. Theoretically, the distance between these two approximations is bounded as follow theorem in [36].

Theorem 2.5.2. *Let* $X(t)$ *be a Poisson process represented by* (2.5.3)*, let* $Y(t)$ *be a diffusion process with initial condition satisfying* $X(0) = Y(0)$ *and solves the following stochastic differential equation*

$$Y(t) = Y(0) + \sum_k \frac{l_k}{V}\left[V\int_0^t f_k(Y(s))ds + B_k\left(V\int_0^t f_k(Y(s))ds\right)\right] \tag{2.5.4}$$

where the $B_k(\cdot)$ *are independent standard Wiener processes. As* $V \to \infty$*,*

$$\sup|X(t) - Y(t)| = O\left(\frac{\log V}{V}\right). \tag{2.5.5}$$

The error of diffusion approximation is proportional to $\frac{\log V}{V}$, which converges to 0 as $V \to \infty$. In macroscopic chemical reaction system V is at the magnitude of Avogadro's number. Therefore, the entire diffusion term can be safely ignored. However, in many ecologic systems or cellular chemical reaction systems, the effective volume cannot be simply treated as infinity. This motivates us to consider the sensitivity of the quasi-stationary-distributions (QSDs) against the diffusion approximation. For any finite capacity V, the finite time error of the diffusion approximation can be explicitly simulated. Paper [36] gives the constructive procedure to generate discretized trajectories of the two processes $X(t)$ and $Y(t)$ on the same probability space that they stay close to each other *trajectory by trajectory* with probability one. We apply the algorithm to compute the finite time error in Chapter 3.

2.5.5 Paired trajectories of Poisson process and of the diffusion process

Recall that according to Lemma 2.5.1 a unit-rate Poisson process has a strong diffusion approximation. Hence equation (2.5.3) also has a strong approximation given by equation (2.5.4). As the processes $P_k(\cdot)$ and $B_k(\cdot)$ are continuous time processes, we apply the τ-leaping approximation for equation (2.5.3) with the same step size h. This gives

$$\hat{X}_{n+1} = \hat{X}_n + \sum_k \frac{l_k}{V}\left[P_k\left(Vh\sum_{m=0}^{n} f_k(\hat{X}_m)\right) - P_k\left(Vh\sum_{m=0}^{n-1} f_k(\hat{X}_m)\right)\right] \tag{2.5.6}$$

with $\hat{X}_0 = X_0$. Similarly, the discretized approximation of equation (2.5.3), or the Euler-Maruyama method reads

$$\begin{aligned}\hat{Y}_{n+1} = \hat{Y}_n &+ \sum_k \frac{l_k}{V}(Vhf_k(\hat{Y}_n)) \\ &+ \sum_k \frac{l_k}{V}\left[B_k\left(Vh\sum_{m=0}^{n} f_k(\hat{Y}_m)\right) - B_k\left(Vh\sum_{m=0}^{n-1} f_k(\hat{Y}_m)\right)\right]\end{aligned} \tag{2.5.7}$$

with initial condition $\hat{Y}_0 = Y_0$.

The paired trajectories of $P_k(t)$ and $B_k(t)$ can be numerically generated by applying the KMT algorithm. The KMT algorithm actually generates a sequence of standard Poisson random variables $\{P_n\}$ and a sequence of standard normal random variables $\{W_n\}$, such that $\sum_{n=1}^{N} P_n$ is approximated by $N + \sum_{n=1}^{N} W_n$ for each finite N. Then after a re-scaling, one obtains a pair of discretized trajectories of $P_k(t)$ and $B_k(t)$ respectively. We refer [36] for a detailed review of the KMT algorithm.

CHAPTER 3

DATA-DRIVEN SOLVER FOR QSD

Recall that QSD solves the equation $-\lambda u = \mathcal{L}u$ where $\mathcal{L}$ is the Fokker-Planck operator defined in (2.2.2). The QSD solver consists of three components: an estimator of the killing rate λ, a Monte Carlo simulator of QSD that produces a reference solution, and an optimization problem similar as in [29].

3.1 Estimation of λ

Let $\hat{X} = \{\hat{X}_n^\delta,\ n \in \mathbb{Z}_+\}$ be a long numerical trajectory of X_t as described in Algorithm 1. Let $\boldsymbol{\tau} = \{\tau_m\}_{m=0}^M$ be recordings of killing times of the numerical trajectory such that X_t hits $\partial\mathcal{X}$ at $\tau_0, \tau_0+\tau_1, \tau_0+\tau_1+\tau_2, \cdots$ when running Algorithm 1. Note that $\boldsymbol{\tau}$ is an 1D vector and each element in $\boldsymbol{\tau}$ is a sample of the killing time. It is well known that if the QSD μ exists for a Markov process, then there exists a constant $\lambda > 0$ such that

$$\mathbb{P}_\mu[\tau > t] = e^{-\lambda t}\,.$$

Recall that the killing times $\boldsymbol{\tau}$ be exponentially distributed and the rate can be approximated by

$$\lambda = \frac{1}{\text{mean of } \boldsymbol{\tau}}.$$

One pitfall of the previous approach is that Algorithm 1 only gives a QSD when the time approaches to infinity. It is possible that $\boldsymbol{\tau}$ has not converged close enough to the desired exponential distribution. So it remains to check whether the limit is achieved. Our approach is to check the exponential tail in a log-linear plot. After having $\boldsymbol{\tau}$, it is easy to choose a sequence of times $t_0, t_1, \cdots, t_n$ and calculate $n_i = |\{\tau_m > t_i \,|\, 0 \le m \le M\}|$ for each $i = 0, \cdots n$. Then $p_i = n_i/M$ is an estimator of $\mathbb{P}_\mu[\tau > t_i]$. Now let p_i^u (resp. p_i^l) be the upper (resp. lower) bound of the confidence interval of p_i such that

$$p_i^u = \tilde{p} + z\sqrt{\frac{\tilde{p}}{\tilde{n}_i}(1-\tilde{p})} \quad (\text{ resp. } p_i^l = \tilde{p} - z\sqrt{\frac{\tilde{p}}{\tilde{n}_i}(1-\tilde{p})}) \,,$$

where $z = 1.96$, $\tilde{n}_i = n_i + z^2$ and $\tilde{p} = \frac{1}{\tilde{n}_i}(n_i + \frac{z^2}{2})$ [1]. If $p_i^l \le e^{-\lambda t_i} \le p_i^u$ for each $0 \le i \le n$, we accept the estimate λ. Otherwise we need to run Algorithm 1 for longer time to eliminate the initial bias in $\boldsymbol{\tau}$.

3.2 Data driven QSD solver.

The data driven solver for the Fokker-Planck equation introduced in [29] can be modified to solve the QSD for the stochastic differential equation (2.2.1). We use the same 2D setting in Section 2.3 to introduce the algorithm. Let the domain D and the boxes $\{O_{i,j}\}_{i=1,j=1}^{i=N,j=M}$ be the same as defined in Section 2.3. Let $\mathbf{u}$ be a vector in $\mathbb{R}^{N\times M}$ such that u_{ij} approximates the probability density function at the center of the box $O_{i,j}$. As introduced in [14], we consider $\mathbf{u}$ as the solution to the following linear system given by the spatial discretization of the Fokker-Planck equation (2.2.2) with respect to each center point:

$$\mathbf{A_0}\mathbf{u} = \lambda\mathbf{u}, \tag{3.2.1}$$

where $\mathbf{A_0}$ is an $(N-2)(M-2) \times (NM)$ matrix, which is called the discretized Fokker-Planck operator, and λ is the killing rate, which can be obtained by the way we mentioned in previous subsection. More precisely, each row in $\mathbf{A_0}$ describes the finite difference scheme of equation (2.2.2) with respect to a non-boundary point in the domain D.

Motivated by [29], we need the Monte Carlo simulation to produce a reference solution $\mathbf{v}$, which can be obtained via **Algorithm 1** in Chapter 2. Let $\hat{X} = \{\hat{X}_n^\delta\}_{n=1}^N$ be a long numerical trajectory of time-δ sample chain of process X_t produced by **Algorithm 1**, and let $\mathbf{v} = \{v_{i,j}\}_{i=1,j=1}^{i=N,j=M}$ such that

$$v_{i,j} = \frac{1}{Nh^2} \sum_{n=1}^{N} \mathbf{1}_{O_{i,j}}(\hat{X}_n^{\delta})$$

It follows from the convergence result in Proposition 2.2.1 that $\mathbf{v}$ is an approximate solution of equation (2.2.2) when the trajectory is sufficiently long. However, as discussed in [29], the trajectory needs to be extremely long to make $\mathbf{v}$ accurate enough. Noting that the error term of $\mathbf{v}$ has little spatial correlation, we use the following optimization problem to improve the accuracy of the solution.

$$\begin{gathered} \min \|\mathbf{u} - \mathbf{v}\|^2 \\ \text{subject to } \mathbf{A_0 u} = \lambda \mathbf{u}. \end{gathered} \tag{3.2.2}$$

The solution to the optimization problem (3.2.2) is called the least norm solution, which satisfies $\mathbf{u} = \mathbf{v} - \mathbf{A^T}(\mathbf{A}\mathbf{A^T})^{\mathbf{-1}}(\mathbf{A}\mathbf{v})$, with $\mathbf{A} = \mathbf{A_0} - \lambda \mathbf{I}$. [29]

An important method called the block data-driven solver is introduced in [14], in order to reduce the scale of numerical linear algebra problem in high dimensional problems. By dividing domain D into $K \times L$ blocks $\{D_{k,l}\}_{k=K,l=L}^{k=1,l=1}$ and discretizing the Fokker-Planck equation, the linear constraint on $D_{k,l}$ is

$$\mathbf{A_{k,l} u^{k,l}} = \lambda \mathbf{u^{k,l}},$$

where $\mathbf{A_{k,l}}$ is an $(N/K-2)(M/L-2) \times (NM/KL)$ matrix. The optimization problem on $D_{k,l}$ is

$$\mathbf{u_{k,l}} = -\mathbf{A_{k,l}^T}(\mathbf{A_{k,l}}\mathbf{A_{k,l}^T})^{\mathbf{-1}}\mathbf{A_{k,l}}\mathbf{v_{k,l}} + \mathbf{v_{k,l}},$$

where $\mathbf{v^{k,l}}$ is a reference solution obtained from the Monte-Carlo simulation. Then the numerical solution to Fokker-Planck equation (3.2.1) is collage of all $\{u_{k,l}\}_{k=K,l=L}^{k=1,l=1}$ on all blocks. However, the optimization problem ”pushes” most error terms to the boundary of domain, which makes the solution is less accurate near the boundary of each block. Paper [14] introduced the overlapping block method and the shifting blocks method to reduce the interface error. The overlapping block method enlarges the blocks and set the interior solution restricted to the original block as new solution, while the shifting block method moves the interface to the interior by shifting all blocks and recalculate the solution.

Note that in Section 3.1, we assume that λ is a pre-determined value given by the Monte Carlo simulation. Theoretically one can also search for the minimum of $\|\mathbf{u}-\mathbf{v}\|^2$ with respect to both λ and $\mathbf{v}$. But empirically we find that the result is not as accurate as using the killing rate λ from the Monte Carlo simulation, possibly because $\mathbf{v}$ has too much error.

One natural question is that how the simulation error in λ would affect the solution $\mathbf{u}$ to the optimization problem (3.2.2). Some linear algebraic calculation shows that the optimization problem (3.2.2) is fairly robust against small change of λ.

Theorem 3.2.1. *Let* $\mathbf{u}$ *and* $\mathbf{u}_1$ *be the solution to the optimization problem* (3.2.2) *with respect to killing rates* λ *and* λ_1 *respectively, where* $|\lambda - \lambda_1| = \epsilon \ll 1$. *Then*

$$\|\mathbf{u}-\mathbf{u}_1\| \leq 2s_{min}^{-1}\epsilon\|\mathbf{v}\| + O(\epsilon^2),$$

where s_{min} *is the smallest singular value of* $\mathbf{A}$.

Proof. Let $\mathbf{E} = \left[\epsilon\mathbf{I}_{(N-2)(M-2)}|\mathbf{0}\right]$ be an $(N-2)(M-2)\times(NM)$ perturbation matrix. Let $\widetilde{\mathbf{A}} = \mathbf{A}+\mathbf{E}$, $\mathbf{B} = \mathbf{A}^{\mathbf{T}}(\mathbf{A}\mathbf{A}^{\mathbf{T}})^{-\mathbf{1}}\mathbf{A}$ and $\widetilde{\mathbf{B}} = \widetilde{\mathbf{A}}^{\mathbf{T}}(\widetilde{\mathbf{A}}\widetilde{\mathbf{A}}^{\mathbf{T}})^{-\mathbf{1}}\widetilde{\mathbf{A}}$. Since $\mathbf{u} = \mathbf{v}-\mathbf{B}\mathbf{v}$ and $\mathbf{u}_1 = \mathbf{v}-\widetilde{\mathbf{B}}\mathbf{v}$, it is sufficient to prove

$$\|\mathbf{B}-\widetilde{\mathbf{B}}\| \leq 2s_{\min}^{-1}\epsilon + O(\epsilon^2).$$

Note that

$$\widetilde{\mathbf{A}}\widetilde{\mathbf{A}}^{\mathbf{T}} = (\mathbf{A}+\mathbf{E})(\mathbf{A}+\mathbf{E})^{\mathbf{T}} = \mathbf{A}\mathbf{A}^{\mathbf{T}} + \mathbf{A}\mathbf{E}^{\mathbf{T}} + \mathbf{E}\mathbf{A}^{\mathbf{T}} + \mathbf{E}\mathbf{E}^{\mathbf{T}}.$$

Since $\|\mathbf{EE^T}\|$ is $O(\epsilon^2)$, we can neglect it when we consider the inverse matrix of $\widetilde{\mathbf{A}}\widetilde{\mathbf{A}}^{\mathbf{T}}$. This means

$$(\widetilde{\mathbf{A}}\widetilde{\mathbf{A}}^{\mathbf{T}})^{\mathbf{-1}} \approx (\mathbf{AA^T})^{\mathbf{-1}} - (\mathbf{AA^T})^{\mathbf{-1}}(\mathbf{AE^T} + \mathbf{EA^T})(\mathbf{AA^T})^{\mathbf{-1}}.$$

Without considering the high order term $O(\epsilon^2)$, we can see

$$\begin{aligned}
\widetilde{\mathbf{A}}^{\mathbf{T}}(\widetilde{\mathbf{A}}\widetilde{\mathbf{A}}^{\mathbf{T}})^{\mathbf{-1}}\widetilde{\mathbf{A}} &\approx (\mathbf{A^T} + \mathbf{E^T})((\mathbf{AA^T})^{\mathbf{-1}} - (\mathbf{AA^T})^{\mathbf{-1}}(\mathbf{AE^T} + \mathbf{EA^T})(\mathbf{AA^T})^{\mathbf{-1}})(\mathbf{A} + \mathbf{E}) \\
&= \mathbf{A^T}(\mathbf{AA^T})^{\mathbf{-1}}\mathbf{A} + (\mathbf{E^T}(\mathbf{AA^T})^{\mathbf{-1}}\mathbf{A} - \mathbf{A^T}(\mathbf{AA^T})^{\mathbf{-1}}(\mathbf{AE^T})(\mathbf{AA^T})^{\mathbf{-1}}\mathbf{A}) \\
&\quad + (\mathbf{A^T}(\mathbf{AA^T})^{\mathbf{-1}}\mathbf{E} - \mathbf{A^T}(\mathbf{AA^T})^{\mathbf{-1}}(\mathbf{EA^T})(\mathbf{AA^T})^{\mathbf{-1}}\mathbf{A}) \\
&= \mathbf{B} + [\mathbf{E^T} - \mathbf{A^T}(\mathbf{AA^T})^{\mathbf{-1}}(\mathbf{AE^T})](\mathbf{AA^T})^{\mathbf{-1}}\mathbf{A} \\
&\quad + \mathbf{A^T}(\mathbf{AA^T})^{\mathbf{-1}}[\mathbf{E} - (\mathbf{EA^T})(\mathbf{AA^T})^{\mathbf{-1}}\mathbf{A}].
\end{aligned}$$

Consider the singular value decomposition(SVD) of matrix $\mathbf{A}$, i.e. $\mathbf{A} = \mathbf{uSv^T}$, wherein $\mathbf{S} = \left[\begin{array}{ccc|c} s_1 & & & 0 \\ & \ddots & & \vdots \\ & & s_{(N-2)(M-2)} & 0 \end{array}\right]$ is an $(N-2)(M-2) \times (NM)$ matrix and both $\mathbf{u} \in \mathbb{R}^{(N-2)(M-2)\times(N-2)(M-2)}, \mathbf{v} \in \mathbb{R}^{(NM)\times(NM)}$ are orthogonal. Then $\mathbf{A^T}(\mathbf{AA^T})^{\mathbf{-1}}\mathbf{A} = \mathbf{vD_1v^T}$, where $\mathbf{D_1} = \begin{bmatrix} \mathbf{I}_{(N-2)(M-2)} & 0 \\ 0 & 0 \end{bmatrix}_{(NM)\times(NM)}$, and

$$\begin{aligned}
(\mathbf{E^T} - \mathbf{A^T}(\mathbf{AA^T})^{\mathbf{-1}}\mathbf{AE^T}) &= (\mathbf{E^T} - \mathbf{vD_1v^TE^T}) \\
&= \mathbf{vD_2v^TE^T},\ \text{where } \mathbf{D_2} = \mathbf{I} - \mathbf{D_1}. \\
\mathbf{E} - (\mathbf{EA^T})(\mathbf{AA^T})^{\mathbf{-1}}\mathbf{A} &= \mathbf{E} - \mathbf{EA^T}(\mathbf{AA^T})^{\mathbf{-1}}\mathbf{A} \\
&= \mathbf{E} - \mathbf{EvD_1v^T} \\
&= \mathbf{EvD_2v^T}.
\end{aligned}$$

Note that $\|\mathbf{vD_2v^TE^T}\| \leq \epsilon, \|\mathbf{EvD_2v^T}\| \leq \epsilon$. Since $\mathbf{A^T}(\mathbf{AA^T})^{\mathbf{-1}}$ and $(\mathbf{AA^T})^{\mathbf{-1}}\mathbf{A}$ are two generalized inverse of $\mathbf{A}$,

$$\mathbf{A^T(AA^T)^{-1} = vS^{*T}u^T},\ \mathbf{(AA^T)^{-1}A = uS^*v},$$

where

$$\mathbf{S}^* = \left[\begin{array}{ccc|c} \frac{1}{s_1} & & & 0 \\ & \ddots & & \vdots \\ & & \frac{1}{s_{(N-2)(M-2)}} & 0 \end{array}\right] \tag{3.2.3}$$

and $\|\mathbf{A^T(AA^T)}^{-1}\| = \|\mathbf{(AA^T)^{-1}A}\| = \frac{1}{s_{\min}}$, hence we conclude that

$$\|\mathbf{B} - \widetilde{\mathbf{B}}\| \leq 2{s_{\min}}^{-1}\epsilon + O(\epsilon^2).$$

□

Remark 3.2.1. *It is very difficult to estimate the minimum singular value of matrix* $\mathbf{A}$ *analytically, even for the simplest case when the Fokker-Planck equation is just a heat equation. But empirically we find that* s_{min}^{-1} *is usually not very large. For example,* s_{min}^{-1} *for the gradient flow with a double well potential is Section 4.3 is* 0.4988*, and* s_{min}^{-1} *for the "ring example" in Section 3.3 is only* 0.2225*.*

3.3 Numerical Results

The first SDE example is the Ornstein–Uhlenbeck process :

$$\mathrm{d}X_t = \theta(\mu - X_t)\mathrm{d}t + \sigma\mathrm{d}W_t, \tag{3.3.1}$$

where $\theta > 0$ and $\sigma > 0$ are parameters, μ is a constant. In addition, W_t is a Wiener process, and σ is the strength of the noise. In our simulation, we set $\theta = 1, \mu = 2, \sigma = 1$ and the absorbing set $\partial\mathcal{X} = (-\infty, 0] \cup [3, \infty)$. In addition, we apply the Monte Carlo simulation with 512 mesh points on the interval $[0, 3]$.

We first need to use Algorithm 1 to estimate the survival rate λ. Our simulation uses Euler-Maruyama scheme with $\delta = 0.001$ and sample size $N = 10^6$ and $N = 10^8$ depending on the setting. All samples of killing times are recorded to plot the exponential tail. The mean of killing times gives an estimate $\lambda = 0.267176$. The exponential tail of $\mathbb{P}[\tau > t]$ vs. t, the upper and lower bound of the confidence interval, and the plot of $e^{-\lambda t}$ are compared in Figure 3.1. We can see that the plot of $e^{-\lambda t}$ falls in the confidence interval for all t. Hence the estimate of λ is accepted.

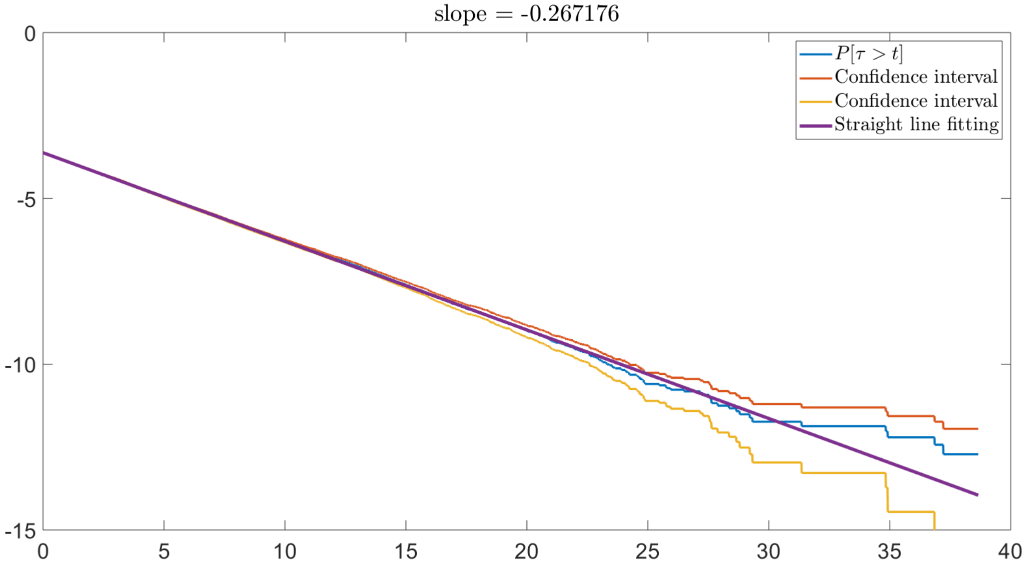

Figure 3.1: Plot of $\mathbb{P}(\tau > t)$ vs. t, confidence interval(upper bound and lower bound) and function $y = e^{-\lambda t}$

.

Furthermore, we would like to show the robustness of our data-driven QSD solver. The QSD is not explicit given so we use very large sample size (10^{10} samples) and very small step size (10^{-4}) to obtain a much more accurate solution, which is served as the benchmark. Then we compare the numerical solutions obtained by the Monte Carlo method and the data-driven method for QSD with $N = 10^6$ and $N = 10^8$ samples, respectively. The result is shown in the first column of Figure 3.2. The data-driven solver performs much better than the Monte Carlo approximation for $N = 10^6$ samples. It takes 10^8 samples for the direct Monte Carlo sampler to produce a solution that looks as good as the QSD solver. Similar as the data-driven Fokker-Planck solver, our data-driven QSD solver can tolerate high level error in Monte Carlo simulation that has small spatial correlation.

It remains to check the effect of Brownian Bridge. We apply different time step sizes $\delta = 0.01$ and $\delta = 0.001$ for each trajectory. We use 10^7 samples for $\delta = 0.001$ and 10^6 samples for $\delta = 0.01$ to make sure that the number of killing events (for estimating the killing rate) are comparable. When $\delta = 0.001$, the error is small with and without Brownian bridge correction. But Brownian bridge correction obviously improves the quality of solution when $\delta = 0.01$. See the lower left panel of Figure 3.2. This is expected because, with larger time step size, the probability that the Brownian bridge hits the absorbing set $\partial\mathcal{X}$ gets higher.

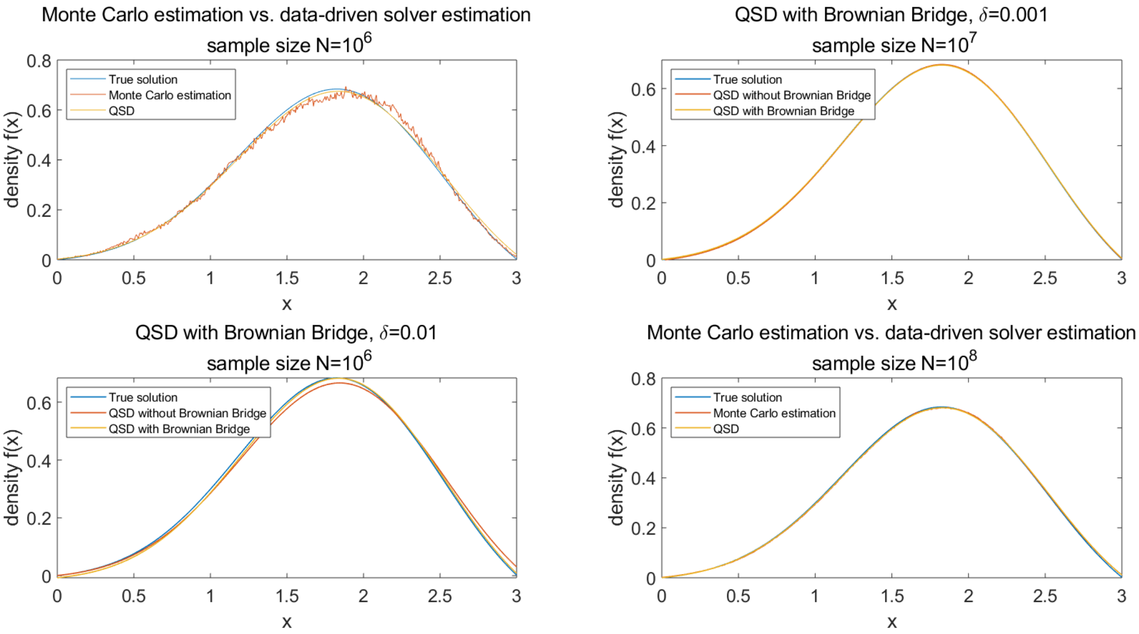

Figure 3.2: **Upper panel**:(**Left**) Monte Carlo estimation vs. data-driven solver estimation for sample size $N = 10^6$. (**Right**) Effect of Brownian Bridge for sample size $N = 10^7$ and time step size 0.001. **Lower panel**: (**Left**) Effect of Brownian Bridge for sample size $N = 10^6$ and time step size 0.01. (**Right**) Monte Carlo estimation vs. data-driven solver estimation for sample size $N = 10^8$.

3.3.1 Wright-Fisher Diffusion

The second numerical example is the Wright-Fisher diffusion model, which describes the evolution of colony undergoing random mating, possible under the additional actions of mutation and selection with or without dominance [22]. The Wright-Fisher model is an SDE

$$dX_t = -X_t dt + \sqrt{X_t(1 - X_t)} dW_t,$$

where W_t is a Wiener process and $\partial \mathcal{X} = \{0\}$ is the absorbing set. By the analysis of [22], the Yaglom limit, i.e., the QSD, satisfies

$$\lim_{t \to \infty} \mathbb{P}[X_t \in dy | \tau > t] = 2(1 - y)dy.$$

The goal of this example is to show the effect of Brownian bridge when the coefficient of noise is singular at the boundary. Since the Euler-Maruyama scheme only has an order of accuracy 0.5, in the simulation, we apply the Milstein scheme, which reads

$$\hat{X}^\delta_{n+1} = \hat{X}^\delta_n - \hat{X}^\delta_n \delta + \sqrt{\hat{X}^\delta_n(1 - \hat{X}^\delta_n)}(W_{(n+1)\delta} - W_{n\delta}) + \frac{1}{4}(1 - 2\hat{X}^\delta_n)[(W_{(n+1)\delta} - W_{n\delta})^2 - \delta]$$

One difficulty of using the Brownian bridge correction in this model is that the coefficient of the Brownian motion is vanishing at the boundary. Recall that the strength coefficient of Brownian bridge is denoted by ϕ. Larger ϕ means X_t has higher probability of hitting the boundary. Since the coefficient of the Brownian motion is vanishing at the boundary, the effective strength coefficient ϕ becomes dramatically smaller when X_t gets closer to the boundary. As a result, it is not a good idea to still approximate X_t by a Brownian motion. And the original strength coefficient $\phi = \sqrt{\hat{X}^\delta_n(1 - \hat{X}^\delta_n)}$ can dramatically overestimate the probability of hitting the boundary. Estimating the hitting probability of this diffusion bridge is a difficult problem that is well beyond the scope of this thesis. To the best of our knowledge, it is not possible to explicitly calculate the conditional distribution of the diffusion bridge that starts from $x := \hat{X}^{\delta}_{n}$ and ends at $y := \hat{X}^{\delta}_{n+1}$. Instead, we use an empirically found strength coefficient $\phi^2 = \frac{1}{3}\min\{x(1-x), y(1-y)\}$ can fix this problem. We note that this is not a simple ad-hoc solution because in a stochastic differential equation, the diffusion plays a dominate role in a very short time interval. Hence a similar modification of the Brownian bridge should work for many stochastic differential equations with $\sqrt{X_t}\mathrm{d}B_t$ noise terms. In particular, the vanishing coefficient $\sqrt{X_t}\mathrm{d}B_t$ also appears in many ecological models. We will implement this modified Brownian bridge correction when simulating these models.

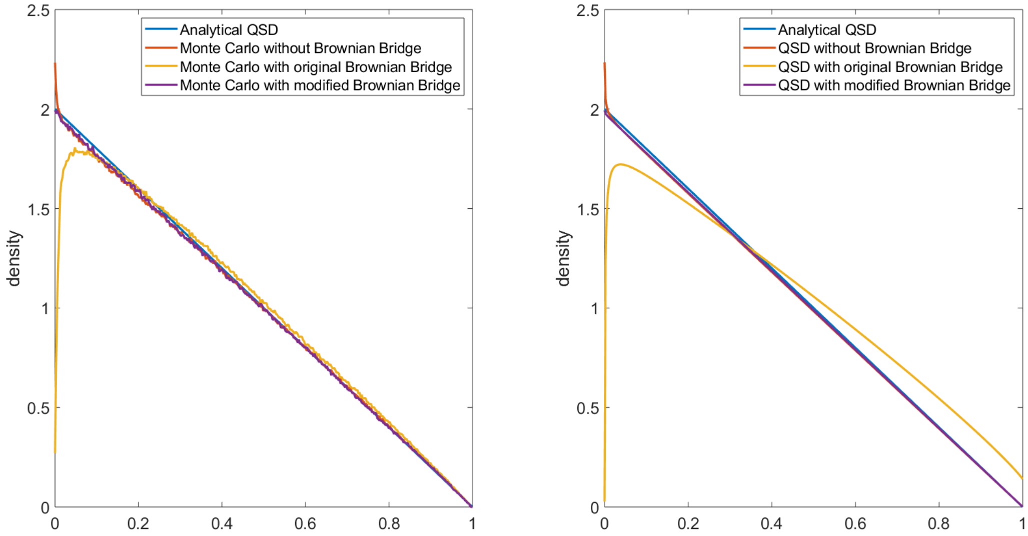

Figure 3.3: Effect of Brownian Bridge and a correction of Brownian Bridge. **Left**: Monte Carlo approximations without Brownian Bridge correction, with original Brownian Bridge correction, and with modified Brownian Bridge correction, in comparison to the analytical QSD. **Right**: Result from the data-driven QSD solver using the Monte Carlo simulation data from Left panel.

The effect of Brownian bridge is shown in the left side of Figure 3.3. We compare the solutions obtained via Monte Carlo method and the data-driven method with Brownian Bridge by running 10^7 samples on $[0, 1]$ with time step size $\delta = 0.01$. The Monte Carlo approximation is far from the true density function of Beta(1,2) near $x = 0$, while the use of the original Brownian Bridge only makes things worse. The modified Brownian Bridge solves this boundary effect problem reasonably well. The output of the data-driven QSD solver has a similar result (Figure 3.3 Right). Let $x = \hat{X}^{\delta}_{n}$ and $y = \hat{X}^{\delta}_{n+1}$. One can see that the numerical QSD is much closer to the true distribution if we replace the strength coefficient of the Brownian bridge $\phi^2 = x(1-x)$ by the modified strength coefficient $\phi^2 = \frac{1}{3}\min\{x(1-x), y(1-y)\}$.

3.3.2 Ring density function

Consider the following stochastic differential equation:

$$dX = (-4X(X^2+Y^2-1)+Y)dt + \epsilon dW_t^X$$
$$dY = (-4X(X^2+Y^2-1)-X)dt + \epsilon dW_t^Y,$$

where W_t^X and W_t^Y are independent Wiener processes. In the simulation, we set the strength of noise $\epsilon = 1$.

We first look at the approximation obtained by Monte Carlo method with 256×256 mesh points on the domain $D = [-1.5, 1.5] \times [-1.5, 1.5]$. The simulation uses step size $\delta = 0.001$ and $N = 10^8$ samples. (See upper left panel in Figure 3.4). The Monte Carlo approximation has too much noise to be useful. The quality of this solution can be significantly improved by using the data-driven QSD solver. See upper right panel in Figure 3.4.

The simulation result shows the estimated rate of killing $\lambda = -0.176302$. We use this example to test the sensitivity of solution $\mathbf{u}$ against small change of the killing rate. We compare the approximations obtained by setting the killing rate be λ, 1.1λ and 0.9λ respectively. Heat maps of the difference between QSDs with “correct” and “wrong” killing rates are shown in two middle panels in Figure 3.4. It shows that difference brought by “wrong” rates is only $\approx O(10^{-4})$, which can be neglected. This result coincides with the analysis in Theorem 3.2.1 in this dissertation.

Finally, we would like to emphasis that the data-driven QSD solver can tolerate very high level of spatially uncorrelated noise in the reference solution $\mathbf{v}$. For example, if we use the same long trajectory with 10^8 samples that generates the top left panel of Figure

3.4, but only select 10^5 samples with intervals of 10^3 steps of the numerical SDE solver, the Monte Carlo data becomes very noisy (Bottom left panel of Figure 3.4). However, longer intermittency between samples also reduces the spatial correlation between samples. As a result, the output of the QSD solver has very little change except at the boundary grid points, because the optimization problem (3.2.2) projects most of error to the boundary of the domain. (See bottom right panel of Figure 3.4.) This result highlights the need of high quality samplers. A Monte Carlo sampler with smaller spatial correlation between samples can significantly reduce the number of samples needed in the data-driven QSD solver.

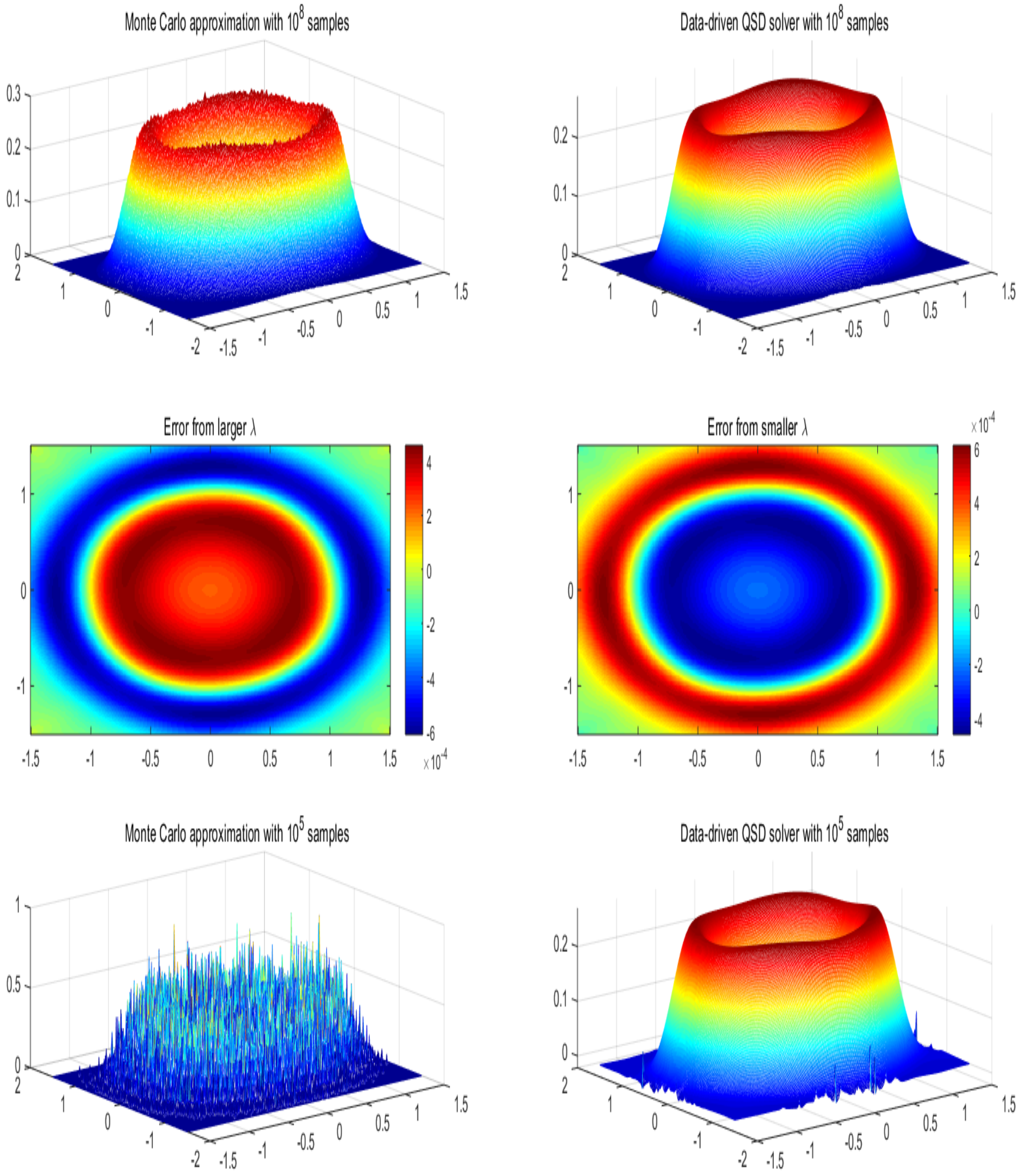

Figure 3.4: (Ring density) **Upper panel:**The approximation by Monte Carlo simulation(left) and the algorithm in Section 3.2(right) with 256×256 mesh points and 10^8 samples. **Middle panel**: Sensitivity effect of small change to killing rate λ. **Lower panel:** The approximation by Monte Carlo simulation with smaller samples(left) and the output of the data-driven QSD solver(right).

CHAPTER 4

SENSITIVITY ANALYSIS OF QSDS

A stochastic differential equation has a QSD usually because it has a natural absorbing state. For example, in ecological models, this absorbing state is the natural boundary of the domain at which the population of some species is zero. Obviously invariant probability measures are easier to study than QSDs. One interesting question is that if we slightly modify the equation such that it does not have absorbing states any more, how can we quantify the difference between QSD and the invariant probability measure after the modification? This is called the sensitivity analysis of QSDs.

In this chapter, we focus on the difference between the QSD of a stochastic differential equation $X = \{X_t, t \in \mathbb{R}\}$ and the invariant probability measure of a modification of X, denoted by $Y = \{Y_t, t \in \mathbb{R}\}$. For the sake of simplicity, this thesis only compares the QSD (resp. invariant probability measure) of the numerical trajectory of X (resp. Y), denoted by $\hat{X} = \{\hat{X}_n^\delta, n \in \mathbb{Z}_+\}$ (resp. $\hat{Y} = \{\hat{Y}_n^\delta, n \in \mathbb{Z}_+\}$). Denote the QSD (resp. invariant probability measure) of $\hat{X}$ (resp. $\hat{Y}$) by $\hat{\mu}_X$ (resp. $\hat{\mu}_Y$) and the QSD (resp. invariant probability measure) of the original SDE X (resp. Y) by μ_X (resp. μ_Y). The sensitivity of invariant probability measure against time discretization has been addressed in [15]. When the time step size of the time discretization is small enough, the invariant probability measure μ_Y is close to the numerical invariant probability measure $\hat{\mu}_Y$. The case of QSD is analogous. Hence $d(\hat{\mu}_X, \hat{\mu}_Y)$ is usually a good approximation of $d(\mu_X, \mu_Y)$.

We are mainly interested in the following two different modifications of X.

4.1 Modification of X

4.1.1 Case 1: Reflection at $\partial\mathcal{X}$

One easy way to modify the numerical trajectory $\hat{X}$ to eliminate the absorbing state is to add a reflecting boundary. This method preserves the local dynamics but changes the boundary condition. More precisely, the trajectory of $\hat{X}$ follows that of $\hat{Y}$ until it hits the boundary $\partial\mathcal{X}$. Without loss of generality assume $\partial\mathcal{X}$ is a smooth manifold embedded in $\mathbb{R}^d$. If $\hat{Y}^\delta_n = \hat{X}^\delta_n \in \mathcal{X}^a$ but $\hat{X}^\delta_{n+1} \notin \mathcal{X}^a$, then $\hat{Y}^\delta_{n+1}$ is the mirror reflection of $\hat{X}^\delta_{n+1}$ against the boundary of $\mathcal{X}$. Denote the intersection of $\partial\mathcal{X}$ and the line segment from $\hat{X}^\delta_n$ to $\hat{X}^\delta_{n+1}$ by X_a. Denote the unit normal vector of $\partial\mathcal{X}$ at X_a by $\mathbf{n}$. Then some simple calculations give

$$\hat{Y}^\delta_{n+1} = X_a + (\mathrm{Id} - 2\mathbf{n}\mathbf{n}^T)(\hat{X}^\delta_n - X_a)\,.$$

In most ecological and epidemiological models the natural boundary is $\partial\mathcal{X} = \{(x_1, \cdots, x_n)\,|\, x_i = 0 \text{ for some } 1 \le i \le n, x_j \ge 0 \text{ for all } 1 \le j \le n\}$. In this setting $\hat{Y}^\delta_{n+1}$ has an easy expression $\hat{Y}^\delta_{n+1} = \mathrm{abs}(\hat{X}^\delta_{n+1})$, where abs$(\cdot)$ means element-wise absolute value of a vector.

We remark that this reflection of the numerical trajectory is not consistent with the stochastic differential equation with reflecting boundary conditions, which solves a Skorohod's equation [39]. The problem is that the noise vanishes at $\partial\mathcal{X}$ for most ecological models. Hence ∂X remains to be absorbing even if the stochastic differential equation has reflecting boundaries. The "reflection" we use here is only for the numerical trajectory. It can be interpreted as a small random number of individuals are artificially added immediately after the extinction of this species happens. The goal is to show that sometimes the quasi-stationary distribution is not very sensitive against a change of the boundary condition. See the summary at the end of Section 4.2.

4.1.2 Case 2: Demographic noise in ecological models.

We would also like to address the case of demographic noise in a class of ecological models, such as population model and epidemic model. For the sake of simplicity consider an 1D birth-death process with some environment noise

$$\mathrm{d}Y_t = f(Y_t)\mathrm{d}t + \sigma Y_t \mathrm{d}W_t\,, \tag{4.1.1}$$

where $Y_t \in \mathbb{R}$. Note that for the birth-death process model, $\{0\}$ is an absorbing state. It is known that for suitable $f(Y_t)$, the probability of Y_t hitting zero in finite time is zero [21]. However, if we take the demographic noise, i.e., the randomness of birth/death events, into consideration, the birth-death process becomes

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \sigma' X_t \mathrm{d}W_t^{(1)} + \epsilon\sqrt{X_t}\mathrm{d}W_t^{(2)}\,, \tag{4.1.2}$$

where $\epsilon \ll 1$ is a small parameter that is proportional to $-1/2$-th power of the scale of the population size, σ' is the new parameter that address the separation of environment noise and demographic noise. For example, if the steady state of X_t is around 1, we can choose $\sigma' = \sqrt{\sigma^2 - \epsilon^2}$.

Different from equation (4.1.1), the magnitude of random perturbation in equation (4.1.2) near the boundary is much larger. As a result, equation (4.1.2) could hit the boundary in finite time with strictly positive probability. (For example $\mathrm{d}X_t = rX_t\mathrm{d}t + \epsilon\sqrt{X_t}\mathrm{d}W_t$ has strictly positive extinction probability in finite time whenever $\epsilon > 0$. This can be checked by applying the Itô's formula to a test function $1/X_t$ then take the expectation. Whether equation (4.1.2) has finite time extinction depends on details of $f(x)$ and ϵ. But most ecological model has $f(X_t) \approx O(1) \times X_t$ when X_t is small in order to model the intrinsic growth of the population. That fits the setting of the example above.) Therefore, it is common for equation (4.1.1) to admit an invariant probability measure while equation (4.1.2) has a QSD. One very interesting question is that, if ϵ is sufficiently small, how different is the invariant probability measure of equation (4.1.1) from the QSD of equation (4.1.2)? This is very important in the study of ecological models because theoretically every model is subject a small demographic noise. If the invariant probability measure is dramatically different from the QSD after adding a small demographic noise term, then the equation (4.1.1) is not a good model due to high sensitivity, and we must study the equation (4.1.2) directly.

4.2 Methodology

We roughly follow [15] to carry out the sensitivity analysis of QSD. Here we slightly modify $\hat{X}^\delta_n$ such that if $\hat{X}^\delta_n \in \partial\mathcal{X}$, instead of sampling from the occupation measure, we immediately re-sample $\hat{X}^\delta_n$ from the QSD $\hat{\mu}_X$. This new process, denoted by $\tilde{X}^\delta_n$, admits an invariant probability measure $\hat{\mu}_X$. Now denote the transition kernel of $\tilde{X}^\delta_n$ and $\hat{Y}^\delta_n$ by P_X and P_Y respectively. The following Proposition is motivated by [23].

Proposition 4.2.1. *For any* $T > 0$*, if there exists a constant* $0 < \alpha < 1$ *such that*

$$d_w(\hat{\mu}_X P_Y^T, \hat{\mu}_Y P_Y^T) \leq \alpha d_w(\hat{\mu}_X, \hat{\mu}_Y)\,,$$

then we have

$$d_w(\hat{\mu}_X, \hat{\mu}_Y) \leq \frac{d_w(\hat{\mu}_X P_X^T, \hat{\mu}_X P_Y^T)}{1-\alpha}.$$

Proof. Let $d_w(\cdot,\cdot)$ be the 1-Wasserstein distance defined in Section 2.4. We can decompose $d_w(\hat{\mu}_X, \hat{\mu}_Y)$ via the following triangle inequality:

$$d_w(\hat{\mu}_X, \hat{\mu}_Y) \leq d_w(\hat{\mu}_X P_X^T, \hat{\mu}_X P_Y^T) + d_w(\hat{\mu}_X P_Y^T, \hat{\mu}_Y P_Y^T).$$

Since the transition kernel P_Y^T has enough contraction such that

$$d_w(\hat{\mu}_X P_Y^T, \hat{\mu}_Y P_Y^T) \leq \alpha d_w(\hat{\mu}_X, \hat{\mu}_Y)$$

for some $\alpha < 1$, after some simplification, we have

$$d_w(\hat{\mu}_X, \hat{\mu}_Y) \leq \frac{d_w(\hat{\mu}_X P_X^T, \hat{\mu}_X P_Y^T)}{1-\alpha}.$$

□

Therefore, in order to estimate $d_w(\hat{\mu}_X, \hat{\mu}_Y)$, we need to look for suitable numerical estimators of the finite time error and the speed of contraction of P_Y^T. The finite time error can be easily estimated in both cases. And the speed of contraction α comes from the geometric ergodicity of the Markov process $\hat{Y}$. If our numerical estimation gives

$$\mathrm{d}_w(\mu P_Y^T, \nu P_Y^T) \leq Ce^{-\gamma T}\,, \tag{4.2.1}$$

then we set $\alpha = e^{-\gamma T}$. As discussed in [15], this is a quick way to estimate α. In fact, in all examples that we have tested, when starting from $\hat{\mu}_X$ and $\hat{\mu}_Y$, the prefactor C of the coupling probability in equation (4.2.1) is not far away from 1. Hence in practice it does not differ from the "true upper bound" very much. The "true upper bound" of α in [15] comes from the extreme value theory, which is much more expensive to compute.

4.2.1 Estimation of contraction rate

Motivated by [31], we use the following coupling method to estimate the contraction rate α. Let $\hat{Z}_n^\delta = (\hat{Y}_n^1, \hat{Y}_n^2)$ be a Markov process in $\mathbb{R}^{2d}$ such that $\hat{Y}_n^1$ and $\hat{Y}_n^2$ are two copies of $\hat{Y}$. Recall that τ^c is the coupling time, which is also the first passage time to the "diagonal" hyperplane $\{(\mathbf{x}, \mathbf{y} \in \mathbb{R}^{2d}) | \mathbf{y} = \mathbf{x}\}$. Then by Lemma 2.4.1

$$d_w(\hat{\mu}_X P_Y^t, \hat{\mu}_Y P_Y^t) \leq \mathbb{P}[\tau^c > t].$$

As discussed in [31], we need a hybrid coupling scheme to make sure that two numerical trajectories can couple. Some coupling methods such as reflection coupling or synchronous coupling are implemented in the first phase to bring two numerical trajectories together. Then we compare the probability density function for the next step and couple these two numerical trajectories with the maximal possible probability (called the maximal coupling). After doing this for many times, we will have many samples of τ^c denote by $\boldsymbol{\tau}^c$. We use the exponential tail of $\mathbb{P}[\tau^c > t]$ to estimate the contraction rate α. More precisely, we look for a constant $\gamma > 0$ such that

$$-\gamma = \lim_{t\to\infty} \frac{1}{t} \log(\mathbb{P}[\tau^c > t])$$

if the limit exists. See **Algorithm 3** for the detail of implementation of coupling. Note that we cannot simply compute the contraction rate start from $t = 0$ because only the tail of coupling time can be considered as exponential distributed. In practice, we apply the same method as we compute the killing rate in section 3.1. After having $\boldsymbol{\tau}^c$, it is easy to choose a sequence of times $t_0, t_1, \cdots, t_n$ and calculate $n_i = |\{\tau^c_m > t_i \,|\, 0 \le m \le M\}|$ for each $i = 0, \cdots n$. Then $p_i = n_i/M$ is an estimator of $\mathbb{P}_\mu[\tau^c > t_i]$. Now let p^u_i (resp. p^l_i) be the upper (resp. lower) bound of the confidence interval of p_i such that

$$p^u_i = \tilde{p} + z\sqrt{\frac{\tilde{p}}{\tilde{n}_i}(1-\tilde{p})} \quad \text{resp. } p^l_i = \tilde{p} - z\sqrt{\frac{\tilde{p}}{\tilde{n}_i}(1-\tilde{p})}\,,$$

where $z = 1.96$, $\tilde{n}_i = n_i + z^2$ and $\tilde{p} = \frac{1}{\tilde{n}_i}(n_i + \frac{z^2}{2})$. Let t_n be the largest time that we can still collect available samples. If there exist constants C and $i_0 < n$ such that $p^l_i \le Ce^{\gamma t_i} \le p^u_i$ for each $i_0 \le i \le n$, we say that the exponential tail starts at $t = t_{i_0}$. We accept the estimate of the exponential tail with rate $e^{-\gamma t}$ if the confidence interval $p^u_{i_0} - p^l_{i_0}$ is sufficiently small, i.e., the estimate of coupling probability at $t = dt_{i_0}$ is sufficiently trustable. Otherwise we need to run Algorithm 1 for longer time to eliminate the initial bias in $\boldsymbol{\tau}^c$.

4.2.2 Estimator of error terms

It remains to estimate the finite time error $d_w(\hat{\mu}_X P^T_X, \hat{\mu}_X P^T_Y)$. As we mentioned in the beginning of this section, we will consider two different cases and estimate the finite time errors respectively.

4.2.2.1 Case 1: Reflection at $\partial\mathcal{X}$

Recall that the modified Markov process $\hat{Y}$ reflects at the boundary $\partial\mathcal{X}$ when it hits the boundary. Hence two trajectories

Algorithm 3 Estimation of contraction rate α

```
Input: Initial values $x, y \in K$
Output: An estimation of contraction rate $\alpha$
  Choose threshold $d > 0$
  for $i = 1$ to $N_s$ do
    $\tau_i^c = 0, t = 0, (\hat{Y}_t^1, \hat{Y}_t^2) = (x, y)$
    Flag $= 0$
    while Flag=0 do
      if $|\hat{Y}_t^1 - \hat{Y}_t^2| > d$ then
        Compute $(\hat{Y}_{t+1}^1, \hat{Y}_{t+1}^2)$ using reflection coupling or independent coupling
        $t \leftarrow t + 1$
      else
        Compute $(\hat{Y}_{t+1}^1, \hat{Y}_{t+1}^2)$ using maximal coupling
        if coupled successfully then
          Flag=1
          $\tau_i^c = t$
        else
          $t \leftarrow t + 1$
        end if
      end if
    end while
  end for
  Use $\tau_1^c, \cdots, \tau_{N_s}^c$ to compute $\mathbb{P}(\tau^c > t)$
  Fit the tail of $\log \mathbb{P}(\tau^c > t)$ versus $t$ by linear regression. Compute the slope $\gamma$.
```

$$\tilde{X}^{\delta}_{n+1} = \tilde{X}^{\delta}_{n} + f(\tilde{X}^{\delta}_{n})\delta + \sigma(\tilde{X}^{\delta}_{n})(W_{(n+1)\delta} - W_{n\delta})$$
$$\hat{Y}^{\delta}_{n+1} = \hat{Y}^{\delta}_{n} + f(\hat{Y}^{\delta}_{n})\delta + \sigma(\hat{Y}^{\delta}_{n})(W_{(n+1)\delta} - W_{n\delta})$$

are identical if we set the same noise in the simulation process. $\tilde{X}$ only differs from $\hat{Y}$ when $\tilde{X}$ hits the boundary $\partial\mathcal{X}$. When $\tilde{X}$ and $\hat{Y}$ hit the boundary, $\tilde{X}$ is resampled from $\hat{\mu}_X$, and $\hat{Y}$ reflects at the boundary. Hence the finite time error $d_w(\hat{\mu}_X P^T_X, \hat{\mu}_X P^T_Y)$ is bounded from above by the killing probability within the time interval $[0, T]$ when starting from $\hat{\mu}_X$.

In order to sample initial value $\mathbf{x}$ from the numerical invariant measure $\hat{\mu}_X$, we consider a long trajectory $\{\tilde{X}^{\delta}_n\}$. The distance between $\tilde{X}$ and the modified trajectory $\hat{Y}$ is recorded after time T. Then we nm j m,let $\hat{Y}^{\delta}_0 = \tilde{X}^{\delta}_0 = \mathbf{x}_{i+1}$ and restart the simulation, where $\mathbf{x}_{i+1} = \tilde{X}^{\delta}_T$ from the $i-$th iteration. See the **Algorithm 4** for the detail.

Algorithm 4 Estimate finite time error for Case 1

```
Input: Initial value $\mathbf{x_1}$
Output: An estimator of $d_w(\hat{\mu}_X P^T_X, \hat{\mu}_X P^T_Y)$
    for $i = 1$ to $N_s$ do
        Using the same noise, simulate $\tilde{X}$ and $\hat{Y}$ with initial value $\mathbf{x_i}$ up to T
        Set $d_i = 0$
        if $\tau < T$ then
            Regenerate $\tilde{X}$ as its empirical distribution
            $d_i = d(\tilde{X}^{\delta}_T, \hat{Y}^{\delta}_T)$
        end if
        Let $\mathbf{x_{i+1}} = \tilde{X}^{\delta}_T$
    end for
    Return $\frac{1}{N_s}\sum_{i=1}^{N_s} d_i$
```

When the number of samples is sufficiently large, $\mathbf{x_1}, \cdots, \mathbf{x_{N_s}}$ are from a long trajectory of the time-T skeleton of $\tilde{X}_T$. Hence they are approximately sampled from $\hat{\mu}_X$. The error term $d_i = d(\tilde{X}^{\delta}_T, \hat{Y}^{\delta}_T)$ for $\tilde{X}^{\delta}_0 = \hat{Y}^{\delta}_0 = \mathbf{x_i}$ estimates $d(\tilde{X}_T, \hat{Y}_T)$. Let $\hat{\mu}^2_X$ be the probability measure on $\mathbb{R}^d \times \mathbb{R}^d$ that is supported by the hyperplane $\{(x, y) \in \mathbb{R}^d \times \mathbb{R}^d \,|\, x = y\}$ such that $\hat{\mu}^2_X(\{(x, x) \,|\, x \in A\}) = \hat{\mu}_X(A)$ for any $A \in \mathcal{B}(\mathcal{X})$. Since the pushforward map $\hat{\mu}^2_X(P^T_X \times P^T_Y)$ is a coupling, it is easy to see that the output of **Algorithm 4** gives an upper bound of $d_w(\hat{\mu}_X P^T_X, \hat{\mu}_X P^T_Y)$.

From the analysis above, we have the following lemma, which gives an upper bound of the finite time error $d_w(\hat{\mu}_X P_X^T, \hat{\mu}_X P_Y^T)$.

Lemma 4.2.1. *For the Wasserstein distance induced by the distance given in (2.4.1), we have*

$$d_w(\hat{\mu}_X P_X^T, \hat{\mu}_X P_Y^T) \leq \int_{\mathbb{R}^d} \mathbb{P}_x(\tau < T)\hat{\mu}_X(dx),$$

where x is the initial value with distribution μ_Y.

Proof. Note that $\mu_X^2(P_X^T \times P_Y^T)$ is a coupling of $\mu_X P_X^T$ and $\mu_X P_Y^T$. From the definition of Wasserstein distance, we have

$$\begin{aligned} d_w(\hat{\mu}_X P_X^T, \hat{\mu}_X P_Y^T) &\leq \int_{\mathbb{R}^d \times \mathbb{R}^d} d(x,y)\hat{\mu}_X^2(P_X^T \times P_Y^T)(\mathrm{d}x\mathrm{d}y) \\ &= \int_{\mathbb{R}^d} \mathbb{E}_{(x,x)}[d(\tilde{X}_T^\delta, \hat{Y}_T^\delta)]\hat{\mu}_X(dx) \\ &= \int_{\mathbb{R}^d} \mathbb{P}_x(\tau < T) d(\tilde{X}_T^\delta, \hat{Y}_T^\delta)\hat{\mu}_X(dx) \\ &\leq \int_{\mathbb{R}^d} \mathbb{P}_x(\tau < T)\hat{\mu}_X(dx), \end{aligned}$$

the inequality in the last step comes from the definition $d(x,y) = \max(1, \|x-y\|)$. □

4.2.2.2 Case 2: Impact of a demographic noise $\epsilon\sqrt{X_t}\mathrm{d}W_t$

Another common way of modification in ecological models is to add a demographic noise. Let $\hat{X}$ be the numerical trajectory of the SDE with an additive demographic noise $\epsilon\sqrt{X_t}\mathrm{d}W_t$. Let $\tilde{X}$ be the modification of $\hat{X}$ that resample from $\hat{\mu}_X$ whenever hitting $\partial\mathcal{X}$ so that it admits $\hat{\mu}_X$ as an invariant probability measure. Let $\hat{Y}$ be the numerical trajectory of the SDE without demographic noise so that $\hat{Y}$ admits an invariant probability measure. We have trajectories

$$\begin{aligned} \tilde{X}_{n+1}^\delta &= \tilde{X}_n^\delta + f(\tilde{X}_n^\delta)\delta + \sigma(\tilde{X}_n^\delta)(W_{(n+1)\delta} - W_{n\delta}) + \epsilon\sqrt{\tilde{X}_n^\delta}(W'_{(n+1)\delta} - W'_{n\delta}) \\ \hat{Y}_{n+1}^\delta &= \hat{Y}_n^\delta + f(\hat{Y}_n^\delta)\delta + \sigma'(\hat{Y}_n^\delta)(W_{(n+1)\delta} - W_{n\delta})\,. \end{aligned}$$

Here we assume that $\hat{Y}$ has zero probability to hit the absorbing set $\partial\mathcal{X}$ in finite time. Different from the Case 1, we will need to study the effect of the demographic noise. When estimating the finite time error $d_w(\hat{\mu}_X P_X^T, \hat{\mu}_X P_Y^T)$, we still need to sample the initial value $\mathbf{x}$ from $\hat{\mu}_X$ and record the distance between these two trajectories $\tilde{X}$ and $\hat{Y}$ up to time T. The distance between $\tilde{X}$ and $\hat{Y}$ can be decomposed into two parts: one is from the killing and resampling, the other is from the demographic noise. The first term is the same as in Case 1. The second term is due to the nonzero demographic noise that can separate $\tilde{X}$ and $\hat{Y}$ before the killing. In a population model, this effect is more obvious when one species has small population, because $\sqrt{x} \gg x$ when $0 < x \ll 1$. See the description of **Algorithm 5** for the full detail.

Algorithm 5 Estimate finite time error for Case 2

```
Input: Initial value $\mathbf{x_1}$
Output: An estimator of $d_w(\mu_X P_X^T, \mu_X P_Y^T)$
  for $i = 1$ to $N_s$ do
    Using the same noise, simulate $\tilde{X}$ and $\hat{Y}$ with initial value $\mathbf{x_i}$ up to T
    Set Flag = 0, $d_i = 0$
    if $\tau < T$ then
      Regenerate $\tilde{X}$ as its empirical distribution
      $\eta_i = d(\tilde{X}^\delta_T, \hat{Y}^\delta_T)$
      Flag = 1
    else
      $\theta_i = d(\tilde{X}^\delta_T, \hat{Y}^\delta_T)$
    end if
    $d_i = \theta_i + \mathbf{1}_{\{\text{Flag}=1\}}(\eta_i - \theta_i)$
    Let $\mathbf{x_{i+1}} = \tilde{X}^\delta_T$
  end for
  Return $\frac{1}{N_s}\sum_{i=1}^{N_s} d_i$
```

When N_s is sufficiently large, $\mathbf{x_1}, \cdots, \mathbf{x_{N_s}}$ are from a long trajectory of the time-T skeleton of $\tilde{X}_T$. Hence they are approximately sampled from $\hat{\mu}_X$. The error term d_i for $\tilde{X}^\delta_0 = \hat{Y}^\delta_0 = \mathbf{x_i}$ estimates $d(\tilde{X}_T, \hat{Y}_T)$. A similar coupling argument shows that the output of Algorithm 5 is an upper bound of $d_w(\hat{\mu}_X P_X^T, \hat{\mu}_X P_Y^T)$.

For each initial value $x \in \mathbb{R}^d$, denote

$$\theta_x = \mathbb{E}_x[d(\tilde{X}_T^\delta, \hat{Y}_T^\delta) \,|\, \tilde{X}_0^\delta = \hat{Y}_0^\delta = x, \tau > T]\,. \tag{4.2.2}$$

Similar as in Case 1, the following lemma gives an upper bound for the finite time error $d_w(\mu_X P_X^T, \mu_X P_Y^T)$.

Lemma 4.2.2. *For the Wasserstein distance induced by the distance given in (2.4.1), we have*

$$d_w(\hat{\mu}_X P_X^T, \hat{\mu}_X P_Y^T) \le \int \mathbb{P}_x(\tau < T)\hat{\mu}_X(dx) + \int \theta_x \hat{\mu}_X(dx),$$

where x is the initial value with distribution $\hat{\mu}_X$ and θ_x .

Proof. Note that $\hat{\mu}_X^2(P_X^T \times P_Y^T)$ is a coupling of $\hat{\mu}_X P_X^T$ and $\hat{\mu}_X P_Y^T$. From the definition of the Wasserstein distance, we have

$$\begin{aligned} d_w(\hat{\mu}_Y P_X^T, \hat{\mu}_Y P_Y^T) &\le \int_{\mathbb{R}^d \times \mathbb{R}^d} d(x,y) \hat{\mu}_X^2(P_X^T \times P_Y^T)(\mathrm{d}x\mathrm{d}y) \\ &= \int_{\mathbb{R}^d} \mathbb{E}_{(x,x)}[d(\tilde{X}_T^\delta, \hat{Y}_T^\delta)]\hat{\mu}_X(dx) \\ &= \int_{\mathbb{R}^d} \mathbb{P}_x(\tau < T) d(\tilde{X}_T^\delta, \hat{Y}_T^\delta)\hat{\mu}_X(dx) + \int_{\mathbb{R}^d} \mathbb{P}_x(\tau > T) d(\tilde{X}_T^\delta, \hat{Y}_T^\delta)\hat{\mu}_X(dx) \\ &\le \int_{\mathbb{R}^d} \mathbb{P}_x(\tau < T)\hat{\mu}_X(dx) + \int \theta_x \hat{\mu}_X(dx) \end{aligned}$$

according to the definition of θ_x. □

In summary, the sensitivity of QSD depends on both mixing rate of the modified process and the killing probability. Both higher mixing rate and lower killing probability per unit time lead to a more robust QSD that is not sensitive against small change of the dynamics or the boundary condition. More precisely, let T be the constant time we choose and α be the contraction of operator P_Y^T.

(1) If the modified process $\hat{Y}$ only differs from $\hat{X}$ with a reflection, then we have

$$d_w(\hat{\mu}_X, \hat{\mu}_Y) \le (1-\alpha)^{-1} \int_{\mathbb{R}^d} \mathbb{P}_x[\tau < T]\hat{\mu}_X(\mathrm{d}x)\,.$$

(2) If the demographic noise is removed from $\hat{Y}$, then

$$d_w(\hat{\mu}_X, \hat{\mu}_Y) \le (1-\alpha)^{-1} \left\{ \int_{\mathbb{R}^d} \mathbb{P}_x(\tau < T)\hat{\mu}_X(dx) + \int_{\mathbb{R}^d} \theta_x \hat{\mu}_X(dx) \right\} ,$$

where θ_x is defined in equation (4.2.2).

4.3 Numerical Results

4.3.1 Sensitivity of QSD: 1D examples

In this section, we use 1D examples to study the sensitivity of QSDs against changes on boundary conditions. Consider an 1D gradient flow of the potential function $V(x)$ with an additive noise perturbation

$$X_t = -V'(X_t)dt + \sigma dW_t. \tag{4.3.1}$$

Let $(-\infty, 0]$ be the absorbing state of X_t. So if $V(0) < \infty$, X_t admits a QSD, denoted by μ_X. If we let the stochastic process reflect at $x = 0$, the modified stochastic process, denoted by Y_t, admits an invariant probability measure denoted by μ_Y. We will compare the sensitivity of μ_X against the change of boundary condition for two different cases whose speed of mixing are different, namely a single well potential function and a double well potential function.

We choose a single well potential function $V_1(x) = (x-1)^2$ and a double well potential function $V_2(x) = x^4 - 4\sqrt{2}x^3 + 10x^2 - 4\sqrt{2}x + 1$. The values of minima of both V_1 and V_2 are zero. The values of V_1 and V_2 at the absorbing state are 1. And the height of the barrier between two local minima of V_2 is 1. The strength of noise is $\sigma = 0.7$ in both examples. See Figure 4.1 middle column for plots of these two potential functions. In both cases, the QSD and the invariant probability measure are computed on the domain $D = [0, 3]$. To further distinguish these two cases, we denote the QSD of equation (4.3.1) with absorbing state $x = 0$ and potential function $V_1(x)$ (resp. $V_2(x)$) by μ_X^1 (resp. μ_X^2) and the invariant probability measure of equation (4.3.1) with reflection boundary at $(-\infty, 0]$ and potential function $V_1(x)$ (resp. $V_2(x)$) by μ_Y^1 (resp. μ_Y^2). Probability measures μ_X^1 and μ_Y^1 (resp. μ_X^2

and μ_Y^2) are compared in Figure 4.1 right column. We can see that the QSD and the invariant probability measure have small difference for the single well potential function V_1. But they look very different for the double well potential function V_2. With the double well potential function, there is a visible difference between probability density functions of μ_X^2 and μ_Y^2. The density function of QSD is much smaller than the invariant probability measure around the left local minimum $x = 1 - \sqrt{2}$ because this local minimum is closer to the absorbing set, which makes killing and regeneration more frequent when X_t is near this local minimum. In other words, the QSD of equation (4.3.1) with respect to the double well potential is very sensitive against the change at the boundary.

The reason of the high sensitivity is illustrated by the coupling argument. We first run Algorithm 3 with 8 independent long trajectories with length of 10^6 and collect the coupling times. The slope of exponential tail of the coupling time gives the rate of contraction of P_Y^T. The $\mathbb{P}(\tau^c > t)$ versus t plot is demonstrated in log-linear plot in Figure 4.1 left column. The slope of exponential tail is $\gamma = 2.031414$ for the single well potential V_1, and $\gamma = 0.027521$ for the double well potential case. Then we run Algorithm 4 to estimate the finite time error $\mathrm{d}_w(\mu_X P_X^T, \mu_X P_Y^T)$ for both cases. Since the single well potential case has a much faster coupling speed, we can choose $T = 0.5$. The output of Algorithm 4 is $d_w(\mu_X P_X^T, \mu_X P_Y^T) \approx 0.00391083$. This gives an estimate $d_w(\mu_X, \mu_Y) \approx 0.0061$. The double well potential case converges much slower. We choose $T = 20$ to make sure that the denominator $1 - e^{-\gamma T}$ is not too small. As a result, Algorithm 4 gives an approximation $d_w(\mu_X P_X^T, \mu_X P_Y^T) \approx 0.06402$, which means $d_w(\mu_X, \mu_Y) \approx 0.1512$. This is consistent with the right column seen in Figure 4.1, the QSD of the double well potential is much more sensitive against a change of the boundary condition than the single well potential case.

4.3.2 Lotka-Volterra Competitive Dynamics

In this example, we focus on the effect of demographic noise on the classical Lotka-Volterra competitive system. The Lotka-Volterra competitive system with some environ-

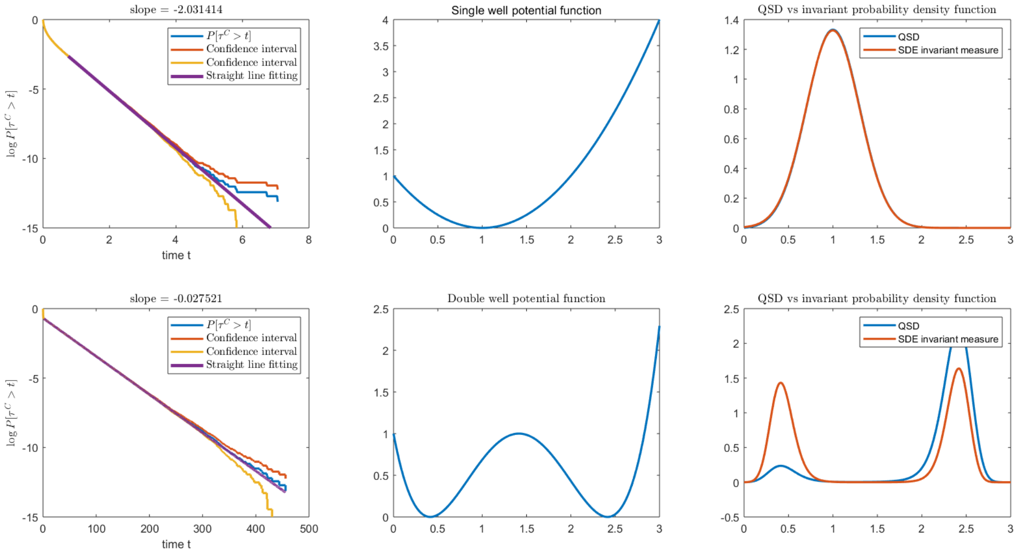

Figure 4.1: (Single well vs. Double well potential function) **Left column**: $\mathbb{P}(\tau^c > t)$ vs. t. **Middle column:** Single well and double well potential functions. **Right column**: QSD vs. invariant density function.

mental fluctuation has the form

$$\begin{aligned} dY_1(t) &= Y_1(t)(l_1 - a_{11}Y_1(t) - a_{12}Y_2(t))dt + \sigma_1' Y_1(t) dW_1(t), \\ dY_2(t) &= Y_2(t)(l_2 - a_{22}Y_1(t) - a_{21}Y_1(t))dt + \sigma_2' Y_2(t) dW_2(t). \end{aligned} \tag{4.3.2}$$

Here $l_i > 0$ is the per-capita growth rate of species i and $a_{ij} > 0$ is the per-capita competition rate between species i and j. More details can be found in [21]. Model parameters are chosen to be $l_1 = 2, l_2 = 4, a_{11} = 0.8, a_{12} = 1.6, a_{21} = 1, a_{22} = 5$. Let $\partial\mathcal{X}$ be the union of x-axis and y-axis. For suitable σ_1' and σ_2', Y_1 and Y_2 can coexist such that the probability of (Y_1, Y_2) hits $\partial\mathcal{X}$ in finite time is zero. So equation (4.3.2) admits an invariant probability measure, denoted by μ_Y.

As a modification, we add a small demographic noise term to equation (4.3.3). The equation becomes

$$\begin{aligned} dX_1(t) &= X_1(t)(l_1 - a_{11}X_1(t) - a_{12}X_2(t))dt + \sigma_1 X_1(t)dW_1(t) + \epsilon\sqrt{X_1(t)}dW_1'(t), \\ dX_2(t) &= X_2(t)(l_2 - a_{22}X_1(t) - a_{21}X_1(t))dt + \sigma_2 X_2(t)dW_2(t) + \epsilon\sqrt{X_2(t)}dW_2'(t). \end{aligned} \tag{4.3.3}$$

It is easy to see that equation (4.3.3) can exit to the boundary $\partial\mathcal{X}$. It admits a QSD, denoted by μ_X.

In order to study the effect of demographic noise, we compare μ_Y, the numerical invariant measure of equation (4.3.2), and μ_X, the QSD of equation (4.3.3). In our simulation, we fix the strength of demographic noise as $\epsilon = 0.05$ and compare μ_X and μ_Y at two different levels of the environment noise $\sigma_1 = \sigma_2 = 0.75$ and $\sigma_1 = \sigma_2 = 1.1$ respectively. The coefficient σ_1' and σ_2' in equation (4.3.2) satisfies $\sigma_i' = \sqrt{\sigma_i^2 + \epsilon^2}$ for $i = 1, 2$ to match the effect of the additional demographic noise. Compare Figure 4.2 and Figure 4.3, one can see that μ_Y has significant concentration at the boundary when $\sigma_1 = \sigma_2 = 1.1$.

The result for $\sigma_1 = \sigma_2 = 0.75$ is shown in Figure 4.2. Left bottom of Figure 4.2 shows the invariant measure. The QSD is shown on right top of Figure 4.2. The total variation distance between these two measures are shown at the bottom of Figure 4.2. The difference is very small and it just appears around boundary. This is reasonable because with high probability, the trajectories of equation (4.3.3) moves far from the absorbing set $\partial\mathcal{X}$ in both cases, which makes the regeneration events happen less often. This is consistent with the result of Lemma 4.2.2. We compute the distribution of the coupling time. The coupling time distribution and its exponential tail are shown in Figure 4.2 Top Left. Then we use Algorithm 5 to compute the finite time error. To better match two trajectories given by equations (4.3.2) and (4.3.3), we separate the noise term in equation (4.3.2) into $\sigma_i' Y_i(t)\mathrm{d}W_i(t) = \sigma_i Y_i(t)\mathrm{d}W_i^{(1)}(t) + \epsilon Y_i(t)\mathrm{d}W_i^{(2)}(t)$ for $i = 1, 2$, where $W_i^{(1)}(t)$ and $W_i^{(2)}(t)$ use the same Wiener process trajectory as $W_i(t)$ and $W_i'(t)$ in equation (4.3.3) for $i = 1, 2$. Let $T = 4$. The finite time error caused by the demographic noise is 0.01773. As a result, the upper bound given in Lemma 4.2.2 is 0.02835. Note that as seen in Figure 4.2, this upper bound actually significantly overestimates the distance between the invariant probability measure and the

QSD. The empirical total variation distance is much smaller that our theoretical upper bound. This is because the way of matching $\sigma_i' Y_i \mathrm{d}W_i(t)$ and $\sigma_i X_i(t)\mathrm{d}W_i(t) + \epsilon\sqrt{X_i(t)}\mathrm{d}W_i(t)$ is very rough. A better approach of matching those noise terms will likely lead to a more accurate estimation of the upper bound of the error.

The results for $\sigma_1 = \sigma_2 = 1.1$ are shown in Figure 4.3. The total variation distance between these two measures are shown at the bottom of Figure 4.3. It is not hard to see the difference is significantly larger than case $\sigma = 0.75$. The reason is that trajectories of equation (4.3.3) have high probability moving along the boundary in this parameter setting. This makes the probability of falling into the absorbing set $\partial\mathcal{X}$ much higher. Same as above, we compute the distribution of the coupling time and demonstrate the coupling time distribution (as well as the exponential tail) in Figure 4.3 Top Left. The coupling in this example is slower so we choose $T = 12$ to run Algorithm 5. The probability of killing before T is approximately 0.11186 and the total finite time error caused by the demographic noise is 0.06230. As a result, the upper bound given in Lemma 4.2.2 is 0.1356. This is consistent with the numerical finding shown in Figure 4.3 Bottom Right.

4.3.3 Chaotic attractor

In this section, we consider a non-trivial 3D example that has interactions between chaos and random perturbations, called the Rossler oscillator. The random perturbation of the Rossler oscillator is

$$\begin{cases} dx = (-y - z)dt + \epsilon dW_t^x \\ dy = (x + ay)dt + \epsilon dW_t^y \\ dz = (b + z(x - c))dt + \epsilon dW_t^z, \end{cases} \tag{4.3.4}$$

where $a = 0.2, b = 0.2, c = 5.7$, and W_t^x, W_t^y and W_t^z are independent Wiener processes. The strength of noise is chosen to be $\epsilon = 0.1$. This system is a representative example of chaotic ODE systems appearing in many applications of physics, biology and engineering. We consider equation (4.3.4) restricted to the box $D = [-15, 15] \times [-15, 15] \times [-1.5, 1.5]$.

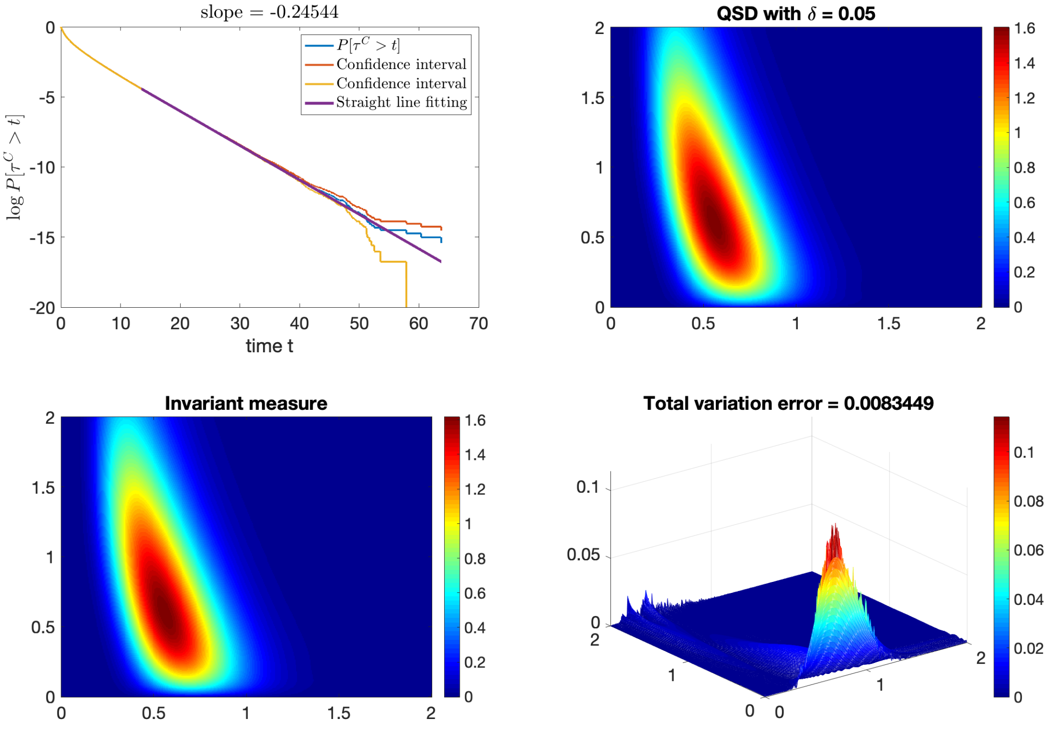

Figure 4.2: (Case $\sigma_1 = \sigma_2 = 0.75$) **Upper panel**: (**Left**) $\mathbb{P}(\tau^c > t)$ vs.t. (**Right**) QSD with demographic noise coefficient $\epsilon = 0.05$. **Lower panel**: (**Left**) Invariant density function for $\sigma = 0.75$. (**Right**) Total variation of QSD and invariant density function.

Therefore, it admits a QSD supported by D. In this example, a grid with $1024 \times 1024 \times 128$ mesh points is constructed on D.

It is very difficult to use traditional PDE solver to compute a large scale 3D problem. To analyze the QSD of this chaotic system, we apply the blocked version of the Fokker-Planck solver studied in [14]. More precisely, a big mesh is divided into many "blocks". Then we solve the optimization problem (3.2.2) in parallel. The collaged solution is then processed by the "shifting block" technique to reduce the interface error, which means the blocks are reconstructed such that the center of new blocks cover the boundary of old blocks. Then we let the solution from the first found serve as the reference data, and solve optimization problem (3.2.2) again based on new blocks. See [14] for the full details of implementation. In this example, the grid is further divided into $32 \times 32 \times 4$ blocks. We run the "shifting block"

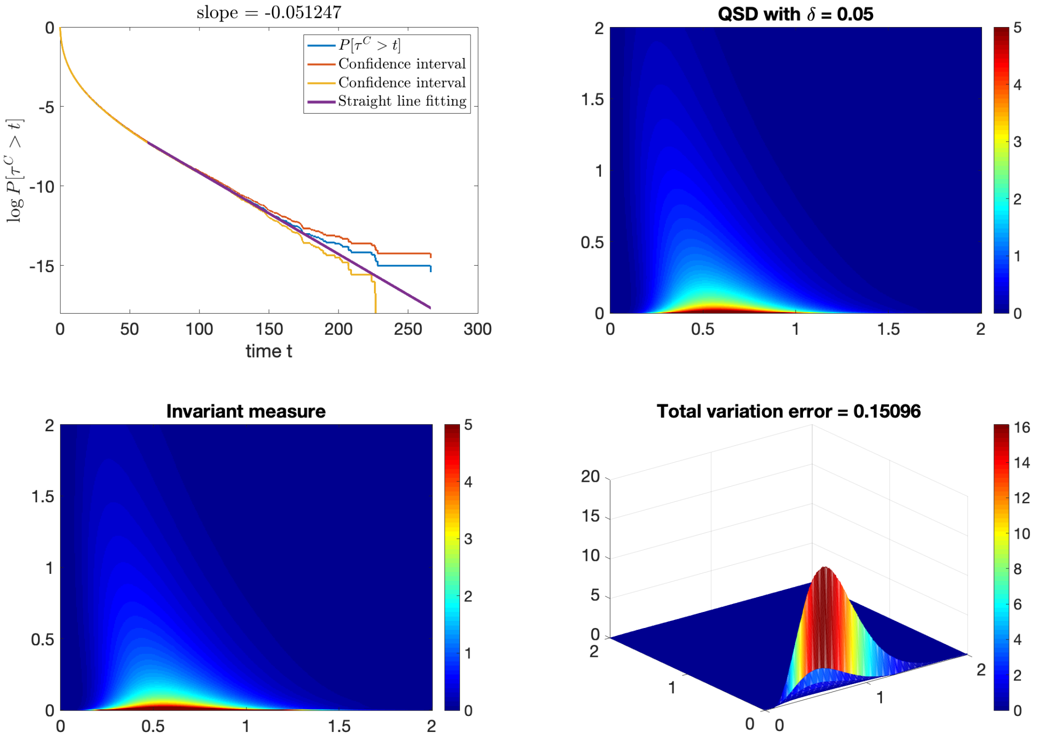

Figure 4.3: (Case $\sigma_1 = \sigma_2 = 1.1$) **Upper panel**: (**Left**) $\mathbb{P}(\tau^c > t)$ vs.t. (**Right**) QSD with demographic noise coefficient $\epsilon = 0.05$. **Lower panel**: (**Left**) Invariant density function for $\sigma = 1.1$. (**Right**) Total variation of QSD and invariant density function.

solver for 3 repeats to eliminate the interface error. The reference solution is generated by a Monte Carlo simulation with 10^9 samples. The killing rate is $\lambda = -0.473011$. Two "slices" of the solution, as seen in Figure 4.4, are then projected to the xy-plane for the sake of easier demonstration. See the caption of Figure 4.4 for the coordinates of these two "slices". The left picture in Figure 4.4 shows the projection of the solution has both dense and sparse parts that are clearly divided. An outer "ring" with high density appears and the density decays quickly outside this "ring." The right picture in Figure 4.4 demonstrates the solution has much lower density when z-coordinate is larger than 1.

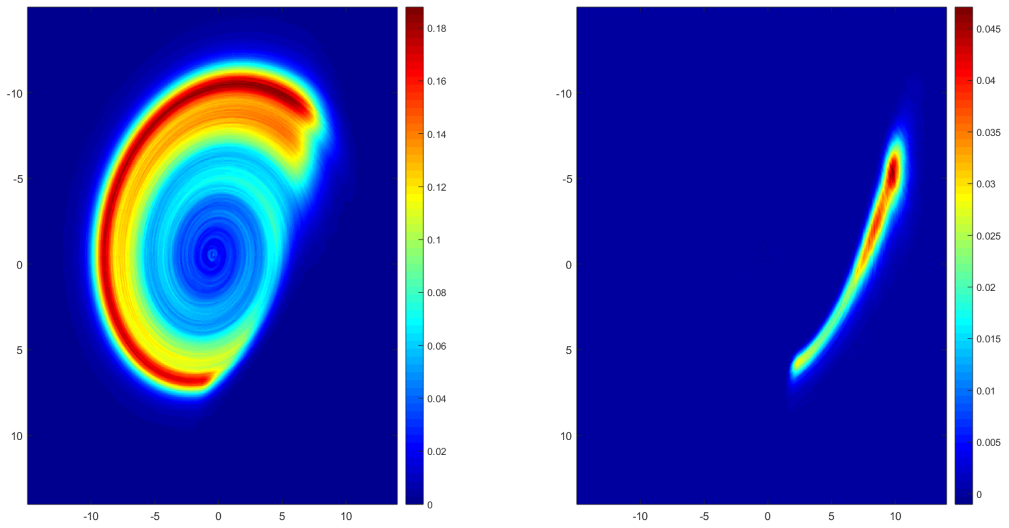

Figure 4.4: (Rossler) Projections of 2 ”slices” of the QSD of the Rossler system to the xy-plane. z-coordinates of 2 slices are [-0.0352, 0.2578], [1.1367, 1.4297]. The solution is obtained by a balf-block shift solver on $[-15, 15] \times [-15, 15] \times [-1.5, 1.5]$ with $1024 \times 1024 \times 128$ mesh points, $32 \times 32 \times 4$ blocks, and 10^9 samples.

4.4 Conclusion

In this chapter we provide some data-driven methods for the computation of quasi-stationary distributions (QSDs) and the sensitivity analysis of QSDs. Both of them are extended from the first author's earlier work about invariant probability measures. When using the Fokker-Planck equation to solve the QSD, we find that the idea of using a reference solution with low accuracy to set up an optimization problem still works well for QSDs. And the QSD is not very sensitively dependent on the killing rate, which is given by the Monte Carlo simulation when producing the reference solution. The data-driven Fokker-Planck solver studied in this dissertation is still based on discretization. But we expect the mesh-free Fokker-Planck solver proposed in [14] to work for solving QSDs. In the sensitivity analysis part, the focus is on the relation between a QSD and the invariant probability measure of a “modified process”, because many interesting problems in applications fall into this category. The sensitivity analysis needs both a finite time truncation error and a contraction rate of the Markov transition kernel. The approach of estimating the finite time truncation error is standard. The contraction rate is estimated by using the novel numerical coupling approach developed in [31]. The sensitivity analysis of QSDs can be extended to other settings, such as the sensitivity against small perturbation of parameters, or the sensitivity of a chemical reaction process against its diffusion approximation. We continue to study sensitivity analysis related to QSDs in next chapter.

CHAPTER 5

SENSITIVITY ANALYSIS OF MASS-ACTION SYSTEMS

5.1 Poisson processes and diffusion processes

Let $\mathbf{X} = \{X(t)\}$ and $\hat{\mathbf{X}} = \{\hat{X}_n\}$ (resp. $\mathbf{Y} = \{Y(t)\}$ and $\hat{\mathbf{Y}} = \{\hat{Y}_n\}$) be the stochastic process given by (2.5.3) (resp. (2.5.4)) and a numerical approximation with step size h, respectively. Recall that two approximations are

$$X(t) = X(0) + \sum_k \frac{l_k}{V} P_k(V \int_0^t f_k(X(s))ds)$$

and

$$Y(t) = Y(0) + \sum_k \frac{l_k}{V} \left[V \int_0^t f_k(Y(s))ds + B_k \left(V \int_0^t f_k(Y(s))ds \right) \right]$$

where where $P_k(t),\ k = \{1, \cdots, K\}$ are independent unit-rate Poisson processes, and $l_k = c_k' - c_k \in \mathbb{R}^d$ denotes the coefficient change of moleculars at reaction k and the $B_k(\cdot)$ are independent standard Wiener processes.

Needless to say a diffusion process is much easier to study than a Poisson process with jumps. One natural question here is that how much the long time dynamics of $\mathbf{X}$ is preserved by its diffusion approximation. This problem is more complicated than it looks because both $\mathbf{X}$ and $\mathbf{Y}$ have natural domain $\mathbb{R}^d_+$. When the number of moleculars of one species reaches 0, the process exits from its domain due to extinction. It is common for equation (2.5.3) or equation (2.5.4) to have finite time extinction. To see this, consider the 1D version of equation (2.5.4):

$$\mathrm{d}Y(t) = f(Y(t))\mathrm{d}t + \frac{1}{\sqrt{V}} \sqrt{f(Y(t))} \mathrm{d}B_t \,. \tag{5.1.1}$$

Let $H(x) = x^{-1}$ be a test function. Applying Ito's formula then take the expectation, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[H(Y(t))] = f(Y(t))\left(\frac{2}{(Y(t))^3} - \frac{1}{(Y(t))^2}\right) .$$

If $f(Y(t)) = cY(t)$ for a constant c, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[H(Y(t))] \geq 2c\left(\mathbb{E}[H(Y(t))]\right)^2 ,$$

which blows up to ∞ in finite time. Hence $Y(t)$ has strictly positive extinction probability in finite time. The calculation above fits the setting of many mass-action systems.

Therefore, to prevent finite time extinction, usually one needs constant influx of each species. That is why often we need to study the quasi-stationary distribution (QSD) instead of the invariant probability distribution. Below we introduce the QSD and its sampling method only for $\mathbf{X}$, as the case of $\mathbf{Y}$ is analogous.

Let $\partial\mathcal{X} = \mathbb{R}^d \setminus \mathbb{R}^d_+$ be the absorbing set of $\mathbf{X}$. The quasi-stationary distribution (QSD) is an invariant probability measure conditioning on $\mathbf{X}$ has not hit the absorbing set yet. We further define

$$\tau = \inf\{t > 0 : X(t) \in \partial\mathcal{X}\}$$

as the first passage time to $\partial\mathcal{X}$.

Remark 5.1.1. *If the first passage time of $\boldsymbol{X}$ to $\partial\mathcal{X}$ is ∞ with probability one, $\{\tau > t\}$ is the full probability space. As a result, QSD in equation* (2.1.1) *becomes the invariant probability measure and QLD in equation* (2.1.2) *becomes the limiting probability measure (which is also invariant). Therefore, when the mass action system admits an invariant probability measure instead of a QSD, all our arguments and algorithms still apply.*

When we define the numerical processes (2.5.6) and (2.5.7), we need to specify the rule of regeneration such that they both sample from QSDs as the time approaches to infinity. To sample from QSD, we need to regenerate a sample once it hits the absorbing set. Therefore, in addition to $\hat{X}_n$, we also need to update a temporal occupation measure

$$\mu_n = \frac{1}{n} \sum_{k=0}^{n-1} \delta_{\hat{X}_k} \, .$$

If the numerical scheme gives $\hat{X}_{n+1} \in \partial \mathcal{X}$, we immediately regenerate $\hat{X}_{n+1}$ from μ_n. More precisely, let the transition kernel of the numerical scheme of $\hat{X}_n$ (without resampling) be $\hat{Q}$. Then $\hat{Q}$ has an absorbing set $\partial \mathcal{X}$ such that $\hat{Q}(\partial \mathcal{X}, \partial \mathcal{X}) = 1$. The transition kernel of $\hat{X}_n$ is the sum of $\hat{Q}$ and the regeneration measure such that

$$\mathbb{P}[\hat{X}_{n+1} \in A \,|\, \hat{X}_n = x] = \hat{Q}(x, A) + \hat{Q}(x, \partial \mathcal{X}) \mu_n(A) \, .$$

The following convergence result follows from [6].

Proposition 5.1.1 (Theorem 2.5 in [6]). *Let $\hat{\mu}$ be the QSD of the numerical process $\hat{X}_n$. Under suitable assumptions about $\hat{X}_n$, the occupation measure μ_n converges to the QSD $\hat{\mu}$ as $n \to \infty$.*

To study the sensitivity of diffusion approximation, we also need a theoretical process $\tilde{\mathbf{X}} = \{\tilde{X}_n\}$ that directly regenerate from the QSD $\hat{\mu}$ once exit to the boundary. Recall that $\hat{Q}$ is the transition kernel of $\hat{X}_n$ (without resampling). The transition kernel of $\tilde{\mathbf{X}}$ is

$$\tilde{P}(x, \cdot) = \hat{Q}(x, \cdot) + \hat{Q}(x, \partial \mathcal{X}) \hat{\mu}(\cdot)$$

for all $x \in \mathbb{R}^d_+$. Note that $\hat{X}_n$ is not a Markov process (but $(\hat{X}_n, \mu_n)$ is a Markov process). But $\tilde{\mathbf{X}}$ is a homogeneous Markov process with an invariant probability measure μ. The case of $Y(t)$ is analogous. We denote the numerical process that resample from a temporal occupation measure by $\hat{\mathbf{Y}} = \{\hat{Y}_n\}$, and the Markov process that directly resample from QSD by $\tilde{\mathbf{Y}} = \{\tilde{Y}_n\}$.

5.2 Decomposition of error term

Let P_X and $\tilde{P}_X$ be the transition kernels of $X(t)$ and $\tilde{X}_n$ respectively. Let P_Y and $\tilde{P}_Y$ be that of $Y(t)$ and $\tilde{Y}_n$ respectively. Denote the QSDs of $X(t)$, $\tilde{X}_n$, $Y(t)$ and $\tilde{Y}_n$ by π_X, $\hat{\pi}_X$, π_Y, and $\hat{\pi}_Y$, respectively. The quantity that we are interested in is $d_w(\pi_X, \pi_Y)$.

Let T be a fixed constant. Motivated by [23], the following decomposition follows easily by the triangle inequality and the invariance.

$$d_w(\pi_X, \pi_Y) \leq d_w(\pi_X, \hat{\pi}_X) + d_w(\hat{\pi}_X, \hat{\pi}_Y) + d_w(\hat{\pi}_Y, \pi_Y) \tag{5.2.1}$$

The sensitivity of invariant probability against time discretization has been addressed in [15]. When the time step size of the time discretization is small enough, the invariant probability measure π_Y is close to the numerical invariant probability measure $\hat{\pi}_Y$. The case of QSD is analogous. Hence the third term $d_w(\pi_Y, \hat{\pi}_Y)$ is proportional to step size h. The estimation of the first term $d_w(\pi_X, \hat{\pi}_X)$ can be obtained by some linear algebraic calculation.

Theorem 5.2.1. *Let $X(t)$ be a continuous time Markov chain with finite state space and $\hat{X}$ be its τ-leaping approximation with step size h. Suppose that π and $\hat{\pi}$ be the true QSD and the numerical approximation of QSD respectively. Let h be the time step size of numerical process. If the generating matrix of $X(t)$ is irreducible, then*

$$\|\pi - \hat{\pi}\| \sim O(h)$$

for $0 < h \ll 1$.

Proof. This proof follows the standard argument of eigenvector perturbation result. The case of stationary distribution is proved in [34]. Here we follow the argument in [12] to prove a similar result for QSDs. Let Q be the generating matrix of $X(t)$. Because π is true QSD and $\hat{\pi}$ is the numerical approximation of QSD, we have

$$\pi^T e^{hQ} = \lambda \pi^T ,\ \hat{\pi}^T (I + hQ) = \hat{\lambda} \hat{\pi}^T ,$$

where λ and $\hat{\lambda}$ are simple eigenvalues. Define a function

$$A(t) \overset{\text{def}}{=} I + hQ + tR(h),$$

where $R(h)$ is an $O(1)$ matrix given by the Taylor expansion $e^{hQ} = I + hQ + h^2 R(h)$. Then we have $A(0) = I + hQ$ and $A(h^2) = e^{hQ}$. Note that $A(0)$ is irreducible for all sufficiently small h because Q is also irreducible. Let $\pi(t)$ be the first eigenvector of $A(t)$ normalized to 1 in l_1 norm. Then the sensitivity of π is reduced to the derivative of $A(t)$.

Since π is normalized to 1 in l_1 norm, it follows from [12] Section 3 that

$$\pi'(0) = S^\sharp A'(0) \pi(0) \, ,$$

where $S = \lambda I - A(0)$, and $S^\sharp$ is the group inverse of S. (We refer [12] for further discussion of the group inverse and derivative of Perron vector.)

When h is small, we have $1 - \lambda = O(h)$. Hence $S = I - O(h) - I - hQ$ is an $O(h)$ small matrix. This means $S^\sharp = O(h^{-1})$. In addition $A'(0) = R = O(1)$ by definition. Hence $\pi'(0) = O(h^{-1})$. Since $\hat{\pi} = \pi(h^2)$, we have

$$\|\pi - \hat{\pi}\| = O(h^{-1}) \times O(h^2) = O(h).$$

This completes the proof.

□

Therefore, we have that $d_w(\pi_X, \hat{\pi}_X) = O(h)$ and $d_w(\pi_Y, \hat{\pi}_Y) = O(h)$, which make the second error term be the key part. The second error term is the difference between numerical Poisson process of a mass-action system and its corresponding numerical diffusion process.

Proposition 5.2.1. *Let $T > 0$ be a fixed constant. We can decompose $d_w(\hat{\pi}_X, \hat{\pi}_Y)$ via the following inequality:*

$$d_w(\hat{\pi}_X, \hat{\pi}_Y) \leq d_w(\hat{\pi}_X \tilde{P}_X^T, \hat{\pi}_X \tilde{P}_Y^T) + d_w(\hat{\pi}_X \tilde{P}_Y^T, \hat{\pi}_Y \tilde{P}_Y^T) \tag{5.2.2}$$

The term $d_w(\hat{\pi}_X \tilde{P}_X^T, \hat{\pi}_X \tilde{P}_Y^T)$ is the finite time error and the term $d_w(\hat{\pi}_X \tilde{P}_Y^T, \hat{\pi}_Y \tilde{P}_Y^T)$ can be bounded by coupling methods.

There are two different ways to think about the distance $d_w(\hat{\pi}_X, \hat{\pi}_Y)$. One method is considering $\hat{\pi}_X$ and $\hat{\pi}_Y$ as conditional distributions on set $\mathbb{R}_+^d / \partial \mathcal{X}$, i.e. $\hat{\pi}_X(A) = \{\hat{X} \in A | t < \tau_X\}$ and $\hat{\pi}_Y(A) = \{\hat{Y} \in A | t < \tau_Y\}$, where τ_X and τ_Y are the killing time for processes $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$, respectively. The other way is to use the $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ that regenerate from QSDs. No conditioning is needed as $\hat{\mu}_X$ and $\hat{\mu}_Y$ are now the invariant probability measures of $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ respectively. There are some fundamental difficulty when computing the finite time error because it is hard to couple $\hat{X}_n$ and $\hat{Y}_n$ when one regenerates while the other does not. Hence we choose to use $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ instead.

5.3 Finite time error

We consider the modified processes $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$, which are regenerated from the corresponding QSDs when they hit the boundary. Let $\hat{\pi}_X$ and $\hat{\pi}_Y$ be the invariant measures of $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$. Let $\tilde{\Gamma}(\mathrm{d}x, \mathrm{d}y) = \hat{\pi}_X^2(\tilde{P}_X^T \times \tilde{P}_Y^T)$, where $\hat{\pi}_X^2$ is the coupled measure of $\hat{\pi}_X$ on the "diagonal" of $\mathbb{R}^d \times \mathbb{R}^d$ that is supported by the hyperplane $\{(x, y) \in \mathbb{R}^{2d} | y = x\}$ such that $\hat{\pi}_X^2(\{(x, x) | x \in A\}) = \hat{\pi}_X(A)$, and $\tilde{P}_X^T \times \tilde{P}_Y^T$ is any coupled process such that two marginal processes are $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ respectively. The following proposition follows easily.

Proposition 5.3.1. *Let $(\tilde{X}_n, \tilde{Y}_n)$ be a coupling of $\tilde{X}_n$ and $\tilde{Y}_n$ with transition kernel $\tilde{P}^T_X \times \tilde{P}^T_Y$, then*

$$d_w(\hat{\pi}_X \tilde{P}^T_X, \hat{\pi}_X \tilde{P}^T_Y) \leq \mathbb{E}_{\hat{\pi}_X}[d(\tilde{X}_T, \tilde{Y}_T)] \,.$$

Proof. By the definition of Wasserstein distance

$$\begin{aligned} d_w(\hat{\pi}_X \tilde{P}^T_X, \hat{\pi}_X \tilde{P}^T_Y) &\leq \int_{\mathbb{R}^d \times \mathbb{R}^d} d(x,y) \hat{\pi}^2_X (\tilde{P}^T_X \times \tilde{P}^T_Y)(dx, dy) \\ &= \int_{\mathbb{R}^d} \mathbb{E}_{(x,x)} d(\tilde{X}_T, \tilde{Y}_T) \hat{\pi}_X(dx) = \mathbb{E}_{\hat{\pi}_X}[d(\tilde{X}_T, \tilde{Y}_T)] \,. \end{aligned}$$

□

The key of estimating the finite time error effectively is to create a good coupled process $(\tilde{X}_n, \tilde{Y}_n)$. That is why we need to use the KMT algorithm to generate matching Wiener process and Poisson processes. Here it remains to define how $\tilde{X}_n$ and $\tilde{Y}_n$ couple when they regenerate from QSDs. Since we do not have QSD *in priori*, we will use $\hat{X}_n$ and $\hat{Y}_n$ to approximate $\tilde{X}_n$ and $\tilde{Y}_n$. In other words, we regenerate samples from the temporal occupation measure. To minimize error during sample regeneration, we define the following coupled process $(\hat{X}_n, \mu^X_n)$ and $(\hat{Y}_n, \mu^Y_n)$, such that $\hat{X}_n$ and $\hat{Y}_n$ follows equations (2.5.6) and (2.5.7) respectively by using paired processes $B_k(t)$ and $P_k(t)$ for each k, and μ^X_n, μ^Y_n are two occupation measures. $S = (Z_1, \cdots, Z_N)$ (N is large enough) is a finite sequence of uniform random variables on $(0, 1)$. Let N_X and N_Y are the total number of regenerations up to time n. In other words when $\tilde{X}_{n+1}$ enters $\partial\mathcal{X}$ at step n and needs regeneration, we increase N_X by one and choose the N_X-th element of S, Z_{N_X} to regenerate $\tilde{X}_{n+1}$, by letting $\tilde{X}_{n+1} = \tilde{X}_{\lfloor Z_{N_X} n \rfloor}$. Then it is easy to see that $(\hat{X}_n, \mu^X_n)$ and $(\hat{Y}_n, \mu^Y_n)$ is a Markov coupling and the marginal processes $(\hat{X}_n, \hat{Y}_n)$ is a coupling of equations (2.5.6) and (2.5.7).

Details of computation are shown in Algorithm 6. When N is large, initial values $\hat{X}^1_1, \cdots, \hat{X}^M_1$ in Algorithm 6 are from a trajectory of the time-T skeleton of $\hat{X}^T$. Hence

Algorithm 6 Estimate finite time error

```
Input: Initial value $\hat{X}_0$
Output: An estimator of $d_w(\hat{\pi}_X^T \tilde{P}_X^T, \hat{\pi}_Y^T \tilde{P}_Y^T)$
    Set initial value $\hat{X}_1^1 = \hat{Y}_1^1$
    Generate a sequence of uniformly distributed random variable S
    for $m = 1$ to $M$ do
      Using the KMT algorithm to generate paired trajectories $\{P_k\}$ and $\{B_k\}$
      If $m \neq 1$, reset initial value $\hat{X}_1^m = \hat{Y}_1^m = \hat{X}_T^{m-1}$
      Let $N_X = N_Y = 0$
      for $n = 1$ to $T$ do
        Update $\hat{X}_{n+1}^m$ and $\hat{Y}_{n+1}^m$ using equations (2.5.6) and (2.5.7) respectively
        if $\hat{X}_{n+1}^m \in \partial\mathcal{X}$ then
          $N_X = N_X + 1$
          Let $\hat{X}_{n+1}^m = \hat{X}_{\lfloor Z_{N_X} n \rfloor}^m$
        end if
        if $\hat{Y}_{n+1}^m \in \partial\mathcal{X}$ then
          $N_Y = N_Y + 1$
          Let $\hat{Y}_{n+1}^m = \hat{Y}_{\lfloor Z_{N_Y} n \rfloor}^m$
        end if
      end for
      Let $d(\hat{X}_T^m, \hat{Y}_T^m) = \min(1, \|\hat{X}_T^m - \hat{Y}_T^m\|)$
    end for
    return $\frac{1}{M} \sum_{m=1}^{M} d(\hat{X}_T^m, \hat{Y}_T^m)$
```

$\hat{X}_1^1, \hat{X}_1^2, \cdots, \hat{X}_1^M$ are approximately sampled from $\hat{\pi}_X$. The error term $d(\hat{X}_T^m, \hat{Y}_T^m)$ evolved from the initial value pair $\hat{X}_1^m = \hat{Y}_1^m = \hat{X}_T^{m-1}$ is recorded. Therefore,

$$\frac{1}{M}\sum_{m=1}^{M} d(\hat{X}_T^m, \hat{Y}_T^m) \tag{5.3.1}$$

is an estimator of

$$\mathbb{E}_{\hat{\pi}_X}[d(\tilde{X}_T, \tilde{Y}_T)], \tag{5.3.2}$$

which is an upper bound of $d_w(\hat{\pi}_X^T \tilde{P}_X^T, \hat{\pi}_X^T \tilde{P}_Y^T)$.

5.4 Coupling inequality and contraction rate

Similar to the coupling inequality of the total variation norm, the distance d we use in this thesis also satisfies the coupling inequality. Let $(Z_t^{(1)}, Z_t^{(2)})$ be a coupling of two stochastic processes and let τ_c be the coupling time. The following Lemma follows easily.

Proposition 5.4.1. *For a Markov coupling $(Z_t^{(1)}, Z_t^{(2)})$, we have*

$$d_w(law(Z_T^{(1)}), law(Z_T^{(2)})) \le \mathbb{P}(Z_T^{(1)} \neq Z_T^{(2)}) = \mathbb{P}(\tau_c > T).$$

Proof. By the definition of the Wasserstein distance,

$$\begin{aligned}
&d_w(\mathrm{law}(Z_T^{(1)}), \mathrm{law}(Z_T^{(2)})) \le \int d(\xi,\eta)\mathbb{P}((Z_T^{(1)}, Z_T^{(2)}) \in (d\xi, d\eta)) \\
&= \int_{\xi\neq\eta} d(\xi,\eta)\mathbb{P}((Z_T^{(1)}, Z_T^{(2)}) \in (d\xi, d\eta)) \\
&\le \int_{\xi\neq\eta} \mathbb{P}((Z_T^{(1)}, Z_T^{(2)}) \in (d\xi, d\eta)) \\
&= \mathbb{P}(Z_T^{(1)} \neq Z_T^{(2)}).
\end{aligned}$$

□

Proposition 5.4.2. *Assume that $d_w(\pi_X, \hat{\pi}_X)$ and $d_w(\pi_Y, \hat{\pi}_Y)$ are in order $O(h)$, then the error*

$$d_w(\pi_X, \pi_Y) \leq \frac{d_w(\hat{\pi}_X \tilde{P}_X^T, \hat{\pi}_X \tilde{P}_Y^T)}{1-\alpha} + O(h),$$

where $\alpha < 1$ is the contraction rate of the transition kernel $\tilde{P}_Y^T$ and $d_w(\hat{\pi}_X \tilde{P}_X^T, \hat{\pi}_X \tilde{P}_Y^T)$ is the finite time error.

Proof. By the triangle inequality,

$$d_w(\pi_X, \pi_Y) \leq d_w(\pi_X, \hat{\pi}_X) + d_w(\hat{\pi}_X, \hat{\pi}_Y) + d_w(\hat{\pi}_Y, \pi_Y).$$

Because both $d_w(\pi_X, \hat{\pi}_X)$ and $d_w(\pi_Y, \hat{\pi}_Y)$ are $O(h)$, we only need to estimate the second term $d_w(\hat{\pi}_X, \hat{\pi}_Y)$. By the triangle inequality again, we have

$$d_w(\hat{\pi}_X, \hat{\pi}_Y) \leq d_w(\hat{\pi}_X \tilde{P}_X^T, \hat{\pi}_X \tilde{P}_Y^T) + d_w(\hat{\pi}_X \tilde{P}_Y^T, \hat{\pi}_Y \tilde{P}_Y^T).$$

If the transition kernel $\tilde{P}_Y^T$ has enough contraction such that

$$d_w(\hat{\pi}_X \tilde{P}_Y^T, \hat{\pi}_Y \tilde{P}_Y^T) \leq \alpha d_w(\hat{\pi}_X, \hat{\pi}_Y)$$

for some $\alpha < 1$, then we have

$$d_w(\hat{\pi}_X, \hat{\pi}_Y) \leq \frac{d_w(\hat{\pi}_X \tilde{P}_X^T, \hat{\pi}_X \tilde{P}_Y^T)}{1-\alpha}. \tag{5.4.1}$$

Hence

$$d_w(\pi_X, \pi_Y) \leq \frac{d_w(\hat{\pi}_X \tilde{P}_X^T, \hat{\pi}_X \tilde{P}_Y^T)}{1-\alpha} + O(h).$$

□

Therefore, in order to estimate $d_w(\pi_X, \pi_Y)$, we need to look for suitable numerical estimators of the finite time error and the speed of contraction of $\tilde{P}_Y^T$. The finite time error can be easily estimated by Algorithm 6. And the speed of contraction α comes from the geometric ergodicity of the Markov process $\hat{\mathbf{Y}}$ is approximated by that of $\hat{\mathbf{Y}}$. If our numerical estimation gives

$$d_w(\hat{\pi}_X \tilde{P}_X^T, \hat{\pi}_X \tilde{P}_Y^T) \approx d_w(\hat{\pi}_X \hat{P}_Y^T, \hat{\pi}_Y \hat{P}_Y^T) \leq Ce^{-\gamma T},$$

then we set $\alpha = e^{-\gamma T}$. Similar as in [15], we use the following coupling method to estimate the contraction rate α. Let $\hat{Z} = (\hat{Y}^{(1)}, \hat{Y}^{(2)})$ be a Markov process in $\mathbb{R}^{2d}$ such that $\hat{Y}^{(1)}$ and $\hat{Y}^{(2)}$ are two copies of $\hat{Y}$. Let the first passage time to the "diagonal" hyperplane $\{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{2d} | \mathbf{y} = \mathbf{x}\}$ be the coupling time. Then by Proposition 5.4.1

$$d_w(\hat{\pi}_X \tilde{P}_Y^T, \hat{\pi}_Y \tilde{P}_Y^T) \leq \mathbb{P}(\tau_c > T).$$

As discussed in [31], we need a hybrid coupling scheme to make sure that two numerical trajectories couple. Under the condition that two trajectories coupled before extinction time, some coupling methods such as reflection coupling or synchronous coupling are implemented in the first phase to bring two trajectories together. Then we compare the probability density function for the next step and couple these two numerical trajectories with the maximal possible probability (called maximal coupling). After doing this for many times, we have many samples of τ_c denote by $\boldsymbol{\tau}_c$. We use the exponential tail of $\mathbb{P}(\tau_c > t)$ to estimate the contraction rate α. We look for a constant $\gamma > 0$ such that

$$-\gamma = \lim_{t \to \infty} \frac{1}{t} \log(\mathbb{P}(\tau_c > t))$$

if the limit exists. See Algorithm 7 for the details of implementation of coupling. Note that we cannot simply compute the contraction rate start from $t = 0$ because only the tail of coupling time can be considered as exponential distributed. Our approach is to check the exponential tail in a log-linear plot. After having $\boldsymbol{\tau}_c$, it is easy to choose a sequence of times $t_0, t_1, \cdots, t_n$ and calculate $n_i = |\{\tau_c^m > t_i | 0 \leq m \leq M\}|$ for each $i = 0, \cdots, n$. Then

$p_i = n_i/M$ is an estimator of $\mathbb{P}_{\hat{\pi}_Y}[\tau_c > t_i]$. Now let p_i^u (resp. p_i^l) be the upper (resp. lower) bound of the confidence interval of p_i such that

$$p_i^u = \tilde{p} + z\sqrt{\frac{\tilde{p}}{\tilde{n}_i}(1-\tilde{p})}\ (resp.\ p_i^l = \tilde{p} - z\sqrt{\frac{\tilde{p}}{\tilde{n}_i}(1-\tilde{p})}),$$

where $z = 1.96$, $\tilde{n}_i = n_i + z^2$ and $\tilde{p} = \frac{1}{\tilde{n}}(n_i + \frac{z^2}{2})$ [1]. If $p_i^l \le e^{-\gamma t_i} \le p_i^u$ for each $0 \le i \le n$, we say that the exponential tail starts at $t = t_{i_0}$. we accept the exponential tail with rate $e^{-\gamma T}$ if the confidence interval $p_{i_0}^u - p_{i_0}^l$ is sufficient small. Otherwise we need to run Algorithm 7 for longer time to eliminate the initial bias in τ_c.

Algorithm 7 Estimation of contraction rate α

```
Input: Initial values $x, y \in \mathcal{X}/\partial\mathcal{X}$
Output: An estimation of contraction rate $\alpha$
  Choose threshold $d > 0$
  for $m = 1$ to $M$ do
    $\tau_c^m = 0, t = 0, (\hat{Y}_0^{(1)}, \hat{Y}_0^{(2)}) = (x, y)$
    Flag = 0
    while Flag=0 do
      if $\hat{Y}_t^{(1)}$ and $\hat{Y}_t^{(2)} \in \mathcal{X}/\partial\mathcal{X}$ then
        if $|\hat{Y}_t^{(1)} - \hat{Y}_t^{(2)}| > d$ then
          Compute $(\hat{Y}_{t+1}^{(1)}, \hat{Y}_{t+1}^{(2)})$ using reflection coupling or independent coupling
          $t \leftarrow t + 1$
        else
          Compute $(\hat{Y}_{t+1}^{(1)}, \hat{Y}_{t+1}^{(2)})$ using maximal coupling
          if coupled successfully then
            Flag=1
            $\tau_c^m = t$
          else
            $t \leftarrow t + 1$
          end if
        end if
      end if
    end while
  end for
  Use $\tau_c^1, \cdots, \tau_c^M$ to compute $\mathbb{P}(\tau_c > t | t < \min(\tau_{Y^{(1)}}, \tau_{Y^{(2)}}))$
  Fit the tail of $\log \mathbb{P}(\tau_c > t | t < \min(\tau_{Y^{(1)}}, \tau_{Y^{(2)}}))$ versus $t$ by linear regression. Compute the slope $\gamma$.
```

5.5 Numerical Examples

5.5.1 SIR model

Consider an epidemic model in which the whole population is divided into three distinct classes S(susceptible), I(infected) and R(recovered), respectively. After non-dimensionalization, the ODE version of an SIR model reads

$$\begin{aligned}\frac{dS}{dt} &= (\alpha - \beta SI - \mu S)\\ \frac{dI}{dt} &= (\beta SI - (\mu + \rho + \gamma)I)\\ \frac{dR}{dt} &= (\gamma I - \mu R)\end{aligned} \tag{5.5.1}$$

where α is the birth rate, μ is the disease-free death rate,ρ is the excess death rate for the infected class,γ is the recover rate for the infected population,and β is the effective contact rate between the susceptible class and infected class [13]. Note R completely depends on S and I. So we just consider the evolutions of S and I.

Now we let V be the total population and consider the corresponding stochastic mass action network. There are four reactions are involved in this network. The stochastic mass action network can be defined by a Poisson process X_n.

$$\begin{aligned}&\emptyset \overset{\alpha}{\Rightarrow} S,\ S + I \overset{\beta}{\Rightarrow} 2I\\ &S \overset{\mu}{\Rightarrow} \emptyset,\ I \overset{\mu+\rho+\gamma}{\Rightarrow} \emptyset\end{aligned} \tag{5.5.2}$$

Applying the numerical representation in (2.5.6), we have the approximate rate functions of Poisson process $\hat{X}_n$:

$$\begin{aligned}q_{1,n} &= \sum_{m=0}^{n-1} Vh\alpha,\ q_{2,n} = \sum_{m=0}^{n-1} Vh\beta S_m I_m,\\ q_{3,n} &= \sum_{m=0}^{n-1} Vh\mu S_m,\ q_{4,n} = \sum_{m=0}^{n-1} Vh(\mu + \rho + \gamma)I_m.\end{aligned}$$

Let $P_i, i = 1, 2, 3, 4$ be independent unit rate Poisson processes. Then $\hat{X}_n$ is driven by the discrete approximation of $\{P_i\}_{i=1}^4$. The rule of update of the numerical approximation $\hat{X}_n$ follows

$$\hat{X}_{n+1} = \begin{pmatrix} S_{n+1} \\ I_{n+1} \end{pmatrix} = \begin{pmatrix} S_n \\ I_n \end{pmatrix} + \frac{1}{V} \begin{pmatrix} \boldsymbol{f}_1(P_1, \cdots, P_4, q_{1,n}, \cdots, q_{4,n}) \\ \boldsymbol{f}_2(P_1, \cdots, P_4, q_{1,n}, \cdots, q_{4,n}) \end{pmatrix}, \tag{5.5.3}$$

where $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ comes from discrete approximation in equation (2.5.6). To improve the readability of the present thesis, we move detailed expressions of $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ to the appendix.

As described in Section 2.1, each Poisson processes $P_i, i = 1, 2, 3, 4$ is path-wisely approximated by a Wiener process $B_i, i = 1, 2, 3, 4$. Further, the discrete approximation $\hat{X}_n$ is pathwisely approximated by a Euler-Maruyama scheme $\hat{Y}_n$ reads

$$\hat{Y}_{n+1} = \begin{pmatrix} S_{n+1} \\ I_{n+1} \end{pmatrix} = \begin{pmatrix} S_n \\ I_n \end{pmatrix} + \frac{1}{V} \begin{pmatrix} \boldsymbol{g}_1(q_{1,n}, \cdots, q_{4,n}) \\ \boldsymbol{g}_2(q_{1,n}, \cdots, q_{4,n}) \end{pmatrix} + \frac{1}{V} \begin{pmatrix} \boldsymbol{\sigma}_1(B_1, \cdots, B_4, q_{1,n}, \cdots, q_{4,n}) \\ \boldsymbol{\sigma}_2(B_1, \cdots, B_4, q_{1,n}, \cdots, q_{4,n}) \end{pmatrix}, \tag{5.5.4}$$

where functions $\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{\sigma}_1$, and $\boldsymbol{\sigma}_2$ follows the expression in equation (2.5.7). We refer the appendix for the detailed form of these functions.

By the stationary increments property of standard Wiener process, we know that every finite difference of B_i is normally distributed. In addition Wiener processes $B_i, i = 1, 2, 3, 4$ are independent. Therefore, equation (5.5.4) can be simplified to:

$$\hat{Y}_{n+1} = \begin{pmatrix} S_{n+1} \\ I_{n+1} \end{pmatrix} = \begin{pmatrix} S_n \\ I_n \end{pmatrix} + \frac{1}{V} \begin{pmatrix} \boldsymbol{g}_1(q_{1,n}, \cdots, q_{4,n}) \\ \boldsymbol{g}_2(q_{1,n}, \cdots, q_{4,n}) \end{pmatrix} + \frac{1}{V} M \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{pmatrix} \tag{5.5.5}$$

where $W_i, i = 1, \cdots, 4$ are independent standard normal random variables, and M is a matrix that depends only on S_n and I_n. We refer readers to the appendix for the full expression of M.

In order to estimate the distance between two QSDs, we need to find the contraction rate α for diffusion process $\hat{Y}$ above. However, the diffusion matrix M in $\hat{Y}$ is not square, which makes a reflection coupling be difficult. Here we define an equivalent diffusion process that is driven by a 2D Wiener process but has the same law as $\hat{Y}$. In our simulation, we compute the 2 by 2 covariance matrix $N = MM^T$, and set the square root of N to be the new diffusion matrix. Then $\hat{Y}$ can be re-written as

$$\begin{aligned}\hat{Y}_{n+1} = &\begin{pmatrix} S_n \\ I_n \end{pmatrix} + \frac{1}{V}\begin{pmatrix} \boldsymbol{g}_1(q_{1,n}, \cdots, q_{4,n}) \\ \boldsymbol{g}_2(q_{1,n}, \cdots, q_{4,n}) \end{pmatrix} \\ &+ \frac{1}{\sqrt{\mathrm{tr}(N) + 2\sqrt{\det(N)}}}(N + \det(N)Id)\begin{pmatrix} W_1 \\ W_2 \end{pmatrix},\end{aligned} \tag{5.5.6}$$

where tr(N) is the trace of N and det(N) is the determinant of N, and Id is the identity matrix. It is easy to see that the diffusion process $\hat{Y}$ in equations (5.5.5) and (5.5.6) are equivalent. Hence we do not change its notation here. The modification of $\hat{Y}$ allows us to run Algorithm 7 to compute the coupling time distribution.

It remains to compute the finite time error. Let $\partial\mathcal{X}$ be the union of x-axis and y-axis. The model parameters are set as $\alpha = 7, \beta = 3, \mu = 1, \rho = 1, \gamma = 2$. Processes $\hat{X}$ and $\hat{Y}$ admit QSDs $\hat{\pi}_X$ and $\hat{\pi}_Y$, respectively. Long trajectories $P(i\Delta)$ and $B(i\Delta)$ for $i = \{1, \cdots, 2^{20}\}$ and $\Delta = 0.01$ are constructed when we consider the trajectory-by-trajectory behaviour of two processes. The time step size is $h = 0.001$ and the fixed time is set as $T = 0.5$.

The result for $V = 1000$ is demonstrated in Figure 5.1. Left bottom of Figure 5.1 shows the QSD of diffusion process $\hat{Y}$. The QSD of the Poisson process is shown on right top of Figure 5.1. The difference of these two QSDs is shown at the bottom of Figure 5.1. We can see that the total variation distance between two QSDs is 0.0901, which is considered to be small. This is reasonable because with high probability, the trajectories of both Poisson process and the diffusion process moves far away from the absorbing set $\partial\mathcal{X}$.

volume V	finite time error	contraction rate γ	$d_w(\hat{\pi}_X, \hat{\pi}_Y)$
1000	0.0026	1.2853	0.0054
400	0.0079	1.2418	0.0170
100	0.0279	1.1613	0.0634
10	0.1748	1.0912	0.3639

Table 5.1: SIR model. Numerical results for different volumes

The total variation distance between two QSDs is consistent with the prediction developed in this thesis. We first use Algorithm 7 to compute the distribution of the coupling time, which is shown in Figure 1 Top Left. Then we use Algorithm 6 to compute the finite time error. The finite time error is 0.0026 for $V = 1000$. As a result, the upper bound given in equation (5.4.1) is 0.0054 for $V = 1000$, which is smaller than the empirical total variation error 0.0901 in this case.

Then we carry out similar computations for $V = 10$. The result is shown in Figure 5.2. To compare with the case for $V = 1000$ on the same mesh, we re-scaled the probability density function obtained from the Monte-Carlo simulation. The probability density in one bin in the coarse mesh is evenly distributed into many bins in the refined mesh. The difference between two QSDs are shown at the bottom of Figure 5.2. It is not hard to see the total variation distance becomes significantly larger when the volume gets smaller. Same as above, we use Algorithm 7 to compute the distribution of the coupling time distribution (Figure 5.2 Top Left) and use Algorithm 6 to compute the finite time error. The finite time error is 0.1748 for $V = 10$. As a result, the upper bound given in (5.4.1) is 0.3639 for $V = 10$. This is consistent with the numerical finding shown in Figure 5.2 Bottom Right.

As we consider the effect of the capacity volume, the finite time error and the contraction rate for different volumes are compared in Table 5.1. The last column $d_w(\hat{\pi}_X, \hat{\pi}_Y)$ is computed using (5.4.1). Being consistent with Theorem 2.5.2, the 1-Wasserstein distance between two QSDs is smaller as V getting larger.

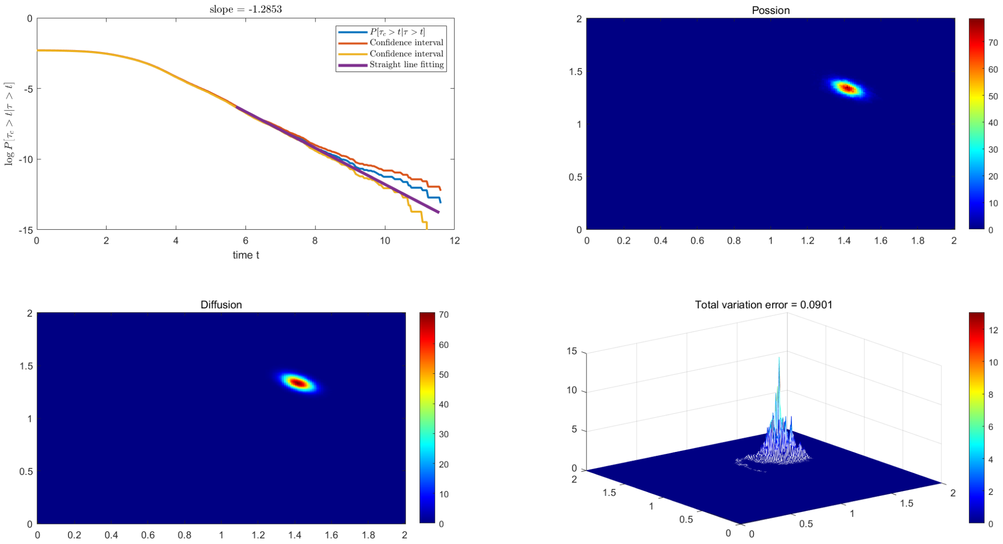

Figure 5.1: (Case $V = 1000$) **Upper panel**: (**Left**) $\mathbb{P}(\tau_c > t|\tau < t)$ vs.t. (**Right**) QSD of Poisson process. **Lower panel**: (**Left**) QSD of diffusion process. (**Right**) Total variation of two QSDs.

5.5.2 Oregonator system

In this example, we consider a well known example of chemical oscillator called the Belousov-Zhabotinsky (BZ) reaction model or ”Oregonator” [7,16,20]. The ODE version of an Oregnator system is given by

$$\begin{aligned}\frac{dS_1}{dt} &= S_1S_2 - C_2S_1S_2 + C_3S_1 - 2C_4S_1^2\\ \frac{dS_2}{dt} &= -C_1S_2 - C_2S_1S_2 + C_5hS_3\\ \frac{dS_3}{dt} &= 2C_3S_1 - C_5S_3.\end{aligned}$$

We refer Figure 5.3 Top Left for a sample trajectories of the Oregonator on $\mathbb{R}^3_+$. The parameter values are chosen as $C_1 = 2560, C_2 = 800000, C_3 = 16000, C_4 = 2000, C_5 = 9000, \delta = 0.4$.

Let V be the volume. Six reactions in this process are shown as following.

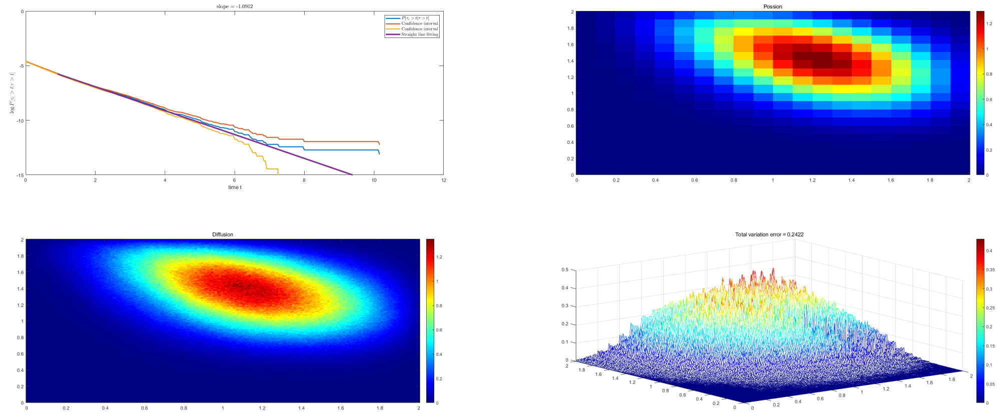

Figure 5.2: (Case $V = 10$) **Upper panel**: (**Left**) $\mathbb{P}(\tau_c > t|\tau < t)$ vs.t. (**Right**) QSD of Poisson process. **Lower panel**: (**Left**) QSD of diffusion process. (**Right**) Total variation of two QSDs.

$$S_2 \overset{C_1}{\Rightarrow} S_1,\ S_1 + S_2 \overset{C_2}{\Rightarrow} \emptyset,\ S_1 \overset{C_3}{\Rightarrow} 2S_1 + 2S_3$$
$$2S_1 \overset{C_4}{\Rightarrow} \emptyset,\ S_3 \overset{C_5\delta}{\Rightarrow} S_2,\ S_3 \overset{C_5(1-\delta)}{\Rightarrow} \emptyset$$

Applying the numerical representation in (2.5.6), we have the approximate rate functions of Poisson process $\hat{X}_n = (S_{1,n}, S_{2,n}, S_{3,n})$:

$$q_{1,n} = \sum_{m=0}^{n-1} VhC_1S_{2,m},\ q_{2,n} = \sum_{m=0}^{n-1} VhC_2S_{1,m}S_{2,m},\ q_{3,n} = \sum_{m=0}^{n-1} VhC_3S_{1,m},$$
$$q_{4,n} = \sum_{m=0}^{n-1} VhC_4S_{1,m}^2,\ q_{5,n} = \sum_{m=0}^{n-1} VhC_5\delta S_{3,m},\ q_{6,n} = \sum_{m=0}^{n-1} VhC_5(1-\delta)S_{3,m}.$$

We remark terms $S_{1,m}$ is the numerical value of species S_1 at time step m, and cases of other terms are analogous. Hence the Poisson process $\hat{X}$ of the Oregonator model can be written as

$$\hat{X}_{n+1} = \begin{pmatrix} S_{1,n+1} \\ S_{2,n+1} \\ S_{3,n+1} \end{pmatrix} = \begin{pmatrix} S_{1,n} \\ S_{2,n} \\ S_{3,n} \end{pmatrix} + \frac{1}{V} \begin{pmatrix} \boldsymbol{f_1}(P_1, \cdots, P_6, q_{1,n}, \cdots, q_{6,n}) \\ \boldsymbol{f_2}(P_1, \cdots, P_6, q_{1,n}, \cdots, q_{6,n}) \\ \boldsymbol{f_3}(P_1, \cdots, P_6, q_{1,n}, \cdots, q_{6,n}) \end{pmatrix},$$

where $P_i, i = \{1, \cdots, 6\}$ are independent unite rate Poisson processes. $\boldsymbol{f_1}$, $\boldsymbol{f_2}$ and $\boldsymbol{f_3}$ comes from discrete approximation in equation (2.5.6). To improve the readability of the present thesis, we move detailed expressions of $\boldsymbol{f_1}$, $\boldsymbol{f_2}$ and $\boldsymbol{f_3}$ to the appendix.

The diffusion approximation $\hat{Y}$ can be written as

$$\hat{Y}_{n+1} = \begin{pmatrix} S_{1,n+1} \\ S_{2,n+1} \\ S_{3,n+1} \end{pmatrix} = \begin{pmatrix} S_{1,n} \\ S_{2,n} \\ S_{3,n} \end{pmatrix} + \frac{1}{V} \begin{pmatrix} \boldsymbol{g_1}(q_{1,n}, \cdots, q_{6,n}) \\ \boldsymbol{g_2}(q_{1,n}, \cdots, q_{6,n}) \\ \boldsymbol{g_3}(q_{1,n}, \cdots, q_{6,n}) \end{pmatrix} + \frac{1}{V} \begin{pmatrix} \boldsymbol{\sigma_1}(B_1, \cdots, B_6, q_{1,n}, \cdots, q_{6,n}) \\ \boldsymbol{\sigma_2}(B_1, \cdots, B_6, q_{1,n}, \cdots, q_{6,n}) \\ \boldsymbol{\sigma_3}(B_1, \cdots, B_6, q_{1,n}, \cdots, q_{6,n}) \end{pmatrix} \tag{5.5.7}$$

where $B_i, i = \{1, \cdots, 6\}$ are independent standard Wiener processes, functions $\boldsymbol{g_1}$, $\boldsymbol{g_2}$, $\boldsymbol{g_3}$, $\boldsymbol{\sigma_1}$, $\boldsymbol{\sigma_2}$ and $\boldsymbol{\sigma_3}$ follows the expression in equation (2.5.7). We refer the appendix for the detailed form of these functions.

By the stationary increments property and independence of Wiener processes $B_i, i = \{1, \cdots, 6\}$, equation (5.5.7) can be simplified to:

$$\hat{Y}_{n+1} = \begin{pmatrix} S_{1,n+1} \\ S_{2,n+1} \\ S_{3,n+1} \end{pmatrix} = \begin{pmatrix} S_{1,n} \\ S_{2,n} \\ S_{3,n} \end{pmatrix} + \frac{1}{V} \begin{pmatrix} \boldsymbol{g_1}(q_{1,n}, \cdots, q_{6,n}) \\ \boldsymbol{g_2}(q_{1,n}, \cdots, q_{6,n}) \\ \boldsymbol{g_3}(q_{1,n}, \cdots, q_{6,n}) \end{pmatrix} + \frac{1}{V} M \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \end{pmatrix} \tag{5.5.8}$$

where $W_i, i = 1, \cdots, 6$ are independent standard normal random variables, and M is a matrix that depends only on S_n and I_n. We refer readers to the appendix for the full expression of M.

Let $\partial\mathcal{X}$ be union of x-axis, y-axis and z-axis. Processes $\hat{X}$ and $\hat{Y}$ admit QSDs $\hat{\pi}_X$ and $\hat{\pi}_Y$, respectively. Long trajectories $P(i\Delta)$ and $B(i\Delta)$ for $i = \{1, \cdots, 2^{29}\}$ and $\Delta = 0.001$ are constructed when we consider the trajectory-by-trajectory behaviour of two processes. The time step size is $h = 10^{-8}$ and the fixed time is set as $T = 0.0002$. Large rate coefficients C_i

make the numerical results easily to beyond the length of long trajectory $B(i\Delta)$, so we pick small time step size h and the fixed finite time T.

Figure 5.3 Top Left shows the solution of the ordinary differential equation. For any initial point, the trajectory eventually converges to the limit cycle. In terms of thermodynamics, the oscillation is induced through dissipation of energy and is often called a self-sustained oscillator [35]. The trajectories of Poisson process and the diffusion process up to fixed time $T = 0.0002$ are shown on the Top Right and Bottom Left. It looks that the trajectories are close and this is reasonable because with high probability, the trajectories of both Poisson process and the diffusion process moves far away from the absorbing set $\partial\mathcal{X}$. There are only a few regeneration events (the lines crossing the limit cycle). We compute the distribution of the coupling time. The coupling time distribution and its exponential tail are shown in Figure 5.3 Top Left. Then we use Algorithm 6 to compute the finite time error. The finite time error is 0.0057 for $V = 1000$. As a result, the upper bound given in (5.4.1) is 0.0116 for $V = 1000$. For $V = 10$, the finite time error is 0.4531 and the upper bound given in (5.4.1) is 0.4531.

To compare the different situations for volume $V = 1000$ and $V = 10$, we plot the trajectories for both processes for each species. Trajectories for $V = 1000$ is shown in the upper row of Figure 5.4 and lower row shows the case for $V = 10$. It is not hard to see the Poisson process is quite close to the diffusion process when $V = 1000$. But when the volume is too small, not much Poisson jumps can be observes in the Poisson process, while significant noise can be seen in the diffusion approximation. As a result, the finite time error for $V = 10$ is 0.0563, which is around ten times larger than that for $V = 1000$. Same as above, we compute the contraction rate γ of the coupling time distribution to be 2.0927×10^5. This is due to the large magnitude of noise in the diffusion approximation. As a result, the upper bound given in (5.4.1) is 0.4531 for $V = 10$. We conclude that the diffusion approximation does not approximate the QSD well when the volume is not large enough.

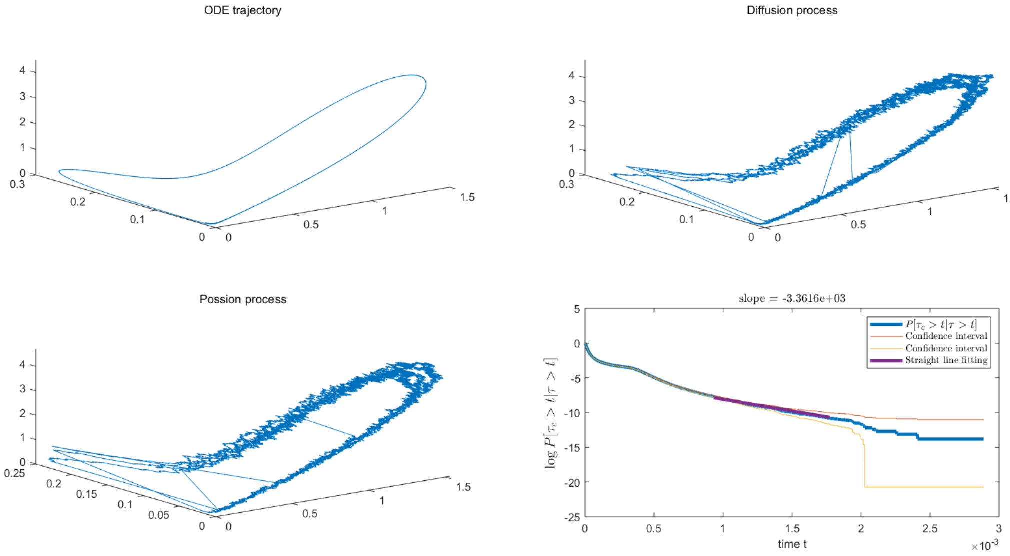

Figure 5.3: (Case $V = 1000$) **Upper panel**: (**Left**) ODE trajectories. (**Right**) Trajectories of Poisson process. **Lower panel**: (**Left**) Trajectories of diffusion process. (**Right**) $\mathbb{P}(\tau_c > t|\tau < t)$ vs.t.

volume V	finite time error	contraction rate γ	$d_w(\hat{\pi}_X, \hat{\pi}_Y)$
1000	0.0057	3.3616*10^3	0.0116
400	0.0205	2.0599*10^4	0.0208
100	0.1322	6.0150*10^4	0.1322
10	0.4531	2.0927*10^5	0.4531

Table 5.2: Oregonator model: Numerical results for different volumes

As we consider the effect of the capacity volume, the finite time error and the contraction rate for different volumes are compared in Table 5.2. The last column $d_w(\hat{\pi}_X, \hat{\pi}_Y)$ is computed via (5.4.1). It is not hard to see that upper bound of $d_w(\hat{\pi}_X, \hat{\pi}_Y)$ is quite larger when $V = 10$. This is consistent with Theorem 2.5.2, the supreme distance between two processes will be smaller as V is getting larger.

5.5.3 4D Lotka-Volterra Competitive Dynamics

Originally derived by Volterra in 1926 to describe the interaction between a predator species and a prey species [33] and independently by Lotka to describe a chemical reaction

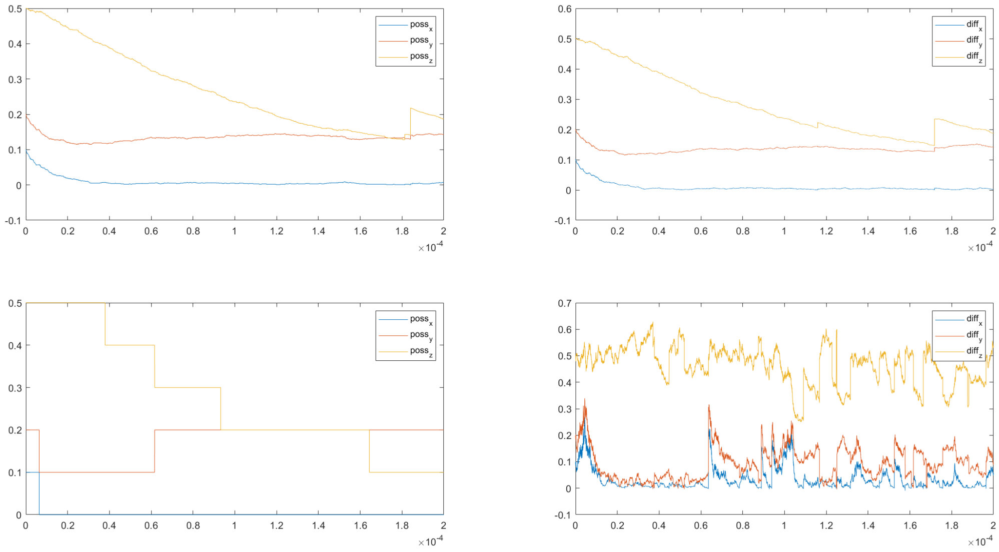

Figure 5.4: ($V = 1000$ vs. $V = 10$) **Upper panel**: (**Left**) Trajectories of Poisson process for $V = 1000$. (**Right**)Trajectories of diffusion process for $V = 1000$. **Lower panel**: (**Left**) Trajectories of Poisson process for $V = 10$. (**Right**) Trajectories of diffusion process for $V = 10$.

[44], the general Lotka-Volterra model is widely used in ecology, biology, chemistry, physics, etc [37]. In this example we consider here a chaotic system in which 4 species with whole population V compete for a finite set of resources. The ODE version of the system reads.

$$\frac{dS_i}{dt} = r_i S_i (1 - \sum_{j=1}^{4} a_{ij} S_j), \;\; i = 1, 2, 3, 4.$$

Here r_i represents the growth rate of species i and a_{ij} represents the extent to which species j competes for resources used by species i. The parameter values are

$$r = (r_i)_{i=1}^4 = \begin{pmatrix} 1 \\ 0.72 \\ 1.53 \\ 1.27 \end{pmatrix}, \;\; A = (a_{ij})_{i,j=1}^4 = \begin{pmatrix} 1 & 1.09 & 1.52 & 0 \\ 0 & 1 & 0.44 & 1.36 \\ 2.33 & 0 & 1 & 0.47 \\ 1.21 & 0.51 & 0.35 & 1 \end{pmatrix}$$

For $i = 1, \cdots, 4$, all reactions in this system are shown as follows.

$$S_i \overset{r_i}{\Rightarrow} 2S_i,\ S_1 + S_i \overset{a_{i1}r_i}{\Rightarrow} S_1,\ S_2 + S_i \overset{a_{i2}r_i}{\Rightarrow} S_2,\ S_3 + S_i \overset{a_{i3}r_i}{\Rightarrow} S_3,\ S_4 + S_i \overset{a_{i4}r_i}{\Rightarrow} S_4$$

The corresponding rate functions are

$$\begin{aligned}
q_{i,1}^n &= \sum_{m=0}^{n-1} Vhr_iS_{i,m} \\
q_{i,2}^n &= \sum_{m=0}^{n-1} Vhr_ia_{i1}S_{1,m}S_{i,m} \\
q_{i,3}^n &= \sum_{m=0}^{n-1} Vhr_ia_{i2}S_{2,m}S_{i,m} \\
q_{i,4}^n &= \sum_{m=0}^{n-1} Vhr_ia_{i3}S_{3,m}S_{i,m} \\
q_{i,5}^n &= \sum_{m=0}^{n-1} Vhr_ia_{i4}S_{4,m}S_{i,m}
\end{aligned}$$

As three zeros appear in coefficient matrix A, this system actually include 17 reactions. Therefore, the Poisson process $\hat{X} = (S_{1,n}, S_{2,n}, S_{3,n}, S_{4,n})$ can be written as

$$\hat{X}_{n+1} = \begin{pmatrix} S_{1,n+1} \\ S_{2,n+1} \\ S_{3,n+1} \\ S_{4,n+1} \end{pmatrix} = \begin{pmatrix} S_{1,n} \\ S_{2,n} \\ S_{3,n} \\ S_{4,n} \end{pmatrix} + \frac{1}{V} \begin{pmatrix} \boldsymbol{f_1}(P_1, \cdots, P_{17}, q_{i,1}^n, \cdots, q_{i,5}^n) \\ \boldsymbol{f_2}(P_1, \cdots, P_{17}, q_{i,1}^n, \cdots, q_{i,5}^n) \\ \boldsymbol{f_3}(P_1, \cdots, P_{17}, q_{i,1}^n, \cdots, q_{i,5}^n) \\ \boldsymbol{f_4}(P_1, \cdots, P_{17}, q_{i,1}^n, \cdots, q_{i,5}^n) \end{pmatrix},$$

where $i = 1, \cdots, 4$, $P_j,\ j = \{1, \cdots, 17\}$ are independent unit rate Poisson processes, $\boldsymbol{f_1}$, $\boldsymbol{f_2}$, $\boldsymbol{f_3}$ and $\boldsymbol{f_4}$ comes from discrete approximation in equation (2.5.6). To improve the readability of the present thesis, we move detailed expressions of $\boldsymbol{f_1}$ to $\boldsymbol{f_4}$ to the appendix.

The diffusion approximation $\hat{Y}$ can be written as

$$\hat{Y}_{n+1}=\begin{pmatrix}S_{1,n+1}\\S_{2,n+1}\\S_{3,n+1}\\S_{4,n+1}\end{pmatrix}=\begin{pmatrix}S_{1,n}\\S_{2,n}\\S_{3,n}\\S_{4,n}\end{pmatrix}+\frac{1}{V}\begin{pmatrix}\boldsymbol{g_1}(q_{i,1}^n,\cdots,q_{i,5}^n)\\\boldsymbol{g_2}(q_{i,1}^n,\cdots,q_{i,5}^n)\\\boldsymbol{g_3}(q_{i,1}^n,\cdots,q_{i,5}^n)\\\boldsymbol{g_4}(q_{i,1}^n,\cdots,q_{i,5}^n)\end{pmatrix}+\frac{1}{V}\begin{pmatrix}\boldsymbol{\sigma_1}(B_1,\cdots,B_{17},q_{i,1}^n,\cdots,q_{i,5}^n)\\\boldsymbol{\sigma_2}(B_1,\cdots,B_{17},q_{i,1}^n,\cdots,q_{i,5}^n)\\\boldsymbol{\sigma_3}(B_1,\cdots,B_{17},q_{i,1}^n,\cdots,q_{i,5}^n)\\\boldsymbol{\sigma_4}(B_1,\cdots,B_{17},q_{i,1}^n,\cdots,q_{i,5}^n)\end{pmatrix}, \tag{5.5.9}$$

where $i=1,\cdots,4$, $B_j,\ j=\{1,\cdots,17\}$ are independent standard Wiener process, functions $\boldsymbol{g_1}$, $\boldsymbol{g_2}$, $\boldsymbol{g_3}$, $\boldsymbol{g_4}$, $\boldsymbol{\sigma_1}$, $\boldsymbol{\sigma_2}$, $\boldsymbol{\sigma_3}$ and $\boldsymbol{\sigma_4}$ follows the expression in equation (2.5.7). We refer the appendix for the detailed form of these functions.

By the stationary increments property and independence of Wiener processes $B_i, i=\{1,\cdots,6\}$, equation (5.5.9) can be simplified to:

$$\hat{Y}_{n+1}=\begin{pmatrix}S_{1,n+1}\\S_{2,n+1}\\S_{3,n+1}\\S_{4,n+1}\end{pmatrix}=\begin{pmatrix}S_{1,n}\\S_{2,n}\\S_{3,n}\\S_{4,n}\end{pmatrix}+\frac{1}{V}\begin{pmatrix}\boldsymbol{g}_1(q_1,\cdots,q_4)\\\boldsymbol{g}_2(q_1,\cdots,q_4)\end{pmatrix}+\frac{1}{V}M\begin{pmatrix}W_1\\W_2\\\vdots\\W_{16}\\W_{17}\end{pmatrix} \tag{5.5.10}$$

where $W_i, i=1,\cdots,17$ are independent standard normal random variables, and M is a matrix that depends only on S_n and I_n. We refer readers to the appendix for the full expression of M.

Let $\partial\mathcal{X}$ be union of 4 axes. Processes $\hat{X}$ and $\hat{Y}$ admit QSDs $\hat{\pi}_X$ and $\hat{\pi}_Y$, respectively. Long trajectories $P(i\Delta)$ and $B(i\Delta)$ for $i=\{1,\cdots,2^{22}\}$ and $\Delta=0.01$ are constructed when we consider the trajectory-by-trajectory behaviour of two processes. The time step size is $h=0.001$ and the fixed time is set as $T=1$.

Figure 5.5 Top Left shows the solution of the ordinary differential equation projected onto $x_1x_2x_3$ space. The trajectories of Poisson process and the diffusion process are shown on the Top Right and Bottom Left. It looks that the trajectories are close and this is

reasonable because with high probability, the trajectories of both Poisson process and the diffusion process moves far away from the absorbing set $\partial\mathcal{X}$. We compute the distribution of the coupling time. The coupling time distribution and its exponential tail are shown in Figure 5.5 Top Left, that gives the contraction rate $\gamma = 0.0849$. Then we apply Algorithm 6 to compute the finite time error. The finite time error is 0.0030 for $V = 1000$. As a result, the upper bound given in (5.4.1) is 0.0375 for $V = 1000$.

To compare the different situations for volume $V = 1000$ and $V = 10$, we plot trajectories of each species for $V = 1000$ in Figure 5.6, and the case for $V = 10$ is shown in Figure 5.7. It is not hard to see the trajectory-by-trajectory behavior between Poisson process and diffusion process is quite remarkable when $V = 1000$. However, more regeneration happens in Poisson process when $V = 10$. So it's not surprised us that the finite time error for $V = 10$ is 0.1286, that around 40 times larger than the case for $V = 1000$. Trajectories of the Poisson process have high probability moving along the boundary in this case. Same as above, we compute the contraction rate γ of the coupling time distribution to be 1.7905. As a result, the upper bound given in (5.4.1) is 0.1543 for $V = 10$.

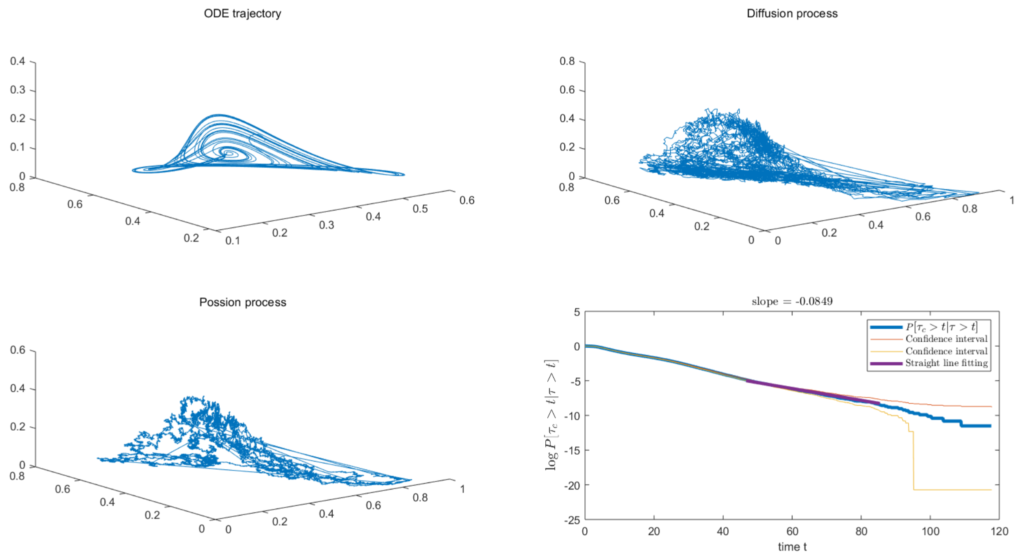

Figure 5.5: (Case $V = 1000$) **Upper panel**: (**Left**) ODE trajectories. (**Right**) Poisson process. **Lower panel**: (**Left**) Diffusion process. (**Right**) $\mathbb{P}(\tau_c > t|\tau < t)$ vs.t.

volume V	finite time error	contraction rate γ	$d_w(\hat{\pi}_X, \hat{\pi}_Y)$
1000	0.0030	0.0849	0.0375
400	0.0110	0.1831	0.0659
100	0.0502	0.3110	0.1878
10	0.1286	1.7905	0.1543

Table 5.3: 4D Lotka-Volterra model: Numerical results for different volumes

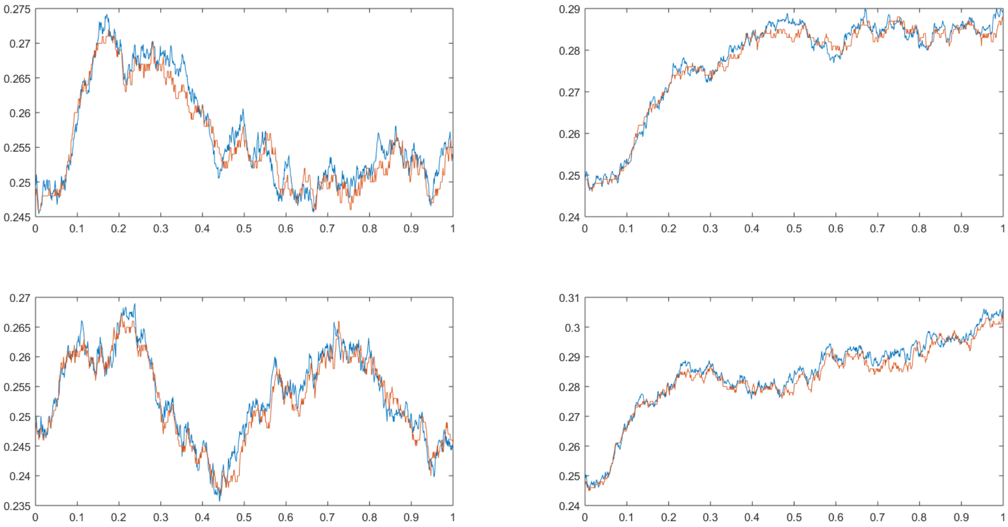

Figure 5.6: (Case $V = 1000$) Poisson trajectories and diffusion trajectories for 4 species .

As we consider the effect of the capacity volume, the finite time error and the contraction rate for different volumes are compared in Table 5.3. The last column $d_w(\hat{\pi}_X, \hat{\pi}_Y)$ is computed via (5.4.1). It is not hard to see that upper bound of $d_w(\hat{\pi}_X, \hat{\pi}_Y)$ is quite larger when $V = 10$. This is consistent with Theorem 2.5.2, the supreme distance between two processes will be smaller as V is getting larger.

5.6 Conclusion

we develop a coupling-based approach to quantitatively estimate the distance between the QSD of a stochastic mass-action process and that of its diffusion approximation. The dependence of QSDs in terms of the volume of the mass-action system is studied. To address

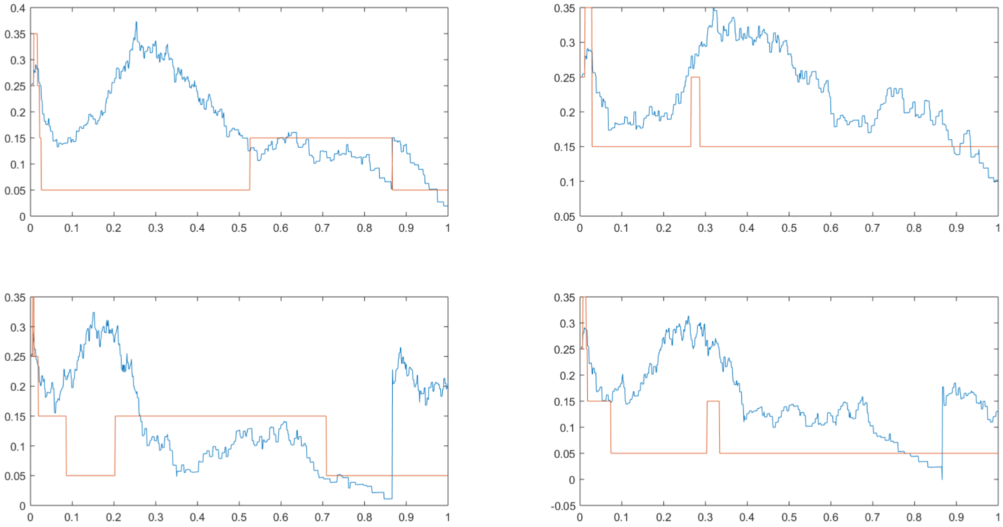

Figure 5.7: (Case $V = 10$) Poisson trajectories and diffusion trajectories for 4 species.

the challenge of QSDs, we use the idea of regeneration from QSDs after exiting to construct a process with stationary distribution. This is the the main change from our previous work [15,31]. Both the coupling algorithm and the path-wise matching of a stochastic mass-action system and its diffusion approximation need to be adapted to the regeneration from QSDs. We compare the finite time error and the rate of contraction for different population size V. All numerical results shows that the distance between two QSDs is smaller for larger population. In general, the effect of demographic noise must be seriously addressed when the population is small.

The study of path-wise approximation of stochastic mass-action systems by diffusion processes and the coupling of diffusion processes motivates a very interesting question. All our existing work relies on the reflection coupling of diffusion processes, which is known to be highly effective. Then how can one effectively couple two continuous-time Markov processes on a lattice? A successful coupling of two trajectories of a mass-action system will extend our framework of sensitivity analysis to many more applications. We believe it is very difficult to couple the exact stochastic mass-action system because random events occur at continuous time. However, there may be some way of building a ”discrete reflection” and coupling two tau-leaping trajectories, i.e., two trajectories of equation (2.5.6) effectively. This will be addressed in our future work.

APPENDIX

EXPRESSIONS OF MASS-ACTION SYSTEMS AND THEIR DIFFUSION APPROXIMATIONS

To improve the readability, we put the explicit formulas of the Poisson approximation and the diffusion approximation for each model in this section.

A.1 SIR model

There are four reactions are involved in the SIR system, so we have 4 pairs of Poisson process P_i and Wiener process B_i appear in the evolution of each class. The rule of update of the numerical approximation $\hat{X}_n$ follows

$$\begin{aligned}
\hat{X}_{n+1} &= \binom{S_{n+1}}{I_{n+1}} \\
&= \binom{S_n}{I_n} + \frac{1}{V}\begin{pmatrix} [P_1(q_{1,n+1}) - P_1(q_{1,n})] - [P_2(q_{2,n+1}) - P_2(q_{2,n}] - [P_3(q_{3,n+1} - P_3(q_{3,n})] \\ [P_2(q_{2,n+1}) - P_2(q_{2,n})] - [P_4(q_{4,n+1}) - P_4(q_{4,n})] \end{pmatrix}, \\
&:= \binom{S_n}{I_n} + \frac{1}{V}\begin{pmatrix} \boldsymbol{f}_1(P_1, \cdots, P_4, q_{1,n}, \cdots, q_{4,n}) \\ \boldsymbol{f}_2(P_1, \cdots, P_4, q_{1,n}, \cdots, q_{4,n}) \end{pmatrix}
\end{aligned}$$

where $P_i, i = \{1, 2, 3, 4\}$ are independent unit rate Poisson processes.

The rule of update of the numerical approximation $\hat{Y}_n$ follows

$$\begin{aligned}
\hat{Y}_{n+1} = \binom{S_{n+1}}{I_{n+1}} &= \binom{S_n}{I_n} + \frac{1}{V}\begin{pmatrix} [q_{1,n+1} - q_{1,n}] - [q_{2,n+1} - q_{2,n}] - [q_{3,n+1} - q_{3,n}] \\ [q_{2,n+1} - q_{2,n}] - [q_{4,n+1} - q_{4,n}] \end{pmatrix} \\
&+ \frac{1}{V}\begin{pmatrix} [B_1(q_{1,n+1}) - B_1(q_{1,n})] - [B_2(q_{2,n+1}) - B_2(q_{2,n})] - [B_3(q_{3,n+1} - B_3(q_{3,n}] \\ [B_2(q_{2,n+1}) - B_2(q_{2,n})] - [B_4(q_{4,n+1}) - B_4(q_{4,n})] \end{pmatrix}, \\
&:= \binom{S_n}{I_n} + \frac{1}{V}\begin{pmatrix} \boldsymbol{g}_1(q_{1,n}, \cdots, q_{4,n}) \\ \boldsymbol{g}_2(q_{1,n}, \cdots, q_{4,n}) \end{pmatrix} + \frac{1}{V}\begin{pmatrix} \boldsymbol{\sigma}_1(B_1, \cdots, B_4, q_{1,n}, \cdots, q_{4,n}) \\ \boldsymbol{\sigma}_2(B_1, \cdots, B_4, q_{1,n}, \cdots, q_{4,n}) \end{pmatrix},
\end{aligned}$$

where $B_i,\ i = \{1, 2, 3, 4\}$ are independent standard Wiener processs.

As two classes S_n and I_n and four reactions are considered in this SIR model, the corresponding diffusion matrix M should be a 2×4 matrix. Specifically, the diffusion matrix M reads as

$$
\begin{aligned}
\hat{Y}_{n+1} = \begin{pmatrix} S_{n+1} \\ I_{n+1} \end{pmatrix} &= \begin{pmatrix} S_n \\ I_n \end{pmatrix} + \frac{1}{V} \begin{pmatrix} [q_{1,n+1} - q_{1,n}] - [q_{2,n+1} - q_{2,n}] - [q_{3,n+1} - q_{3,n}] \\ [q_{2,n+1} - q_{2,n}] - [q_{4,n+1} - q_{4,n}] \end{pmatrix} \\
&+ \frac{1}{V} \begin{pmatrix} \sqrt{q_{1,n+1} - q_{1,n}} & -\sqrt{q_{2,n+1} - q_{2,n}} & -\sqrt{q_{3,n+1} - q_{3,n}} & 0 \\ 0 & \sqrt{q_{2,n+1} - q_{2,n}} & 0 & -\sqrt{q_{4,n+1} - q_{4,n}} \end{pmatrix} \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{pmatrix} \\
&= \begin{pmatrix} S_n \\ I_n \end{pmatrix} + \frac{1}{V} \begin{pmatrix} Vh\alpha - Vh\beta S_n I_n - Vh\mu S_n \\ Vh\beta S_n I_n - Vh(\mu + \rho + \gamma) I_n \end{pmatrix} \\
&+ \frac{1}{V} \begin{pmatrix} \sqrt{Vh\alpha} & -\sqrt{Vh\beta S_n I_n} & -\sqrt{Vh\mu S_n} & 0 \\ 0 & \sqrt{Vh\beta S_n I_n} & 0 & -\sqrt{Vh(\mu + \rho + \gamma) I_n} \end{pmatrix} \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{pmatrix} \\
&:= \begin{pmatrix} S_n \\ I_n \end{pmatrix} + \frac{1}{V} \begin{pmatrix} \boldsymbol{g}_1(q_{1,n}, \cdots, q_{4,n}) \\ \boldsymbol{g}_2(q_{1,n}, \cdots, q_{4,n}) \end{pmatrix} + \frac{1}{V} M \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{pmatrix}
\end{aligned}
$$

where $W_i,\ i = \{1, 2, 3, 4\}$ are independent standard normal distributed random variables.

A.2 Oregnator model

For the Oregnator model, there are six reactions involved. So we have 6 pairs of Poisson process P_i and B_i in the approximations. The rule of update of the numerical approximation $\hat{X}_n$ follows

$$
\begin{aligned}
\hat{X}_{n+1} &= \begin{pmatrix} S_{1,n+1} \\ S_{2,n+1} \\ S_{3,n+1} \end{pmatrix} = \begin{pmatrix} S_{1,n} \\ S_{2,n} \\ S_{3,n} \end{pmatrix} \\
&+ \frac{1}{V} \begin{pmatrix} [P_1(q_{1,n+1}) - P_1(q_{1,n})] - [P_2(q_{2,n+1}) - P_2(q_{2,n})] + [P_3(q_{3,n+1}) - P_3(q_{3,n})] - 2[P_4(q_{4,n+1}) - P_4(q_{4,n})] \\ -[P_1(q_{1,n+1}) - P_1(q_{1,n})] - [P_2(q_{2,n+1}) - P_2(q_{2,n})] + [P_5(q_{5,n+1}) - P_5(q_{5,n})] \\ 2[P_3(q_{3,n+1}) - P_3(q_{3,n})] - [P_5(q_{5,n+1}) - P_5(q_{5,n})] - [P_6(q_{6,n+1}) - P_6(q_{6,n})] \end{pmatrix} \\
&:= \begin{pmatrix} S_{1,n} \\ S_{2,n} \\ S_{3,n} \end{pmatrix} + \frac{1}{V} \begin{pmatrix} \boldsymbol{f_1}(P_1, \cdots, P_6, q_{1,n}, \cdots, q_{6,n}) \\ \boldsymbol{f_2}(P_1, \cdots, P_6, q_{1,n}, \cdots, q_{6,n}) \\ \boldsymbol{f_3}(P_1, \cdots, P_6, q_{1,n}, \cdots, q_{6,n}) \end{pmatrix}
\end{aligned}
$$

where $P_i, i = \{1, \cdots, 6\}$ are independent unite rate Poisson processes.

The diffusion approximation $\hat{Y}$ can be written as

$$
\begin{aligned}
\hat{Y}_{n+1} &= \begin{pmatrix} S_{1,n+1} \\ S_{2,n+1} \\ S_{3,n+1} \end{pmatrix} \\
&= \begin{pmatrix} S_{1,n} \\ S_{2,n} \\ S_{3,n} \end{pmatrix} + \frac{1}{V} \begin{pmatrix} [q_{1,n+1} - q_{1,n}] - [q_{2,n+1} - q_{2,n}] + [q_{3,n+1} - q_{3,n}] - 2[q_{4,n+1} - q_{4,n}] \\ -[q_{1,n+1} - q_{1,n}] - [q_{2,n+1} - q_{2,n}] + [q_{5,n+1} - q_{5,n}] \\ 2[q_{3,n+1} - q_{3,n}] - [q_{5,n+1} - q_{5,n}] - [q_{6,n+1} - q_{6,n}] \end{pmatrix} \\
&+ \frac{1}{V} \begin{pmatrix} [B_1(q_{1,n+1}) - B_1(q_{1,n})] - [B_2(q_{2,n+1}) - B_2(q_{2,n})] + [B_3(q_{3,n+1}) - B_3(q_{3,n})] - 2[B_4(q_{4,n+1}) - B_4(q_{4,n})] \\ -[B_1(q_{1,n+1}) - B_1(q_{1,n})] - [B_2(q_{2,n+1}) - B_2(q_{2,n})] + [B_5(q_{5,n+1}) - B_5(q_{5,n})] \\ 2[B_3(q_{3,n+1}) - B_3(q_{3,n})] - [B_5(q_{5,n+1}) - B_5(q_{5,n})] - [B_6(q_{6,n+1}) - B_6(q_{6,n})] \end{pmatrix} \\
&:= \begin{pmatrix} S_{1,n} \\ S_{2,n} \\ S_{3,n} \end{pmatrix} + \frac{1}{V} \begin{pmatrix} \boldsymbol{g_1}(q_{1,n}, \cdots, q_{6,n}) \\ \boldsymbol{g_2}(q_{1,n}, \cdots, q_{6,n}) \\ \boldsymbol{g_3}(q_{1,n}, \cdots, q_{6,n}) \end{pmatrix} + \frac{1}{V} \begin{pmatrix} \boldsymbol{\sigma_1}(B_1, \cdots, B_6, q_{1,n}, \cdots, q_{6,n}) \\ \boldsymbol{\sigma_2}(B_1, \cdots, B_6, q_{1,n}, \cdots, q_{6,n}) \\ \boldsymbol{\sigma_3}(B_1, \cdots, B_6, q_{1,n}, \cdots, q_{6,n}) \end{pmatrix},
\end{aligned}
$$

where $B_i, \{i = 1, \cdots, 6\}$ are independent Wiener processes.

As we focus on three classes $S_{1,n}, S_{2,n}, S_{3,n}$ and six reactions, we can confirm that the diffusion matrix M is a 3×6 matrix. Specifically, the diffusion matrix M is defined as follows.

$$
\begin{aligned}
\hat{Y}_{n+1} &= \begin{pmatrix} S_{1,n+1} \\ S_{2,n+1} \\ S_{3,n+1} \end{pmatrix} \\
&= \begin{pmatrix} S_{1,n} \\ S_{2,n} \\ S_{3,n} \end{pmatrix} + \frac{1}{V} \begin{pmatrix} [q_{1,n+1} - q_{1,n}] - [q_{2,n+1} - q_{2,n}] + [q_{3,n+1} - q_{3,n}] - 2[q_{4,n+1} - q_{4,n}] \\ -[q_{1,n+1} - q_{1,n}] - [q_{2,n+1} - q_{2,n}] + [q_{5,n+1} - q_{5,n}] \\ 2[q_{3,n+1} - q_{3,n}] - [q_{5,n+1} - q_{5,n}] - [q_{6,n+1} - q_{6,n}] \end{pmatrix} \\
&+ \frac{1}{V} \begin{pmatrix} \sqrt{q_{1,n+1} - q_{1,n}} & -\sqrt{q_{2,n+1} - q_{2,n}} & \sqrt{q_{3,n+1} - q_{3,n}} & -\sqrt{2[q_{4,n+1} - q_{4,n}]} & 0 & 0 \\ -\sqrt{q_{1,n+1} - q_{1,n}} & -\sqrt{q_{2,n+1} - q_{2,n}} & 0 & 0 & \sqrt{q_{5,n+1} - q_{5,n}} & 0 \\ 0 & 0 & \sqrt{2[q_{3,n+1} - q_{3,n}]} & 0 & -\sqrt{q_{5,n+1} - q_{5,n}} & -\sqrt{q_{6,n+1} - q_{6,n}} \end{pmatrix} \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \end{pmatrix} \\
&= \begin{pmatrix} S_{1,n} \\ S_{2,n} \\ S_{3,n} \end{pmatrix} + \frac{1}{V} \begin{pmatrix} VhC_1 S_{2,n} - VhC_2 S_{1,n} S_{2,n} + VhC_3 S_{1,n} - 2VhC_4 S_{1,n}^2 \\ VhC_1 S_{2,n} - VhC_2 S_{1,n} S_{2,n} + VhC_5 \delta S_{3,n} \\ 2VhC_3 S_{1,n} - VhC_5 \delta S_{3,n} - VhC_5 (1 - \delta) S_{3,n} \end{pmatrix} \\
&+ \frac{1}{V} \begin{pmatrix} \sqrt{VhC_1 S_{2,n}} & -\sqrt{VhC_2 S_{1,n} S_{2,n}} & \sqrt{VhC_3 S_{1,n}} & 2\sqrt{VhC_4 S_{1,n}^2} & 0 & 0 \\ -\sqrt{VhC_1 S_{2,n}} & -\sqrt{VhC_2 S_{1,n} S_{2,n}} & 0 & 0 & \sqrt{VhC_5 \delta S_{3,n}} & 0 \\ 0 & 0 & 2\sqrt{VhC_3 S_{1,n}} & 0 & -\sqrt{VhC_5 \delta S_{3,n}} & -\sqrt{VhC_5 (1 - \delta) S_{3,n}} \end{pmatrix} \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \end{pmatrix} \\
&:= \begin{pmatrix} S_{1,n} \\ S_{2,n} \\ S_{3,n} \end{pmatrix} + \frac{1}{V} \begin{pmatrix} \boldsymbol{g_1}(q_{1,n}, \cdots, q_{6,n}) \\ \boldsymbol{g_2}(q_{1,n}, \cdots, q_{6,n}) \\ \boldsymbol{g_3}(q_{1,n}, \cdots, q_{6,n}) \end{pmatrix} + \frac{1}{V} M \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \end{pmatrix},
\end{aligned}
$$

where $W_i, \{i = 1, \cdots, 6\}$ are independent standard normal distributed random variables.

A.3 4D Lotka-Volterra model

For the 4D Lotka-Volterra system, there are 17 reactions involved, so we have 17 pairs of Poisson process P_i and Wiener process B_i. The rule of update of the numerical approximation $\hat{X}_n$ follows

$$
\begin{aligned}
\hat{X}_{n+1} &= \begin{pmatrix} S_{1,n+1} \\ S_{2,n+1} \\ S_{3,n+1} \\ S_{4,n+1} \end{pmatrix} = \begin{pmatrix} S_{1,n} \\ S_{2,n} \\ S_{3,n} \\ S_{4,n} \end{pmatrix} \\
&+ \frac{1}{V} \begin{pmatrix} [P_1(q_{1,n+1}) - P_1(q_{1,n})] - [P_2(q_{2,n+1}) - P_2(q_{2,n})] - [P_3(q_{1,n+1}) - P_3(q_{1,n})] - [P_4(q_{4,n+1}) - P_4(q_{4,n})] \\ [P_5(q_{5,n+1}) - P_5(q_{5,n})] - [P_6(q_{6,n+1}) - P_6(q_{6,n})] - [P_7(q_{7,n+1}) - P_7(q_{7,n})] - [P_8(q_{8,n+1}) - P_8(q_{8,n})] \\ [P_9(q_{9,n+1}) - P_9(q_{9,n})] - [P_{10}(q_{10,n+1}) - P_{10}(q_{10,n})] - [P_{11}(q_{11,n+1}) - P_{11}(q_{11,n})] - [P_{12}(q_{12,n+1}) - P_{12}(q_{12,n})] \\ [P_{13}(q_{13,n+1}) - P_{13}(q_{13,n})] - [P_{14}(q_{14,n+1}) - P_{14}(q_{14,n})] - [P_{15}(q_{15,n+1}) - P_{15}(q_{15,n})] - [P_{16}(q_{16,n+1}) - P_{16}(q_{16,n})] - [P_{17}(q_{17,n+1}) - P_{17}(q_{17,n})] \end{pmatrix} \\
&:= \begin{pmatrix} S_{1,n} \\ S_{2,n} \\ S_{3,n} \\ S_{4,n} \end{pmatrix} + \frac{1}{V} \begin{pmatrix} \boldsymbol{f_1}(P_1, \cdots, P_{17}, q_{i,1}^n, \cdots, q_{i,5}^n) \\ \boldsymbol{f_2}(P_1, \cdots, P_{17}, q_{i,1}^n, \cdots, q_{i,5}^n) \\ \boldsymbol{f_3}(P_1, \cdots, P_{17}, q_{i,1}^n, \cdots, q_{i,5}^n) \\ \boldsymbol{f_4}(P_1, \cdots, P_{17}, q_{i,1}^n, \cdots, q_{i,5}^n) \end{pmatrix},
\end{aligned}
$$

where $i = 1, \cdots, 4$ and $P_j, \{j = 1, \cdots, 17\}$ are independent unit rate Poisson processes. The diffusion approximation $\hat{Y}$ can be written as

$$
\begin{aligned}
\hat{Y}_{n+1} = \begin{pmatrix} S_{1,n+1} \\ S_{2,n+1} \\ S_{3,n+1} \\ S_{4,n+1} \end{pmatrix} &= \begin{pmatrix} S_{1,n} \\ S_{2,n} \\ S_{3,n} \\ S_{4,n} \end{pmatrix} + \frac{1}{V} \begin{pmatrix} [q_{1,n+1}-q_{1,n}]-[q_{2,n+1}-q_{2,n}]-[q_{3,n+1}-q_{3,n}]-[q_{4,n+1}-q_{4,n}] \\ [q_{5,n+1}-q_{5,n}]-[q_{6,n+1}-q_{6,n}]-[q_{7,n+1}-q_{7,n}]-[q_{8,n+1}-q_{8,n}] \\ [q_{9,n+1}-q_{9,n}]-[q_{10,n+1}-q_{10,n}]-[q_{11,n+1}-q_{11,n}]-[q_{12,n+1}-q_{12,n}] \\ [q_{13,n+1}-q_{13,n}]-[q_{14,n+1}-q_{14,n}]-[q_{15,n+1}-q_{15,n}]-[q_{16,n+1}-q_{16,n}]-[q_{17,n+1}-q_{17,n}] \end{pmatrix} \\
&+ \frac{1}{V} \begin{pmatrix} [B_1(q_{1,n+1})-B_1(q_{1,n})]-[B_2(q_{2,n+1})-B_2(q_{2,n})]-[B_3(q_{3,n+1})-B_3(q_{3,n})]-[B_4(q_{4,n+1})-B_4(q_{4,n})] \\ [B_5(q_{5,n+1})-B_5(q_{5,n})]-[B_6(q_{6,n+1})-B_6(q_{6,n})]-[B_7(q_{7,n+1})-B_7(q_{7,n})]-[B_8(q_{8,n+1})-B_8(q_{8,n})] \\ [B_9(q_{9,n+1})-B_9(q_{9,n})]-[B_{10}(q_{10,n+1})-B_{10}(q_{10,n})]-[B_{11}(q_{11,n+1})-B_{11}(q_{11,n})]-[B_{12}(q_{12,n+1})-B_{12}(q_{12,n})] \\ [B_{13}(q_{13,n+1})-B_{13}(q_{13,n})]-[B_{14}(q_{14,n+1})-B_{14}(q_{14,n})]-[B_{15}(q_{15,n+1})-B_{15}(q_{15,n})]-[B_{16}(q_{16,n+1})-B_{16}(q_{16,n})]-[B_{17}(q_{17,n+1})-B_{17}(q_{17,n})] \end{pmatrix} \\
&:= \begin{pmatrix} S_{1,n} \\ S_{2,n} \\ S_{3,n} \\ S_{4,n} \end{pmatrix} + \frac{1}{V} \begin{pmatrix} \boldsymbol{g_1}(q^n_{i,1}, \cdots, q^n_{i,5}) \\ \boldsymbol{g_2}(q^n_{i,1}, \cdots, q^n_{i,5}) \\ \boldsymbol{g_3}(q^n_{i,1}, \cdots, q^n_{i,5}) \\ \boldsymbol{g_4}(q^n_{i,1}, \cdots, q^n_{i,5}) \end{pmatrix} + \frac{1}{V} \begin{pmatrix} \boldsymbol{\sigma_1}(B_1, \cdots, B_{17}, q^n_{i,1}, \cdots, q^n_{i,5}) \\ \boldsymbol{\sigma_2}(B_1, \cdots, B_{17}, q^n_{i,1}, \cdots, q^n_{i,5}) \\ \boldsymbol{\sigma_3}(B_1, \cdots, B_{17}, q^n_{i,1}, \cdots, q^n_{i,5}) \\ \boldsymbol{\sigma_4}(B_1, \cdots, B_{17}, q^n_{i,1}, \cdots, q^n_{i,5}) \end{pmatrix},
\end{aligned}
$$

where $i = 1, \cdots, 4$, $B_j, \{j = 1, \cdots, 17\}$ are independent Wiener processes.

BIBLIOGRAPHY

[1] Agresti, Alan, and Coull, Brent A. Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician 52*, 2 (1998), 119–126.

[2] Anderson, David F, and Kurtz, Thomas G. Continuous time markov chain models for chemical reaction networks. In *Design and analysis of biomolecular circuits*. Springer, 2011, pp. 3–42.

[3] Anderson, William J. *Continuous-time Markov chains: An applications-oriented approach.* Springer Science & Business Media, 2012.

[4] Barton, Russell R, and Schruben, Lee W. Uniform and bootstrap resampling of empirical distributions. In *Proceedings of the 25th conference on Winter simulation* (1993), pp. 503–508.

[5] Benaïm, Michel, Champagnat, Nicolas, and Villemonais, Denis. Stochastic approximation of quasi-stationary distributions for diffusion processes in a bounded domain. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques* (2021), vol. 57, Institut Henri Poincaré, pp. 726–739.

[6] Benaim, Michel, Cloez, Bertrand, Panloup, Fabien, et al. Stochastic approximation of quasi-stationary distributions on compact spaces and applications. *Annals of Applied Probability 28*, 4 (2018), 2370–2416.

[7] Brøns, Morten, and Bar-Eli, Kedma. Canard explosion and excitation in a model of the belousov-zhabotinskii reaction. *The Journal of Physical Chemistry 95*, 22 (1991), 8706–8713.

[8] Burdzy, Krzysztof, Hołyst, Robert, and March, Peter. A fleming–viot particle representation¶ of the dirichlet laplacian. *Communications in Mathematical Physics 214*, 3 (2000), 679–703.

[9] Cloez, Bertrand, and Thai, Marie-Noémie. Quantitative results for the fleming–viot particle system and quasi-stationary distributions in discrete space. *Stochastic Processes and their Applications 126*, 3 (2016), 680–702.

[10] Collet, Pierre, Martínez, Servet, and San Martín, Jaime. *Quasi-stationary distributions: Markov chains, diffusions and dynamical systems.* Springer Science & Business Media, 2012.

[11] Darroch, John N, and Seneta, Eugene. On quasi-stationary distributions in absorbing discrete-time finite markov chains. *Journal of Applied Probability 2*, 1 (1965), 88–100.

[12] Deutsch, Emeric, and Neumann, Michael. On the first and second order derivatives of the perron vector. *Linear algebra and its applications 71* (1985), 57–76.

[13] Dieu, Nguyen Thanh, Nguyen, Dang Hai, Du, Nguyen Huu, and Yin, G. Classification of asymptotic behavior in a stochastic sir model. *SIAM Journal on Applied Dynamical Systems 15*, 2 (2016), 1062–1084.

[14] Dobson, Matthew, Li, Yao, and Zhai, Jiayu. An efficient data-driven solver for fokker-planck equations: algorithm and analysis. *Communications in Mathematical Sciences, accepted* (2021).

[15] Dobson, Matthew, Li, Yao, and Zhai, Jiayu. Using coupling methods to estimate sample quality of stochastic differential equations. *SIAM/ASA Journal on Uncertainty Quantification 9*, 1 (2021), 135–162.

[16] Epstein, Irving R, and Pojman, John A. *An introduction to nonlinear chemical dynamics: oscillations, waves, patterns, and chaos.* Oxford University Press, 1998.

[17] Ferrari, Pablo A, Martínez, Servet, and Picco, Pierre. Existence of non-trivial quasi-stationary distributions in the birth-death chain. *Advances in applied probability* (1992), 795–813.

[18] Gibson, Michael A, and Bruck, Jehoshua. Efficient exact stochastic simulation of chemical systems with many species and many channels. *The journal of physical chemistry A 104*, 9 (2000), 1876–1889.

[19] Gillespie, Daniel T, et al. Stochastic simulation of chemical kinetics. *Annual review of physical chemistry 58*, 1 (2007), 35–55.

[20] Gray, P, and Scott, SK. Chemical oscillations and instabilities; nonlinear chemical kinetics.() xford, 1990.

[21] Hening, Alexandru, and Li, Yao. Stationary distributions of persistent ecological systems. *Journal of Mathematical Biology 82*, 7 (2021), 1–53.

[22] Huillet, Thierry. On wright–fisher diffusion and its relatives. *Journal of Statistical Mechanics: Theory and Experiment 2007*, 11 (2007), P11006.

[23] Johndrow, James E, and Mattingly, Jonathan C. Error bounds for approximations of markov chains used in bayesian sampling. *arXiv preprint arXiv:1711.05382* (2017).

[24] Karatzas, Ioannis, and Shreve, Steven. *Brownian motion and stochastic calculus*, vol. 113. springer, 2014.

[25] Kloeden, Peter E, and Platen, Eckhard. Stochastic differential equations. In *Numerical Solution of Stochastic Differential Equations*. Springer, 1992, pp. 103–160.

[26] Komlós, János, Major, Péter, and Tusnády, Gábor. An approximation of partial sums of independent rv'-s, and the sample df. i. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete 32*, 1 (1975), 111–131.

[27] Komlós, János, Major, Péter, and Tusnády, Gábor. An approximation of partial sums of independent rv's, and the sample df. ii. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete 34*, 1 (1976), 33–58.

[28] Lai, Ying-Cheng, and Tél, Tamás. *Transient chaos: complex dynamics on finite time scales*, vol. 173. Springer Science & Business Media, 2011.

[29] Li, Yao. A data-driven method for the steady state of randomly perturbed dynamics. *Communications in Mathematical Sciences 17*, 4 (2019).

[30] Li, Yao, and Hu, Lili. A fast exact simulation method for a class of markov jump processes. *The Journal of chemical physics 143*, 18 (2015), 184105.

[31] Li, Yao, and Wang, Shirou. Numerical computations of geometric ergodicity for stochastic dynamics. *Nonlinearity 33*, 12 (2020), 6935.

[32] Li, Yao, and Yuan, Yaping. Data-driven computational methods for quasi-stationary distribution and sensitivity analysis. *Journal of Dynamics and Differential Equations* (2022), 1–29.

[33] Lotka, Alfred J. Elements of physical biology. *Science Progress in the Twentieth Century (1919-1933) 21*, 82 (1926), 341–343.

[34] Meyer, Carl D. Sensitivity of the stationary distribution of a markov chain. *SIAM Journal on Matrix Analysis and Applications 15*, 3 (1994), 715–728.

[35] Miyazaki, Jun. *Pattern Formations and Oscillatory Phenomena: 2. Belousov–Zhabotinsky Reaction*. Elsevier Inc. Chapters, 2013.

[36] Mozgunov, Pavel, Beccuti, Marco, Horvath, Andras, Jaki, Thomas, Sirovich, Roberta, and Bibbona, Enrico. A review of the deterministic and diffusion approximations for stochastic chemical reaction networks. *Reaction Kinetics, Mechanisms and Catalysis 123*, 2 (2018), 289–312.

[37] Murray, James D. *Mathematical biology II: spatial models and biomedical applications*, vol. 3. Springer New York, 2001.

[38] Øksendal, Bernt. Stochastic differential equations. In *Stochastic differential equations*. Springer, 2003, pp. 65–84.

[39] Pilipenko, Andrey. *An introduction to stochastic differential equations with reflection*, vol. 1. Universitätsverlag Potsdam, 2014.

[40] Slepoy, Alexander, Thompson, Aidan P, and Plimpton, Steven J. A constant-time kinetic monte carlo algorithm for simulation of large biochemical reaction networks. *The journal of chemical physics 128*, 20 (2008), 05B618.

[41] Van Doorn, Erik A. Quasi-stationary distributions and convergence to quasi-stationarity of birth-death processes. *Advances in Applied Probability* (1991), 683–700.

[42] van Doorn, Erik A, and Schrijner, Pauline. Geomatric ergodicity and quasi-stationarity in discrete-time birth-death processes. *The ANZIAM Journal 37*, 2 (1995), 121–144.

[43] Villemonais, Denis. Interacting particle systems and yaglom limit approximation of diffusions with unbounded drift. *Electronic Journal of Probability 16* (2011), 1663–1692.

[44] Volterra, Vito. Variazioni e fluttuazioni del numero d'individui in specie animali conviventi.

[45] Zhai, Jiayu, Dobson, Matthew, and Li, Yao. A deep learning method for solving fokker-planck equations. In *Mathematical and Scientific Machine Learning* (2022), PMLR, pp. 568–597.