สมการเชิงอนุพันธ์สโตแคสติกที่มีการกระโดดสำหรับประชากรปลานิล

นายคณิติน สุขชุม

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562

STOCHASTIC DIFFERENTIAL EQUATION WITH JUMPS

FOR TILAPIA POPULATION

Mr. Kanitin Sukchum

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Applied Mathematics and

Computational Science

Department of Mathematics and Computer Science

Faculty of Science

Chulalongkorn University

Academic Year 2019

Thesis Title	STOCHASTIC DIFFERENTIAL EQUATION WITH JUMPS FOR TILAPIA POPULATION
By	Mr. Kanitin Sukchum
Field of Study	Applied Mathematics and Computational Science
Thesis Advisor	Associate Professor Khamron Mekchay, Ph.D.
Thesis Co-advisor	Raywat Tanadkithirun, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

. . . . Dean of the Faculty of Science

(Professor Polkit Sangvanich, Ph.D.)

THESIS COMMITTEE

. . . . Chairman

(Associate Professor Petarpa Boonserm, Ph.D.)

. . . . Thesis Advisor

(Associate Professor Khamron Mekchay, Ph.D.)

. . . . Thesis Co-advisor

(Raywat Tanadkithirun, Ph.D.)

. . . . Examiner

(Assistant Professor Boonyarit Intiyot, Ph.D.)

. . . . External Examiner

(Sirod Sirisup, Ph.D.)

คณิติน สุขชุม : สมการเชิงอนุพันธ์สโตแคสติกที่มีการกระโดดสำหรับประชากรปลานิล. (STOCHASTIC DIFFERENTIAL EQUATION WITH JUMPS FOR TILAPIA POPULATION) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : รศ.ดร.คำรณ เมฆฉาย, อ.ที่ปรึกษาวิทยานิพนธ์ร่วม : อ.ดร.เรวัต ถนัดกิจหิรัญ, 64 หน้า.

ประชากรปลานิลที่มีการเก็บเกี่ยวในฟาร์มปลานิลสามารถถูกโมเดลได้ในรูปแบบของสมการเชิงอนุพันธ์สามัญ ในความเป็นจริงปลานิลมักจะได้รับผลกระทบจากหลาย ๆ ปัจจัย เพื่อให้แบบจำลองสำหรับประชากรปลานิลมีความสมจริงมากยิ่งขึ้น เราจึงพัฒนาสมการเชิงอนุพันธ์สามัญให้อยู่ในรูปแบบของสมการเชิงอนุพันธ์สโตแคสติกที่มีการกระโดดซึ่งแสดงถึงการเกิดโรคระบาดที่เกิดขึ้นกับปลานิล นอกจากนี้เราได้ศึกษาถึงผลกระทบจากพารามิเตอร์ที่สำคัญบางตัวที่มีต่อจำนวนประชากรปลานิลในฟาร์มปลานิลผ่านการจำลองทั้งสมการเชิงอนุพันธ์สโตแคสติกสำหรับประชากรปลานิลที่มีการกระโดดและไม่มีการกระโดดจะถูกจำลอง โดยใช้วิธีการของ ออยเลอร์-มารุยามาและวิธีออยเลอร์ที่ดัดแปลงแบบกระโดด

ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	ลายมือชื่อนิสิต ....
		ลายมือชื่อ อ.ที่ปรึกษาหลัก ....
สาขาวิชา	คณิตศาสตร์ประยุกต์และวิทยาการคณนา	ลายมือชื่อ อ.ที่ปรึกษาร่วม ....
ปีการศึกษา	2562	

ID : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE

KEYWORDS : TILAPIA POPULATION / STOCHASTIC DIFFERENTIAL EQUATION / HARVESTING / JUMPS / EPIDEMIC OCCURRING IN TILAPIA

KANITIN SUKCHUM : STOCHASTIC DIFFERENTIAL EQUATION WITH JUMPS FOR TILAPIA POPULATION. ADVISOR : ASSOC. PROF. Khamron Mekchay, Ph.D., THESIS COADVISOR : Raywat Tanadkithirun, Ph.D., 64 pp.

Tilapia population with harvesting in the tilapia farm can be modeled by an ordinary differential equation (ODE). In real life, tilapia population can be affected by many factors. To make the model for tilapia population more realistic, we develop the ODE into the stochastic differential equation (SDE) together with jumps that represent the epidemic occuring for tilapia. Furthermore, we study the effect of some important parameters in the model to the number of tilapia in the tilapia farm via simulation. Both SDEs for tilapia population with and without jumps are simulated by using Euler-Maruyama and jump-adapted Euler methods.

Department	: Mathematics and Computer Science	Student's Signature	....
		Advisor's Signature	....
Field of Study	: Applied Mathematics and Computational Science	Co-advisor's Signature	....
Academic Year	: 2019		

ACKNOWLEDGEMENTS

This thesis can be accomplished with assistance from many persons. I would like to express my sincere thanks and deep appreciation to my thesis advisors, Associate Professor Dr.Khamron Mekchay and Dr.Raywat Tanadkithirun, for giving me the motivation, basic backgrounds and suggestions for this thesis.

Furthermore, I would like to express my sincere thanks and deep appreciation to my committee members, Associate Professor Dr.Petarpa Boonserm, Assistant Professor Dr.Boonyarit Intiyot and Dr.Sirod Sirisup for their invaluable advices and comments in this thesis.

Finally, I feel very grateful to all my friends and colleagues for encouragement and good suggestions. Also, I wish to express my gratitude to my family for their encouragement throughout my study.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

A stochastic differential equation (SDE) with jumps for tilapia population is the model that represents the number of tilapia population under various conditions. In this section, we describe how the SDE with jumps has evolved from an ordinary differential equation (ODE) for tilapia population. Gertjan et al.[1] proposed that tilapia can produce the population in a short time duration in a proper environment. Laham et al.[2] modeled the estimation for the tilapia population with harvesting by an ODE

$$dX_t = \left(rX_t \left(1 - \frac{X_t}{K} \right) - H(t) \right) dt,$$

where t is the time (in months), X_t is the population size of tilapia at time t (in fish), $r > 0$ is the rate of the tilapia that survive at maturity stage (in fish per month), $H(t)$ is a harvest function and $K > 0$ is the original carrying capacity. He obtained the data for the model of tilapia population from the fish owner of selected ponds suggested by the Department of Fisheries of Malaysia situated at Gombak, Selangor, Malaysia. The Department of Fisheries of Malaysia claimed the following statements: (i) the fish pond can sustain 5 tilapia fish for every 1 square meter, (ii) the selected pond has an area of 156100 square meters so that the original carrying capacity of this selected pond is 780500 fish, and (iii) the rate of the tilapia that survives at maturity stage is 0.8 per month. For each year, the selected pond will be harvested with the rate of 156100 fish per month for the first 6 months, but for the rest 6 months, the selected pond will not be harvested.

Hence, the harvest function for this selected pond is

$$H(t) = \begin{cases} 156100, & \text{if} \quad t \in (0, 6] \\ 0, & \text{if} \quad t \in (6, 12] \end{cases},$$

and

$$H(t + 12) = H(t),$$

where $t \geq 0$.

Since the carrying capacity of the tilapia population should be seasonal, Asaduzzaman et al. [3] considered the periodic carrying capacity

$$K(t) = K_0 \left(1 + \varepsilon \cos\left(\frac{2\pi t}{T_0}\right)\right), \tag{1}$$

where $K_0 > 0$ is the original carrying capacity, T_0 is a period of seasonal oscillations in the carrying capacity which equals to 12 months, $\varepsilon > 0$ is the proportion of extreme varying for the carrying capacity, which is much less than 1, with the condition $K(t) = K(t + 12)$ for all $t \geq 0$. Since the carrying capacity for the tilapia population have the highest rate at the middle of the summer, in this work, we let April 1st, which is the middle of the summer, be the starting time $t = 0$. Furthermore, Asaduzzaman et al. [3] modeled the SDE as the form

$$\frac{dX_t}{dt} = \left(rX_t\left(1 - \frac{X_t}{K(t)}\right) - H(t)\right) + \zeta X_t W_t,$$

where $\zeta > 0$ and W_t is a Wiener process.

However, we are interested in only $K(t)$ in this model, so our ODE model for the tilapia population has a form

$$dX_t = \left(rX_t \left(1 - \frac{X_t}{K_0 \left(1 + \varepsilon \cos\left(\frac{2\pi t}{T_0} \right) \right)} \right) - H(t) \right) dt. \tag{2}$$

Since there may be environmental factors that have little impact on the tilapia population, the tilapia population should be not deterministic. With this reason, we add a diffusion term which depends on the current number of tilapia fish into (2), so that the tilapia population will have some random perturbation in time. Here, we use a Wiener process as a noise for the model. Thus, ODE (2) can be developed into an SDE for tilapia population which has the form

$$dX_t = \left(rX_t \left(1 - \frac{X_t}{K_0 \left(1 + \varepsilon \cos\left(\frac{2\pi t}{T_0} \right) \right)} \right) - H(t) \right) dt + \zeta X_t^{\kappa} dW_t, \tag{3}$$

where $\kappa \in [0, 1]$. Since there might be an epidemic occurring with tilapia population, we add a jump term depending on the current number of tilapia fish into the SDE (3). Thus, we have SDE with jumps for tilapia population

$$dX_t = \left(rX_t \left(1 - \frac{X_t}{K_0 \left(1 + \varepsilon \cos\left(\frac{2\pi t}{T_0} \right) \right)} \right) - H(t) \right) dt + \zeta X_t^{\kappa} dW_t - \sum_{i=1}^{4} X_{t^-}^{\eta_i} dJ_t^{(i)}, \tag{4}$$

where $X_{t^-} = \lim_{s \to t^-} X_s$, $\eta_i \in [0, 1]$, $J_t^{(i)}$ is the inhomogeneous compound Poisson process with intensity function $\lambda_i(t)$ and jump size distribution $D^{(i)}$, when $D^{(i)}$ is a beta distribution or a logit-normal distribution for $i = 1, ..., 4$ representing 4 diseases: Columnaris, Epitheliocystis, Red egg and Streptococcus, respectively.

From the SDE (3) and the SDE with jumps (4), it is possible that the process X_t will become negative at some time; consequently, there is no solution for the SDEs with some certain parameters after that time. Let $\tau = \inf\{t > 0 \mid X_t \leq 0\}$ be the time that the number of tilapia population becomes zero. We will set

$X_t = 0$ for all $t > \tau$.

We discuss about the model for tilapia population (4) in more details in chapter 3. The simulation for the models (3) and (4) by using Euler-Maruyama and jump-adapted Euler, respectively, are presented in chapter 2. Then, we explain the simulation results in chapter 4 and conclude our work in chapter 5. Our model (4) for tilapia population is probably a good choice for studying the trend of the tilapia population in the future under the various factors such as epidemic in order to prepare and cope effectively with various conditions that may affect the tilapia population.

CHAPTER II

BACKGROUND KNOWLEDGE

This chapter provides basic knowledge about SDE with jumps (4) in section 2.1. Furthermore, we introduce some numerical methods that are used in this work in section 2.2 and 2.3.

2.1 Introduction to SDE

Definition 2.1.1. (Stochastic Process)[4]

Let I be a subset of $[0, \infty)$. A collection of random variables $\{X_t\}_{t \in I}$, indexed by I, is called a **stochastic process**.

Definition 2.1.2. (Continuous Sample Path)[4]

Let $\{X_t\}_{t \in I}$ be a stochastic process on a probability space $(\Omega, \mathcal{F}, P)$. For a fixed $\omega \in \Omega$, a function $X_\cdot(\omega) : I \to \mathbb{R}$ is called a **sample path**. If $X_\cdot(\omega)$ is a continuous function for all $\omega \in \Omega$, the process $\{X_t\}_{t \in I}$ is said to have **continuous sample paths**.

Definition 2.1.3. (Normal Distribution)[5]

A random variable X is said to have a **normal distribution** with parameters $\mu \in \mathbb{R}$ and $\sigma^2 > 0$, denoted by $X \sim \mathcal{N}(\mu, \sigma^2)$, if its probability density function is given by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad , \quad x \in \mathbb{R}.$$

If $X \sim \mathcal{N}(0, 1)$, then we say that X has a **standard normal distribution**, i.e., its probability density function is given by

$$g(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad , \quad x \in \mathbb{R}.$$

Proposition 2.1.1. If $X \sim \mathcal{N}(\mu, \sigma^2)$, then

$$E[X] = \mu$$

and

$$Var[X] = \sigma^2.$$

Definition 2.1.4. (Standard Brownian Motion)[6]

A **standard Brownian motion**, or **standard Wiener process** on $[0, T]$, is a stochastic process $\{W_t\}_{t \in [0,T]}$ which satisfies the following conditions.

1. $W_0 = 0$, with probability 1.
2. For $0 \leq s < t \leq T, W_t - W_s \sim \mathcal{N}(0, t - s)$.
3. For $0 \leq s < t \leq u < v \leq T$, the increments $W_t - W_s$ and $W_v - W_u$ are independent.
4. $\{W_t\}_{t \in [0,T]}$ has continuous sample paths.

Definition 2.1.5. (Itô Stochastic Integral)[4]

Let $P_n = \{t_0, t_1, t_2, ..., t_n\}$ where $0 = t_0 < t_1 < t_2 < ... < t_n = t$ be a partition for the closed interval $[0, t]$. Define $\|P_n\| = \max_{i \in \{1,2,...,n\}} (t_i - t_{i-1})$. Then, an **Itô stochastic integral**

$$\int_0^t \sigma(X_s) dW_s$$

can be defined by

$$\lim_{\|P_n\| \to 0} \sum_{i=1}^{n} \sigma(X_{t_{i-1}})(W_{t_i} - W_{t_{i-1}}).$$

Definition 2.1.6. (Stochastic Differential Equations)[4]

An **SDE** typically has the form

$$dX_t = F(X_t)dt + G(X_t)dW_t,$$

where F and G are real-valued functions, $\{W_t\}_{t\in[0,T]}$ is a standard Brownian motion and X_0 is a constant. This equation is a differential form which should be understood as the stochastic integral equation

$$X_t = X_0 + \int_0^t F(X_s)ds + \int_0^t G(X_s)dW_s,$$

where $\int_0^t F(X_s)ds$ is a Riemann integral and $\int_0^t G(X_s)dW_s$ is an Itô integral.

Theorem 2.1.1. (Itô Formula)[6]

Let X_t be a solution of the SDE

$$dX_t = F(X_t)dt + G(X_t)dW_t$$

and $f(x,t)$ is a function such that f_x, f_{xx} and f_t exist. Then, the SDE for $f(X_t,t)$ is given by

$$df(X_t,t) = f_t(X_t,t)dt + f_x(X_t,t)dX_t + \frac{1}{2}f_{xx}(X_t,t)(dX_t \cdot dX_t),$$

where

$$dt \cdot dt = dt \cdot dW_t = dW_t \cdot dt = 0$$

and

$$dW_t \cdot dW_t = dt.$$

Therefore,

$$df(X_t,t) = \left(f_t(X_t,t) + f_x(X_t,t)F(X_t) + \frac{1}{2} f_{xx}(X_t,t)(G(X_t))^2 \right) dt$$
$$+ \left(f_x(X_t,t)G(X_t) \right) dW_t.$$

Definition 2.1.7. (Beta Distribution)[5]

A random variable X is said to have a **beta distribution** with parameters $\alpha > 0$ and $\beta > 0$, denoted by $X \sim Beta(\alpha, \beta)$, if its probability density function is given by

$$f(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)} \quad , \quad x \in (0,1),$$

where

$$B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$$

and Γ is the Gamma function

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt.$$

Proposition 2.1.2. If $X \sim Beta(\alpha, \beta)$, then

$$E[X] = \frac{\alpha}{\alpha+\beta}$$

and

$$Var[X] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}.$$

Definition 2.1.8. (Logit-Normal Distribution)[7]

A random variable X is said to have a **logit-normal distribution** with parameters $\mu \in \mathbb{R}$ and $\sigma^2 > 0$, denoted by $X \sim P(\mathcal{N}(\mu, \sigma^2))$, if its probability density function is given by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\text{logit}(x)-\mu)^2}{2\sigma^2}} \frac{1}{x(1-x)} \quad , \quad x \in (0,1),$$

where

$$\text{logit}(x) = \log\left(\frac{x}{1-x}\right).$$

Proposition 2.1.3. If $X \sim P(\mathcal{N}(\mu, \sigma^2))$, then the location of the mode x is given by

$$\text{logit}(x) = \sigma^2(2x-1) + \mu,$$

and the median is

$$\frac{1}{1+e^{-\mu}}.$$

Definition 2.1.9. (Exponential Distribution)[8]

A random variable X is said to have an **exponential distribution** with parameters $\lambda > 0$, denoted by $X \sim Exp(\lambda)$, if its probability density function is given by

$$f(x) = \lambda e^{-\lambda x}, \quad x \geq 0.$$

Proposition 2.1.4. If $X \sim Exp(\lambda)$, then

$$E[X] = \frac{1}{\lambda}$$

and

$$Var[X] = \frac{1}{\lambda^2}.$$

Definition 2.1.10. (Poisson Distribution)[5]

A random variable X is said to have a **Poisson distribution** with parameters

$\lambda > 0$, denoted by $X \sim Poi(\lambda)$, if its probability mass function is given by

$$P(X = k) = \frac{e^{-\lambda}\lambda^k}{k!} \quad , \quad k \in \mathbb{N} \cup \{0\}.$$

Proposition 2.1.5. If $X \sim Poi(\lambda)$, then

$$E[X] = \lambda$$

and

$$Var[X] = \lambda.$$

Definition 2.1.11. (Point Process)[9]

A **point process** on $[0, \infty)$ with an infinite number of strictly positive jump instants without accumulation point, is a process $\{N_t, t \geq 0\}$ with values in $\mathbb{N} \cup \{0\}$, vanishing at 0, non-decreasing, right continuous, with unit jumps, and with infinite limit, i.e., for $0 \leq s \leq t < \infty$,

$$0 = N_0 \leq N_s \leq N_t = N_{t^+},$$

$$N_t - N_{t^-} \in \{0, 1\},$$

$$\lim_{t \to \infty} N_t = \infty,$$

with the notation $N_{t^+} = \lim_{u \to t^+} N_u$, $N_{t^-} = \lim_{u \to t^-} N_u$ and $N_{0^-} = N_0$.

Definition 2.1.12. (Poisson Process)[9]

A **Poisson process** $\{N_t, t \geq 0\}$ with parameter $\lambda > 0$ is a point process satisfying the following conditions.

1. $N_0 = 0$.

2. For $0 \leq s < t, N_t - N_s \sim Poi(\lambda(t - s))$.

3. For $0 \leq s < t \leq u < v$, the increments $N_t - N_s$ and $N_v - N_u$ are independent.

The parameter λ is called the **intensity or rate** of the Poisson process.

Definition 2.1.13. (Inhomogeneous Poisson Process)[10]
A point process $\{N_t, t \geq 0\}$ is said to be an inhomogeneous Poisson process with intensity function $\lambda(t) \geq 0$, if

1. $N_0 = 0$ with probability 1.
2. The process $\{N_t, t \geq 0\}$ is the point process with independent increments and right continuous piecewise constant trajectories.
3. For $h > 0$,

$$P(N_{t+h} - N_t = k) = \frac{(\int_t^{t+h} \lambda(x)dx)^k}{k!} e^{-\int_t^{t+h} \lambda(x)dx}.$$

Definition 2.1.14. (Compound Poisson Process)[9]
Let $\{N_t, t \geq 0\}$ be a Poisson process with intensity λ and $\{D_i\}_{i\in\mathbb{N}}$ a sequence of independent and identically distributed random variables with distribution D, and $\{D_i\}_{i\in\mathbb{N}}$ is independent of $\{N_t, t \geq 0\}$. Define

$$J_t = \sum_{i=1}^{N_t} D_i,$$

where $\sum_{i=1}^{0} D_i$ is defined to be zero. Then, $\{J_t, t \geq 0\}$ is called a **compound Poisson process** with **intensity** λ and **jump size distribution** D.

Definition 2.1.15. (Inhomegeneous Compound Poisson Process)[11]
Let $\{N_t, t \geq 0\}$ be an inhomogeneous Poisson process with intensity function $\lambda(t)$ and $\{D_i\}_{i\in\mathbb{N}}$ a sequence of independent and identically distributed random

variables with distribution D, and $\{D_i\}_{i\in\mathbb{N}}$ is independent of $\{N_t, t \geq 0\}$. Define

$$J_t = \sum_{i=1}^{N_t} D_i,$$

where $\sum_{i=1}^{0} D_i$ is defined to be zero. Then, $\{J_t, t \geq 0\}$ is called an **inhomogeneous compound Poisson process** with **intensity function** $\lambda(t) \geq 0$ and **jump size distribution** D.

Definition 2.1.16. (Stochastic Integrals with Jumps)[9]
Let $\{N_t, t \geq 0\}$ be an inhomogeneous Poisson process with intensity $\lambda(t)$ and $\{J_t, t \geq 0\}$ the corresponding inhomogeneous compound Poisson process with jump size distribution D. We can define a stochastic integral of a stochastic process $\{\phi_t, t \geq 0\}$ with respect to $\{J_t, t \geq 0\}$ by

$$\int_0^T \phi_t dJ_t = \int_0^T \phi_t D_{N_t} dN_t = \sum_{i=1}^{N_T} \phi_{T_i} D_i,$$

where T_i's are jump instants of the process $\{N_t, t \geq 0\}$.

Definition 2.1.17. (Total variation for functions of one real variable) [12]
The **total variation** of a real-valued function f, defined on an interval $[a, b] \subset \mathbb{R}$ is the quantity

$$V_b^{(a)}(f) = \sup_{\mathcal{P}} \sum_{i=0}^{n_P - 1} |f(x_{i+1}) - f(x_i)|,$$

where the supremum runs over the set of all partitions $\mathcal{P} = \{P = \{x_0, ..., x_{n_P}\} \mid P$ is a partition of $[a, b]\}$ of the given interval.

2.2 Introduction to Numerical Methods for SDEs

2.2.1 Euler-Maruyama

For an SDE

$$dX_t = F(X_t)dt + G(X_t)dW_t, \quad t \in [0, T], \tag{4}$$

where X_0 is a constant, and W_t is a Wiener process. The Euler-Maruyama method is a numerical method to approximate a numerical solution of an SDE. It has the following procedure [11].

1. Discretize the interval $[0, T]$ into N equal pieces for some $N \in \mathbb{N}$ and let $\Delta t = \frac{T}{N}$.
2. Define $t_n = n\Delta t$ and denote the numerical solution of X_{t_n} by x_n for $n = 0, ..., N$.
3. The Euler-Maruyama scheme for the SDE (4) has the form

$$\begin{aligned} x_0 &= X_0, \\ x_n &= x_{n-1} + F(x_{n-1})\Delta t + G(x_{n-1})\Delta W_n, \text{ for } n = 1, 2, ..., N, \end{aligned}$$

where

$$\Delta W_n = W_{t_n} - W_{t_{n-1}} \sim \mathcal{N}(0, \Delta t).$$

2.2.2 Jump-Adapted Euler

For an SDE with jumps

$$dX_t = F(X_t)dt + G(X_t)dW_t + \sum_{i=1}^{4} V_i(X_{t^-})dJ_t^{(i)}, \quad t \in [0, T], \tag{5}$$

where X_0 is a constant, W_t is a Wiener process, and $J_t^{(i)}$ is an inhomogeneous compound Poisson process with intensity $\lambda_i(t)$ and jump size distribution $D^{(i)}$ for $i = 1, 2, 3, 4$. The jump-adapted Euler method is a numerical method to approximate a numerical solution of an SDE with jumps. It has the following procedure [11].

1. Discretize the interval $[0, T]$ into M equal pieces for some $M \in \mathbb{N}$ and let $\Delta t = \frac{T}{M}$.
2. Define $\tau_m = m\Delta t$ for $m = 0, ..., M$.
3. For all $i = 1, ..., 4$, generate all inhomogeneous jump instants between 0 and T, namely $\hat{\tau}_p^{(i)}$ for $p = 1, ..., Q^{(i)}$, where $Q^{(i)}$ is the number of jump instants, from the inhomogeneous Poisson process with intensity $\lambda_i(t)$.
4. Let $\{t_n \mid n = 0, 1, ..., N\} = \bigcup_{i=1}^{4} \{\hat{\tau}_p^{(i)} \mid p = 1, ..., Q^{(i)}\} \bigcup \{\tau_m \mid m = 0, ..., M\}$ be a jump-adapted time discretization, where $t_0 < t_1 < ... < t_N$.
5. Denote the numerical solution of X_{t_n} by x_n for $n = 0, ..., N$.
6. The jump-adapted Euler scheme for the SDE with jumps (5) has the form

$$
\begin{aligned}
x_0 &= X_0, \\
x_n^- &= x_{n-1} + F(x_{n-1})\Delta t_n + G(x_{n-1})\Delta W_n, \\
x_n &= x_n^- + \sum_{i=1}^{4} V_i(x_n^-)\Delta J_n^{(i)}, \text{ for } n = 1, 2, ..., N,
\end{aligned}
$$

where

$$\Delta t_n = t_n - t_{n-1},$$

$$\Delta W_n = W_{t_n} - W_{t_{n-1}} \sim \mathcal{N}(0, \Delta t_n),$$

$$\Delta J_n^{(i)} = J_{t_n}^{(i)} - J_{t_{n-1}}^{(i)} = \sum_{k=N_{t_{n-1}}^{(i)}+1}^{N_{t_n}^{(i)}} D_k^{(i)}, \text{ where } D_k^{(i)} \sim D^{(i)}.$$

2.3 Acceptance-Rejection Method

In this work, we use MATLAB to simulate our tilapia population model. We can use built-in MATLAB functions for generating random numbers from most well-known distributions. However, we need to use the acceptance-rejection method to generate random numbers from the logit-normal distribution. The acceptance-rejection method, initiated by Von Neumann [13], is the method that generates samples from a target distribution by first generating candidates from a more convenient distribution and then rejecting a random subset of the generated candidates. The rejection mechanism is designed so that the accepted samples are indeed distributed according to the target distribution. The technique is by no means restricted to univariate distributions.

Suppose that we wish to generate samples from a density f defined on some set χ. This could be a subset of the real line, of $\mathbb{R}^d$, or a more general set. Let g be a density on χ from which we know how to generate samples and with the property that

$$f(x) \leq cg(x), \quad \text{for all}\, x \in \chi$$

for some constant c. In the acceptance-rejection method, we generate a sample X from g and accept the sample with probability $\frac{f(X)}{cg(X)}$; this can be implemented by sampling U uniformly over (0,1) and accepting X if $U \leq \frac{f(X)}{cg(X)}$. If X is rejected, a new candidate is sampled from g and the acceptance test is applied again. The process repeats until the acceptance test is passed, and the accepted value is returned as a sample from f.

CHAPTER III

SDE WITH JUMPS MODEL OF TILAPIA POPULATION

In this chapter, we describe in details of the harvest functions in section 3.1, the diffusion term in section 3.2, and the jump terms in section 3.3.

3.1 Harvest Function

Recall that we have a harvest function used in a farm in Malaysia [2]. In this work, we not only consider the harvest function introduced in [2] but also define other 5 harvest functions that have different behaviors including the harvestment in Thailand. The 6 harvest functions are given by

$$H_1(t) = \begin{cases} 156100, & \text{if} \quad t \in (0,6] \\ 0, & \text{if} \quad t \in (6,12] \end{cases},$$

$$H_2(t) = 78050, \quad \text{if} \quad t \in (0,12],$$

$$H_3(t) = \begin{cases} 156100, & \text{if} \quad t \in (0,2] \\ 0, & \text{if} \quad t \in (2,4] \\ 156100, & \text{if} \quad t \in (4,6] \\ 0, & \text{if} \quad t \in (6,8] \\ 156100, & \text{if} \quad t \in (8,10] \\ 0, & \text{if} \quad t \in (10,12] \end{cases},$$

$$H_4(t) = \begin{cases} 156100\left(1 + 0.1\cos\left(\dfrac{2\pi t}{12}\right)\right), & \text{if} \quad t \in (0,6] \\ 0, & \text{if} \quad t \in (6,12] \end{cases},$$

$$H_5(t) = 78050\left(1 + 0.1\cos\left(\dfrac{2\pi t}{12}\right)\right), \qquad \text{if} \quad t \in (0,12]$$

$$H_6(t) = \begin{cases} 171710, & \text{if} \quad t \in (0,2] \\ 0, & \text{if} \quad t \in (2,4] \\ 140490, & \text{if} \quad t \in (4,6] \\ 0, & \text{if} \quad t \in (6,8] \\ 171710, & \text{if} \quad t \in (8,10] \\ 0, & \text{if} \quad t \in (10,12] \end{cases},$$

and

$$H_i(t+12) = H_i(t),$$

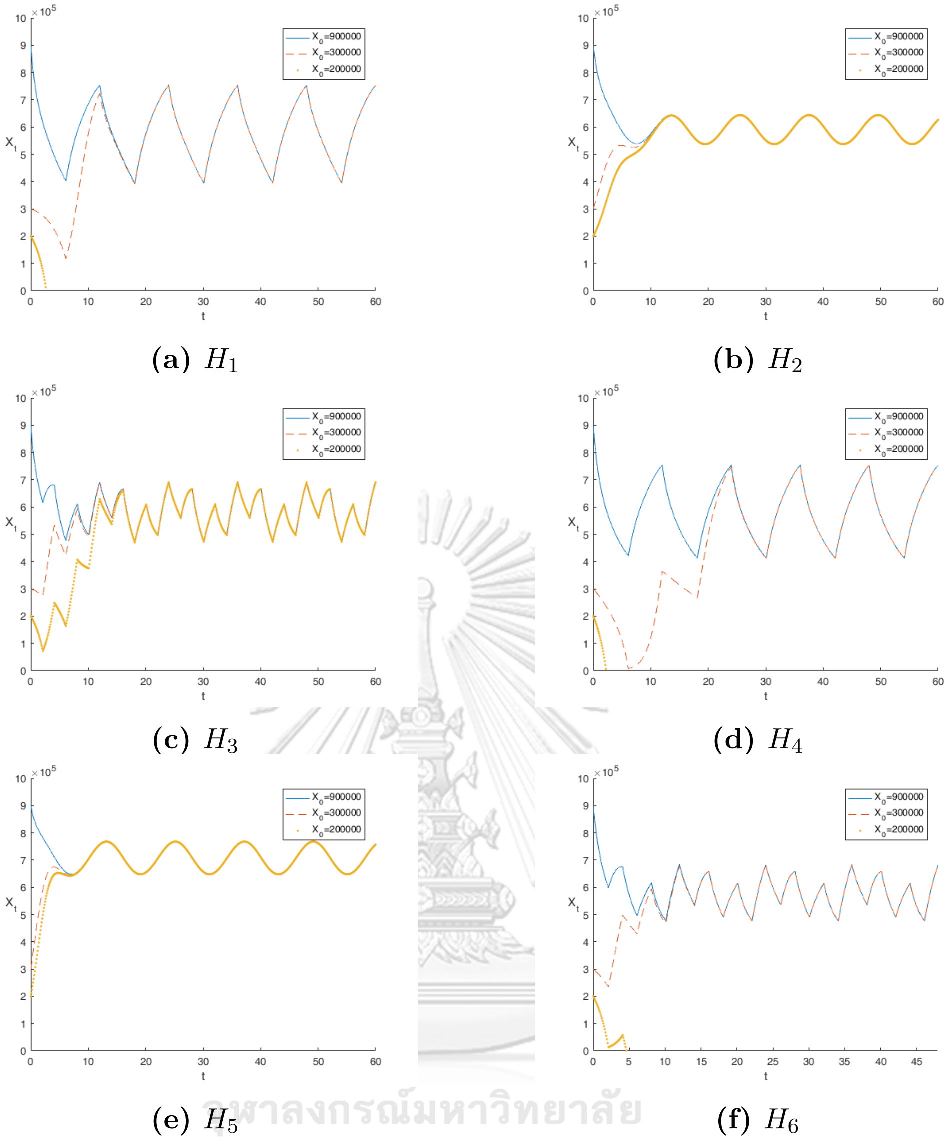
where $t \geq 0$ for $i = 1, 2, 3, 4, 5, 6$. We describe the motivation of defining these 6 harvest functions as follows.

The first harvest function H_1 is the same as in [2]. For each year, a pond will be harvested with the rate of 156100 fish per month for the first 6 months, but for the rest 6 months, the pond will not be harvested. As for the harvest function H_2, we define it to make the tilapia population be harvested all year round with a constant rate of 78050 fish per month, which equals to the half of 156100. The harvest function H_3 is defined similarly to the harvest function H_1 in terms of the harvest rate, but the harvest period does not have the same pattern as the harvest function H_1. As for the harvest function H_4, we define it similarly to the harvest function H_1 in terms of the harvest period. The difference between them is that the harvest rate for H_4 is not just a constant, but it depends on time. Tilapia population will be harvested with the seasonal rate of $156100\left(1+0.1\cos\left(\frac{2\pi t}{12}\right)\right)$ fish per month which resembles to the periodic carrying capacity $K(t)$ defined in (1). The harvest function H_5 is defined similarly to the harvest function H_2 in terms of the harvest period. However, the harvest rate is given by $78050\left(1+0.1\cos\left(\frac{2\pi t}{12}\right)\right)$ fish per month which equals to the half of the harvest rate for the harvest function H_4. As for the last harvest function H_6, we define it similarly to the harvest function H_3 in terms of the harvest period, but the harvest rate for the harvest function H_6 is slightly different from the harvest function H_3. The harvest rate for the harvest function H_6 resembles to the periodic carrying capacity $K(t)$ defined in (1). On month $0-2$ and $8-10$, the harvest rate is given by 171710 fish per month, which equals to the maximum of $156100\left(1+0.1\cos\left(\frac{2\pi t}{12}\right)\right)$. On month $4-6$, the harvest rate is given by 140490 fish per month, which equals to the minimum of $156100\left(1+0.1\cos\left(\frac{2\pi t}{12}\right)\right)$. Note that the harvest functions H_2 and H_5 are suitable for many farms in Thailand up to a scalar multiplication depending on the size of those farms.

To see the effect of each harvest function to the ODE (2), we use the Euler method to find the numerical solutions of ODE (2) with these 6 harvest functions. The numerical solutions are plotted in Figure 3.1 using parameters $r = 0.8$, $\varepsilon = 0.1$, $K_0 = 780500$ and $T_0 = 12$ with 3 initial conditions: 200000, 300000 and 900000. Notice that the number of tilapia population becomes zero for the yellow dotted line in Figures 3.1(a), 3.1(d) and 3.1(f).

3.2 Diffusion Term

From the diffusion term, the second term of the right-hand-side of SDE with jumps (4), if $\kappa = 0$, then the diffusion term will not depend on tilapia population, if $\kappa = 1$, then the diffusion term will depend on tilapia population linearly. However, if $\kappa > 1$, then the tilapia population will be unstable. Thus, κ can indicate how



(a) H_1 (b) H_2

(c) H_3 (d) H_4

(e) H_5 (f) H_6

Figure 3.1: Numerical solutions of ODE (2) with different harvest functions

much the noise change according to the tilapia population. To make the diffusion term more general, we assume that $\kappa \in [0, 1]$. ζ is the parameter indicating how much the noise oscillates according to the tilapia population. We study the proper values of $\zeta > 0$ and κ for the SDE with jumps in chapter 4.

3.3 Jump Terms

Jump terms, the summation term of the right-hand-side of SDE with jumps (4), describe the epidemic occurring with tilapia population based on 4 diseases, Columnaris, Epitheliocystis, Red egg, and Streptococcus. In our model, we let $J_t^{(i)}$ be the inhomogeneous compound Poisson process with intensity $\lambda_i(t)$ and jump size distribution $D^{(i)}$, which is a beta distribution or a logit-normal distribution, where $i = 1, ..., 4$ represent Columnaris, Epitheliocystis, Red egg, and Streptococcus, respectively. We describe all diseases and their corresponding parameters in subsections 3.3.1 - 3.3.4 and explain the meaning of the parameters η_i in subsection 3.3.5.

3.3.1 Columnaris

Chitmanat [14] proposed the fact that the infected tilapia from this disease will have pale body, slime, corrosion on fin and gill, and yellow spots in the wound. The cause of infection is stress from transportation. Thus, the tilapia population can be infected anytime where the chances of infected are different depending on the season. With this reason, AQUADAPT [15] proposed the level of risk of Columnaris in each month, and from [15], we propose the intensity function in time of Columnaris as described by

$$\lambda_1(t) = \begin{cases} 0.1 + \dfrac{0.4}{1 + 48e^{-16+8|t+0.5|}}, & \text{if} \quad t \in (0, 2] \\ 0.1, & \text{if} \quad t \in (2, 2.5] \\ 0.1 + \dfrac{0.2}{1 + 48e^{-24+8|t-6|}}, & \text{if} \quad t \in (2.5, 9.5] \\ 0.1, & \text{if} \quad t \in (9.5, 10] \\ 0.1 + \dfrac{0.4}{1 + 48e^{-16+8|t-12.5|}}, & \text{if} \quad t \in (10, 12], \end{cases}$$

and

$$\lambda_1(t + 12) = \lambda_1(t),$$

for $t \geq 0$. Dong et al. [16] proposed that the mortality of infected tilapia from this disease is approximately $10-70\%$, we propose that the severity of this disease can be modeled by the beta distribution,

$$D_1 \sim Beta(9, 15).$$

The intensity function and the severity function of Columnaris are shown as Figures 3.2(a) and 3.2(b), respectively.

3.3.2 Epitheliocystis

Somridhivej et al. [17] proposed the fact that the infected tilapia from this disease will have cysts in the epithelium cells of the gill that will cause the cells at the tip of the gill to enlarge and look like cysts. This disease can occur all year round. Thus, we propose that the intensity function in time of Epitheliocystis is described by

$$\lambda_2(t) = 0.5,$$

for $t \geq 0$. The mortality of infected tilapia from this disease is approximately $4 - 10\%$. Thus, we propose that the severity of this disease can be modeled by the logit-normal distribution,

$$D_2 \sim P(\mathcal{N}(-2.67, 0.0196)).$$

The intensity function and the severity function of Epitheliocystis are shown as Figures 3.2(c) and 3.2(d), respectively.

3.3.3 Red Egg

Senapin et al. [18] proposed the fact that the eggs of infected tilapia from this disease will change colors from normal yellowish to reddish and eventually fail to hatch. This disease spreads in the winter. Thus, we propose that the intensity function in time of Red egg is described by

$$\lambda_3(t) = \begin{cases} 0.02, & \text{if} \quad t \in (0,6] \\ 0.1 + \dfrac{0.48}{1 + 48e^{-16+8|t-8.5|}}, & \text{if} \quad t \in (6,11] \\ 0.02, & \text{if} \quad t \in (11,12], \end{cases}$$

and

$$\lambda_3(t+12) = \lambda_3(t),$$

for $t \geq 0$. The mortality of infected tilapia from this disease is approximately $10 - 50\%$. Thus, we propose that the severity of this disease can be modeled by the beta distribution,

$$D_3 \sim Beta(14, 36).$$

The intensity function and the severity function of Red Egg are shown as Figures 3.2(e) and 3.2(f), respectively.

3.3.4 Streptococcus

Chitmanat [14] proposed the fact that the infected tilapia from this disease will have bulging white eyes, swollen intestine, and cannot swim. The causes of infection are temperature fluctuation, trauma, and poor water quality. Thus, the tilapia population can be infected anytime where the chances of infection are different depending on the season. With this reason, AQUADAPT [15] proposed the level of risk of Streptococcus in each month, and from [15], we propose the intensity function in time of Streptococcus as described by

$$\lambda_4(t) = \begin{cases} 0.1 + \dfrac{0.4}{1 + 48e^{-16+8|t+0.5|}}, & \text{if} \quad t \in (0, 2] \\ 0.1, & \text{if} \quad t \in (2, 2.5] \\ 0.1 + \dfrac{0.2}{1 + 48e^{-24+8|t-6|}}, & \text{if} \quad t \in (2.5, 9.5] \\ 0.1, & \text{if} \quad t \in (9.5, 10] \\ 0.1 + \dfrac{0.4}{1 + 48e^{-16+8|t-12.5|}}, & \text{if} \quad t \in (10, 12], \end{cases}$$

and

$$\lambda_4(t + 12) = \lambda_4(t),$$

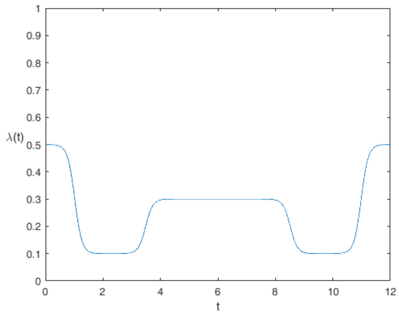
for $t \geq 0$. Assis et al. [19] proposed that the mortality of infected tilapia from this disease is approximately up to 90%. Thus, we propose that the severity of this disease can be modeled by the beta distribution,

$$D_4 \sim Beta(22, 10).$$

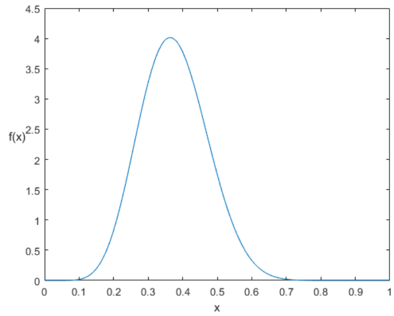
The intensity function and the severity function of Streptococcus are shown as Figures 3.2(g) and 3.2(h), respectively.

3.3.5 Immunity

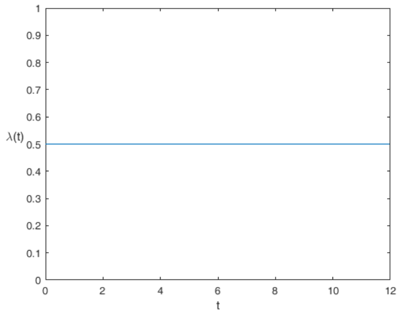
From the jump terms in SDE with jumps 4, we describe the meaning of η_i for $i = 1, ..., 4$ as follows. Assume that the tilapia farm has the number of tilapia population equal to X_t fish. If the tilapia farm wants to make parameter η_i equal to some constant $C \in [0, 1]$, the tilapia farm should maintain the level of immuned tilapia to $X_t - X_t^C$ fish. Thus, this amount of tilapia will not be dead from the i^{th} disease, but the others X_t^C tilapia will not have the immunity and can be dead from the i^{th} disease. We define the parameter η_i to be 1, if the i^{th} disease is not controlled by the tilapia farm. In this case tilapia population in the farm will have low immunity. If the i^{th} disease is controlled by the tilapia farm, tilapia population in the farm will have high immunity. In this case, η_i is less than 1. If the tilapia population have high immunity and do not die from the i^{th} disease, we will define the parameter η_i to be 0. Hence, $\eta_i \in [0, 1]$ represents how much the tilapia population have immunity against the i^{th} disease. In this study, we compare the trends of SDE with jumps (4) when we change the parameter η_i of some diseases, which will be described in chapter 4.



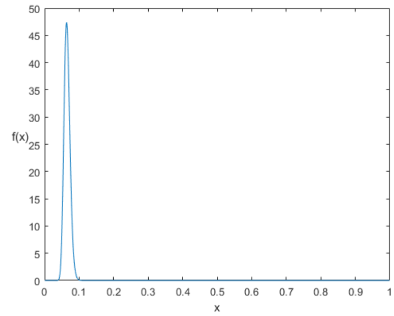
(a) Intensity function of Columnaris



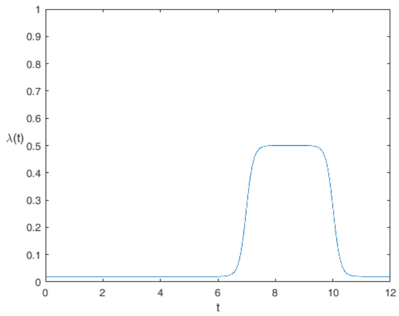
(b) PDF of the severity of Columnaris



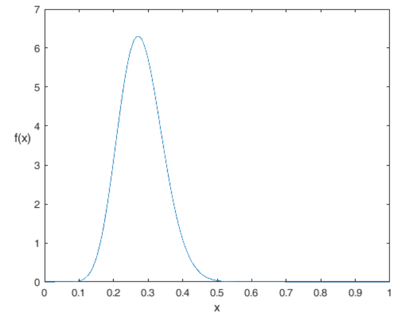
(c) Intensity function of Epitheliocystis



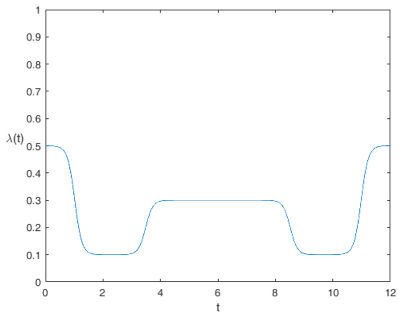
(d) PDF of the severity of Epitheliocystis



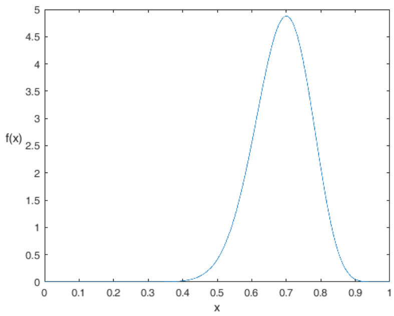
(e) Intensity function of Red egg



(f) PDF of the severity of Red egg



(g) Intensity function of Streptococcus



(h) PDF of the severity of Streptococcus

Figure 3.2: Intensity functions and PDFs of the severity of all diseases

CHAPTER IV

EXPERIMENTS AND RESULTS

In this chapter, we simulate the SDE without jumps (3), for tilapia population without disease occurring, and the SDE with jumps (4), for tilapia population with disease occurring, using the Euler-Maruyama and jump-adapted Euler methods, respectively. We perform the simulation 10000 sample paths over the time domain $[0, T]$ by showing the first 10 sample paths together with the average of all 10000 sample paths. In this work, we observe the simulation results for five years which should be enough to see the long-time behavior of the tilapia population. Thus, we define the maximum time domain $T = 60$. If the initial number of tilapia population is too low, the number of tilapia population may rapidly and easily become zero. Thus, we define the initial tilapia population $X_0 = 900000$. We define the proportion of extreme varying for the carrying capacity $\varepsilon = 0.1$, because from [3], $\varepsilon = 0.1$ is much less than 1. We want to discretize the time domain $[0, T]$ into 6000 equal pieces which is not too high or too low to observe the simulation results. Thus we define $\Delta t = 0.01$.

4.1 Suitable Diffusion Term of the SDE for Tilapia Population

As for the diffusion term in SDE (3), we would like to determine values of ζ and κ which bring about moderate noises. This means that the noises from the diffusion term are not too low or too high for observing the trend of SDE. If the noises are too low, environmental factors seems to disappear as if we had only the original ODE (2). If the noises are too high, the trend of the ODE (2) may no longer remain. Therefore, we simulate the SDE (3) by using $\zeta = 3, 30, 70$ and

$\kappa = 0.4, 0.5, 0.6$ to find proper values of ζ and κ. Here, we use H_1 as the common harvest function in our simulation. The results are shown in Figure 4.1 and they suggest that proper values of ζ and κ are $\zeta = 30$ and $\kappa = 0.5$ which correspond to Figure 4.1(e).

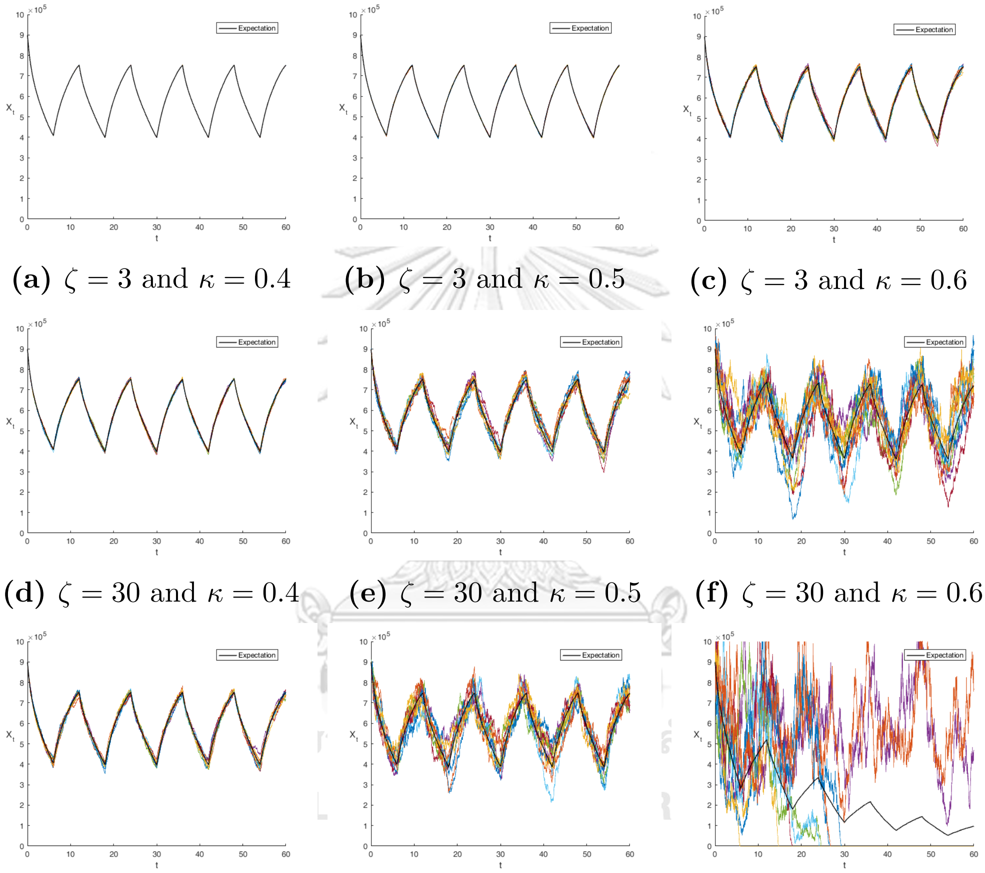
To see a mathematical criterion to choose the proper values of ζ and κ, we use the concept of the total variation for a sample path of an SDE. The numerical approximation of the total variation for a sample path of SDE (3) is given by $\sum_{n=1}^{N} |x_n - x_{n-1}|$. Table 4.1 shows the average of these approximation values of 10 sample paths with 9 different values of ζ and κ: $\zeta = 3, 30, 70$ and $\kappa = 0.4, 0.5, 0.6$. Since, the numerical approximation of the total variation for a sample path of SDE (3) with $\zeta = 30$ and $\kappa = 0.5$ is 11.411000 million fish which is not too low or too high, we suggest that proper values of ζ and κ are $\zeta = 30$ and $\kappa = 0.5$.

		κ		
		0.4	0.5	0.6
ζ	3	3.666500	3.698900	5.283200
	30	4.416500	11.411000	40.032000
	70	7.511400	25.574000	40.074000

Table 4.1: The numerical approximation of the total variation for a sample path of SDE (3) with $\zeta = 3, 30, 70$ and $\kappa = 0.4, 0.5, 0.6$, where the unit is in million fish.

4.2 Harvest Function of SDE of Tilapia Population

In this section, we use $\zeta = 30$ and $\kappa = 0.5$ obtained from section 4.1 to simulate the SDE (3) for tilapia population with the different harvest functions. The results are shown in Figure 4.2, and the MATLAB code for this simulation is given in APPENDIX A-F. From Figure 4.2, the results are similarly to the results from the ODE, Figure 3.1. Although the selected farm in Malaysia harvested only for half a year, the harvest period and the harvest rate of the other farms in Malaysia maybe not the same as the selected farm. Since most of the farms in Thailand always harvest the tilapia throughout the year, the harvest function H_2 and H_5



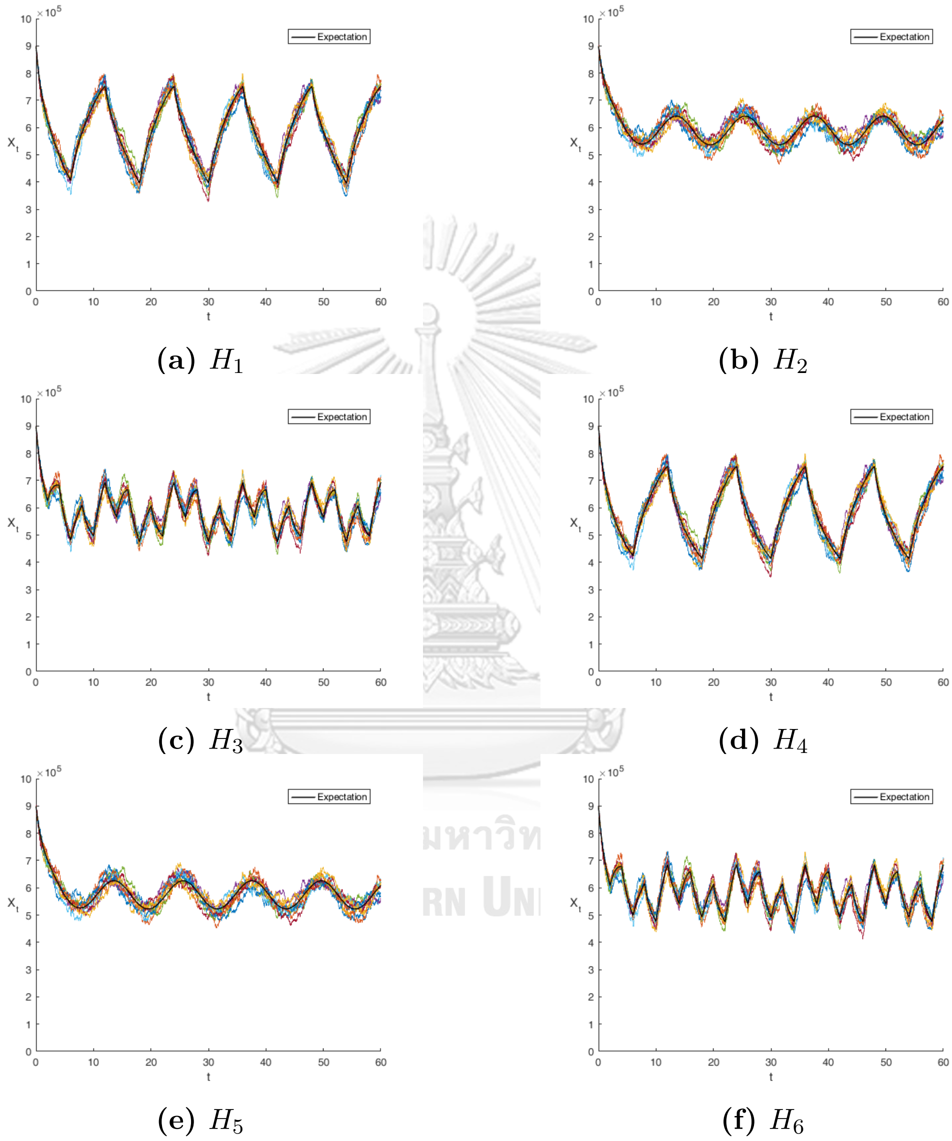
(a) $\zeta = 3$ and $\kappa = 0.4$ **(b)** $\zeta = 3$ and $\kappa = 0.5$ **(c)** $\zeta = 3$ and $\kappa = 0.6$

(d) $\zeta = 30$ and $\kappa = 0.4$ **(e)** $\zeta = 30$ and $\kappa = 0.5$ **(f)** $\zeta = 30$ and $\kappa = 0.6$

(g) $\zeta = 70$ and $\kappa = 0.4$ **(h)** $\zeta = 70$ and $\kappa = 0.5$ **(i)** $\zeta = 70$ and $\kappa = 0.6$

Figure 4.1: Numerical solutions of SDE (3) for tilapia population with harvest function H_1

should be the proper harvest functions for the most farms in Thailand. Note that the harvest rates of these harvest functions may differ from other tilapia farms in Thailand. However, we choose only the harvest function H_2 to observe the effect from the diseases in section 4.3.



(a) H_1 **(b)** H_2 **(c)** H_3 **(d)** H_4 **(e)** H_5 **(f)** H_6

Figure 4.2: Numerical solutions of SDE (3) with $\zeta = 30$, $\kappa = 0.5$ for different harvest functions

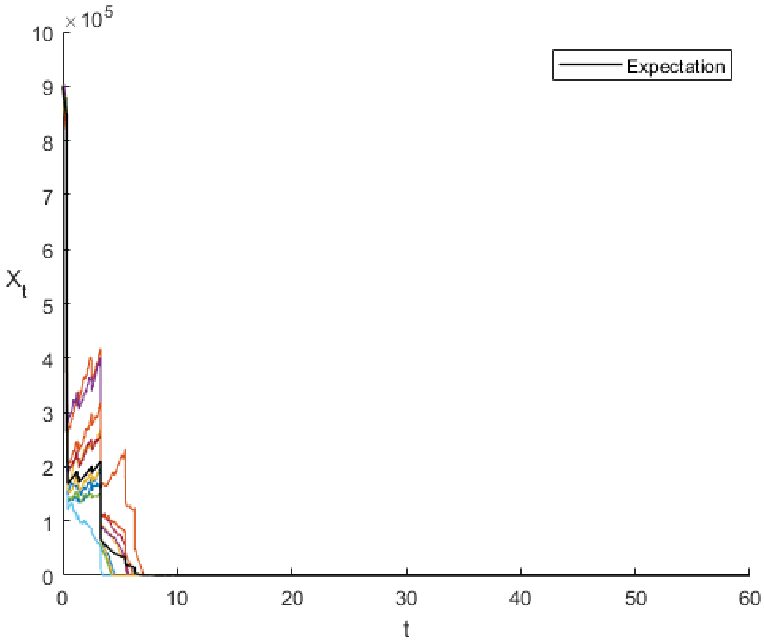


Figure 4.3: Numerical solutions of SDE (4) when all disease are not controlled

4.3 Jumps Effect from the SDE with Jumps of Tilapia Population

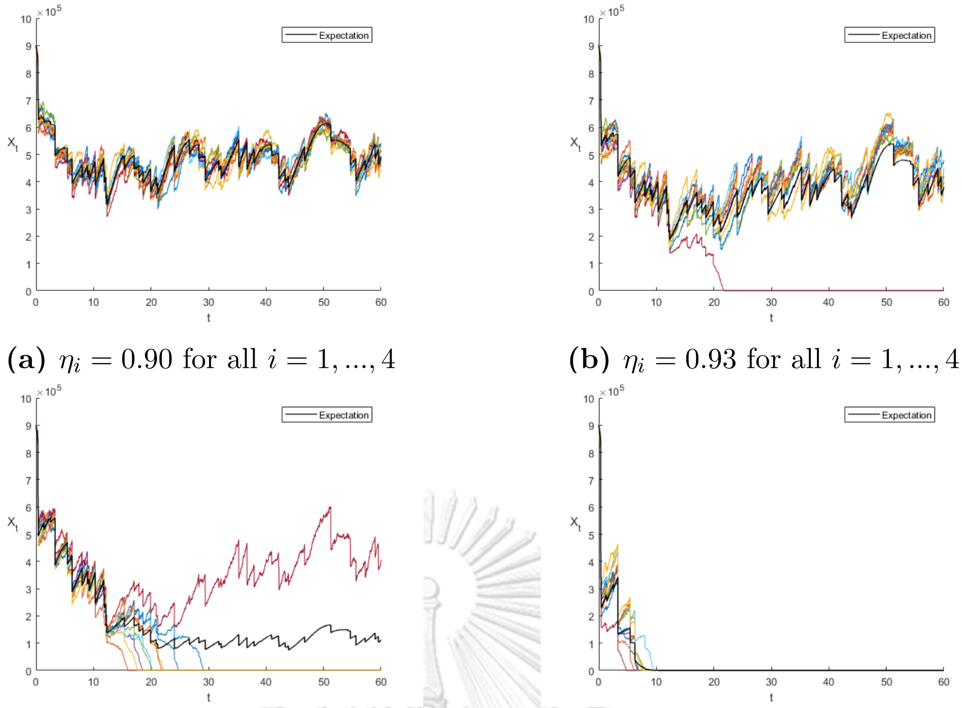
In this section, we divide the simulation into 4 cases. For the first case, we study when the tilapia farms have no control of diseases. For the second case, we study the controll of all disease via controlling of the parameter η_i. For the last 2 cases, we study the effect of disease classified by the time that diseases occur.

4.3.1 No Control of Diseases

In this case, the disease are not controlled, so $\eta_i = 1$ for $i = 1, ..., 4$. The results of SDE with jumps (4) in this subcase are shown in Figure 4.3. From Figure 4.3, the number of tilapia population becomes zero after some time due to the diseases.

4.3.2 Control all Diseases

In this case, we control all diseases using the same value of η_i for $i = 1, ..., 4$. We vary the value of all η_i to study how many tilapia population will become zero if all η_i are changed. After some preliminary experimental trials of the SDE with jumps (4), we choose all $\eta_i = 0.90, 0.91, 0.92, ..., 0.99$, because there are two specific values for all η_i that make the number of tilapia population become positive for all 10000 sample paths and make the number of tilapia population become zero exactly 1 sample path, respectively. These two specific values are between 0.90 and 0.99. Furthermore, we found that the number of tilapia population for all 10000 sample path are positive when all η_i less than 0.9. If tilapia farm maintain the level of immuned tilapia to $X_t - X_t^C$ fish where C is less than 0.9 the number of tilapia population will become positive. Thus, we consider the SDE with jumps (4) where all $\eta_i = 0.90, 0.91, 0.92, ..., 0.99$. Table 4.2 shows the number of sample paths of tilapia population becoming zero when all diseases are controlled with these values of η_i. We count the number of sample paths of tilapia population becoming zero when all diseases are controlled with all $\eta_i = 0.90, 0.91, 0.92, ..., 0.99$. If all η_i increase, the number of sample paths of tilapia population becoming zero will increase. Thus, if tilapia population in the tilapia farm have low immunity, the chance that the number of tilapia population become zero will be high. Furthermore, if all η_i are changed from 0.91 to 0.92, the number of sample paths of tilapia population becoming zero will change from 0 sample path to 11 sample paths. Thus, tilapia farm should maintain the level of immuned tilapia to $X_t - X_t^C$ fish where C is less than or equal to 0.91 to avoid that the number of tilapia population becomes zero. Note that if the initial number of tilapia population is too low, the number of sample paths of tilapia population becoming zero will close to 10000 easily and vice versa. The numerical solutions of (4) when $\eta_i = 0.90, 0.93, 0.94$, and 0.99 are shown in Figure 4.4. From Figure 4.4, if all η_i increase, then the number of sample paths whose numerical solution of (4) at $t = T$ is equal to zero will increase, and the expectation of the numerical solution of (4) for all sample paths will decrease. The MATLAB code for this simulation where $\eta_i = 0.90$ is given in APPENDIX G.



(a) $\eta_i = 0.90$ for all $i = 1, ..., 4$ **(b)** $\eta_i = 0.93$ for all $i = 1, ..., 4$

(c) $\eta_i = 0.94$ for all $i = 1, ..., 4$ **(d)** $\eta_i = 0.99$ for all $i = 1, ..., 4$

Figure 4.4: Numerical solutions of SDE (4) when all diseases are controlled with different value of η_i

4.3.3 Control Columnaris and Streptococcus

Since Columnaris, $i = 1$, and Streptococcus, $i = 4$, often occur in the same season, we control these 2 diseases by letting $\eta_1 = \eta_4 = 0.5$. Note that we can choose η_1 and η_4 around 0.3 to 0.6 which are not too low or too high to observe the simulation results where Columnaris and Streptococcus affect the tilapia population slightly, but we choose $\eta_1 = \eta_4 = 0.5$ in our simulation. Since Epitheliocystis and Red Egg are not controlled, we set $\eta_2 = \eta_3 = 1$. The numerical solutions of (4) in this case are shown in Figure 4.5. From Figure 4.5, since Columnaris and Streptococcus, which have high severity, are controlled, the number of tilapia population will not become zero.

η_i	The number of sample paths of tilapia population becoming zero	η_i	The number of sample paths of tilapia population becoming zero
0.90	0	0.95	9958
0.91	0	0.96	10000
0.92	11	0.97	10000
0.93	924	0.98	10000
0.94	7111	0.99	10000

Table 4.2: The number of sample paths of tilapia population becoming zero when all diseases are controlled with different value of η_i

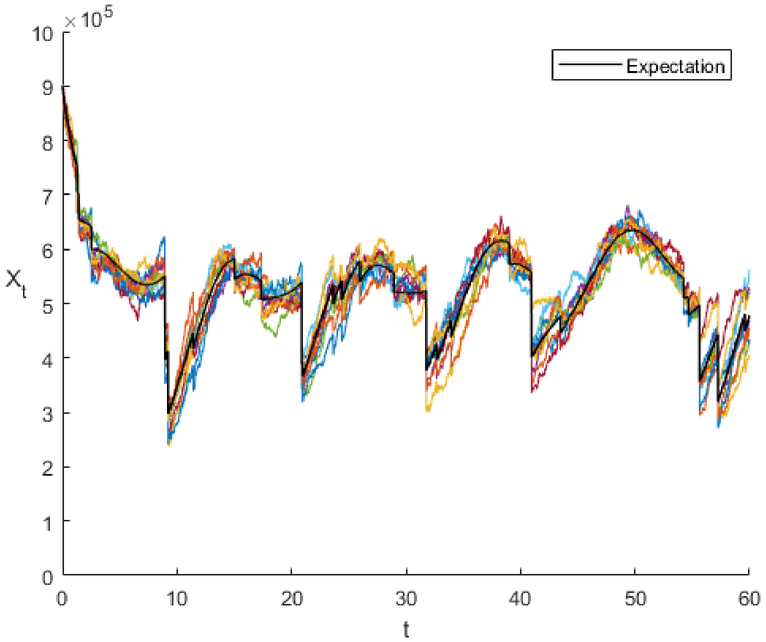


Figure 4.5: Numerical solutions of SDE (4) when Columnaris and Streptococcus are controlled

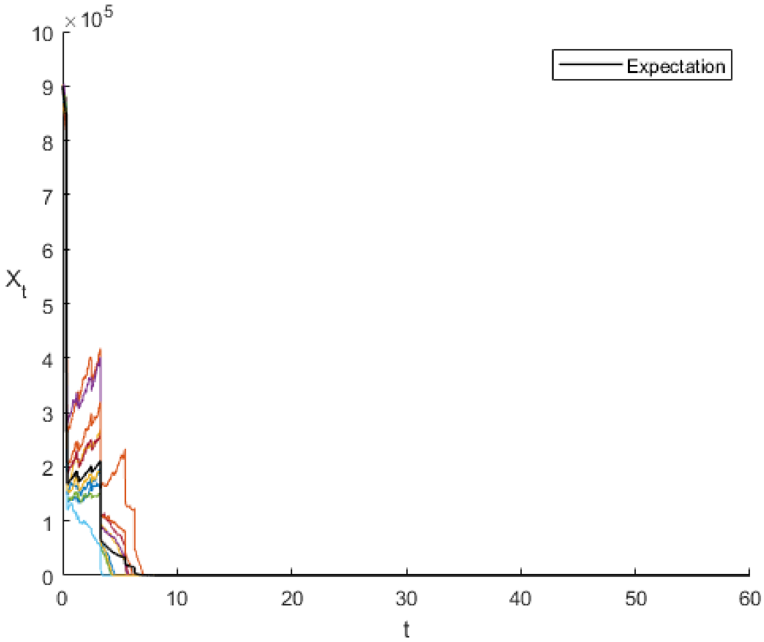


Figure 4.6: Numerical solutions of SDE (4) when Red Egg is controlled

4.3.4 Control Red Egg

In this case, we control Red Egg, $i = 3$, by letting $\eta_3 = 0.5$. Note that we can choose η_3 around 0.3 to 0.6 which are not too low or too high to observe the simulation results where Red Egg affect the tilapia population slightly, but we choose $\eta_3 = 0.5$ in our simulation. Since Columnaris, Epitheliocystis and Streptococcus are not controlled, $\eta_1 = \eta_2 = \eta_4 = 1$. The numerical solutions of (4) in this case is shown in Figure 4.6. From Figure 4.6, although Red egg, which have moderate severity, is controlled, the number of tilapia population becomes zero. This is because the other high severity deseases are not controlled.

CHAPTER V

CONCLUSIONS

In this work, we develop the ODE (2) into the SDE (3) by adding the diffusion term. Next, we develop the SDE (3) into the SDE with jumps (4) by adding the jump terms corresponding to fish diseases. Our model development process is explained in chapter 3 and 4.

In chapter 3, we describe in details of the harvest functions, the diffusion term and the jump terms for the SDE with jumps (4).

As for the harvest functions, we first define other 5 harvest functions that have different behaviors including the harvestment in Thailand which apart from the harvest function from [2] and observe the trends of (2) with different harvest functions by using the Euler method.

As for the the diffusion term, we use a Wiener process as a noise for the model. Furthermore, we assume $\kappa \in [0, 1]$ which can indicate how much the noise change according to the tilapia population and define $\zeta > 0$ which can how much the noise oscillates according to the tilapia population.

As for the the jump terms, we explain about the jump terms for (4) which describe the epidemic occurring with tilapia population based on 4 diseases, Columnaris, Epitheliocystis, Red egg, and Streptococcus. The chance that each disease occurs is consistent with the intensity of inhomogeneous compound Poisson process. The severity of each disease is consistent with the jump size distribution of inhomogeneous compound Poisson process which has a beta distribution or a logit-normal distribution. After that, we explain about the meaning of η_i in the jump terms for (4). If the tilapia farm wants to make parameter η_i equal to some constant $C \in [0, 1]$, the tilapia farm should maintain the level of immuned tilapia to $X_t - X_t^C$ fish. Thus, this amount of tilapia will not be dead from the i^{th} disease, but the other X_t^C tilapia will not have the immunity and can be dead from the i^{th} disease. Hence, $\eta_i \in [0, 1]$ represents how much the tilapia population have immunity against the i^{th} disease.

In chapter 4, we simulate the SDE (3), and the SDE with jumps (4), for tilapia population with disease occurring, using the Euler-Maruyama and jump-adapted Euler methods, respectively. We perform the simulation of 10000 sample paths over the time domain $[0, T]$ by showing the first 10 sample paths together with the average of all 10000 sample paths.

We first simulate the SDE (3) to obtain a proper diffusion term for the SDE with jumps (4) by using the Euler-Maruyama method. From the results, we found a the proper diffusion term has $\zeta = 30$ and $\kappa = 0.5$.

We simulate the SDE without jumps (3) with the proper diffusion term to observe the trends of (3) with different harvest functions by using the Euler-Maruyama method and choose the harvest function H_2 which is a proper harvest function for many farms in Thailand for the SDE with jumps (4). Finally, we divide the simulation into 4 situations to study the jump effect to the tilapia population and obtain the results of SDE with jumps (4) related to the diseases control.

For the first case, all diseases are not controlled by letting all $\eta_i = 1$. The results is that the number of tilapia population becomes zero after some time due to the diseases.

For the second case, we first perform some preliminary experimental trials of the SDE with jumps (4) to find an interval that has two specific values for all η_i that make the number of tilapia population become positive for all 10000 sample paths and make the number of tilapia population become zero exactly 1 sample path, respectively. These two specific values are between 0.90 and 0.99.

Furthermore, we found that the number of tilapia population for all 10000 sample path are positive when all η_i less than 0.9. Thus, we consider the SDE with jumps (4) where all $\eta_i = 0.90, 0.91, 0.92, ..., 0.99$. We find that if tilapia population have low immunity, the chance that the number of tilapia population becomes zero will be high. Moreover, tilapia farm should maintain the level of immuned tilapia to $X_t - X_t^C$ fish where C is less than or equal to 0.91 to avoid that the number of tilapia population becomes zero.

For the third case, Columnaris and Streptococcus are controlled by letting $\eta_1 = \eta_4 = 0.5$. However, Epitheliocystis and Red Egg are not controlled, i.e., $\eta_2 = \eta_3 = 1$. Since Columnaris and Streptococcus, which have high severity, are controlled, the number of tilapia population will not become zero.

For the last case, Red Egg are controlled by letting $\eta_3 = 0.5$. However, Columnaris, Epitheliocystis and Streptococcus are not controlled, i.e., $\eta_1 = \eta_2 = \eta_4 = 1$. Since the other high severity deseases are not controlled, the number of tilapia population eventually becomes zero. Since Epitheliocystis has low severity, we do not have the case that Epitheliocystis is controlled.

However, we did not consider to perform the parameter estimation in this work. If we had data and performed the parameter estimation for the SDE with jumps (4), we could obtain proper parameters and the model (4) should be more realistic.

In reality, there are many other factors apart from our work that affect tilapia population, so our model for tilapia population can be developed more in terms of sources of random noise to the model. However, we still believe that our model (4) for tilapia population is probably a good choice for studying the trend of the tilapia population in the future under the various factors such as epidemic in order to prepare and cope effectively with various conditions that may affect the tilapia population.

REFERENCES

[1] G. De Graaf, P. Dekker, B. Huisman, and J. Verreth, "Simulation of Nile tilapia culture in ponds, through individual-based modeling, using a population dynamic approach," *Aquaculture Research*, vol. 36, pp. 355–471, 2005.

[2] M. F. Laham, I. S. Krishnarajah, and J. M. Shariff, "Fish harvesting management strategies using logistic growth model," *Sains Malaysiana*, vol. 41, no. 2, pp. 171–177, 2012.

[3] M. A. Shah, "Stochastic logistic model for fish growth," *Open Journal of Statistics*, vol. 4, pp. 11–18, 2014.

[4] B. Øksendal, *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

[5] G. Casella and R. L. Berger, *Statistical inference*. Duxbury/Thomson Learning, 2001.

[6] R. Durrett, *Probability: theory and examples*, vol. 49. Cambridge university press, 2019.

[7] J. Atchison and S. M. Shen, "Logistic-normal distributions: Some properties and uses," *Biometrika*, vol. 67, no. 2, pp. 261–272, 1980.

[8] R. L. Kissell and J. Poserina, *Optimal Sports Math, Statistics, and Fantasy*. Academic Press, 2017.

[9] N. Privault, "Notes on stochastic finance," pp. 655–684, 2020.

[10] G. Franciszek, "Nonhomogeneous compound poisson process application to modeling of random processes related to accidents in the baltic sea waters

and ports," *Journal of Polish Safety and Reliability Association*, vol. 9, no. 3, pp. 21–29, 2018.

[11] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, vol. 53. Springer Science & Business Media, 2004.

[12] C. Jordan, "Sur la series de fourier," *CR Acad. Sci., Paris*, vol. 92, pp. 228–230, 1881.

[13] J. Von Neumann, "Various techniques used in connection with random digits," *Apply Mathematics Series*, vol. 12, no. 5, pp. 36–38, 1951.

[14] C. Chitmanat, "Tilapia Diseases," *Chiangmai Veterinary Journal*, vol. 11, pp. 75–86, 2013.

[15] AQUADAPT, "Policy and Practice Brief 7: Tilapia Diseases," *Unit for Social and Environmental Research*.

[16] H. Dong, B. LaFrentz, N. Pirarat, and C. Rodkhum, "Phenotypic characterization and genetic diversity of Flavobacterium columnare isolated from red tilapia, *Oreochromis* sp., in Thailand," *Journal of Fish Diseases*, vol. 38, no. 10, pp. 901–913, 2015.

[17] B. Somridhivej, P. Taveekijakarn, C. Me-anan, T. Somsiri, *et al.*, "Treatment of epitheliocystis in juvenile tilapia.," in *Proceedings of the 47th Kasetsart University Annual Conference, Kasetsart, 17-20 March, 2009. Subject: Fisheries*, pp. 1–8, Kasetsart University, 2009.

[18] S. Senapin, H. T. Dong, W. Meemetta, A. Siriphongphaew, W. Charoensapsri, W. Santimanawong, W. A. Turner, C. Rodkhum, B. Withyachumnarnkul, and R. Vanichviriyakit, "Hahella chejuensis is the etiological agent of a novel red egg disease in tilapia (*Oreochromis* spp.) hatcheries in Thailand," *Aquaculture*, vol. 454, pp. 1–7, 2016.

[19] G. Assis, G. Tavares, F. Pereira, H. Figueiredo, and C. Leal, "Natural coinfection by *Streptococcus agalactiae* and *Francisella noatunensis* subsp. *orientalis* in farmed Nile tilapia (*Oreochromis niloticus* L.)," *Journal of Fish Diseases*, vol. 40, no. 1, pp. 51–63, 2017.

APPENDICES

APPENDIX A : MATLAB code for finding the numerical solutions of the SDE (3) with harvest function 1

```
rng default;
% Define all parameters and all functions for the ODE term with out harvest function
r = 0.8;
K0 = 708500;
echilon = 0.1;
K = @(t) K0*(1+(echilon)*cos((2*pi*t)./12));
F = @(x,t) r*x*(1-(x/K(t)));
% Define all parameters and the functions for the diffusion term
kappa = 0.5;
zeta = 30;
G = @(x) zeta*(x.^kappa);
% Define all normal times maximum time domain
T = 60;
% Time step for normal times
dt = 0.01;
t = 0:dt:T;
% Find all numerical solutions by using Euler-Maruyama method with harvest function 1
Path = 10000;
N = length(t)-1;
x_sim = zeros(Path,N+1);
x_sim(1:Path,1) = 900000;
H0 = 156100;
H = @(t) H0;
for i=1:Path
for j=1:length(t)-1
dw = randn();
if mod(t(j),12)>0 && mod(t(j),12)<=6
x_sim(i,j+1) = x_sim(i,j)+...
```

```
(F(x_sim(i,j),t(j))-H(t(j)))*(dt)+...
G(x_sim(i,j))*sqrt(dt)*dw;
else
x_sim(i,j+1) = x_sim(i,j)+...
(F(x_sim(i,j),t(j)))*(dt)+...
G(x_sim(i,j))*sqrt(dt)*dw;
end
if x_sim(i,j+1) <= 0
x_sim(i,j+1) = 0;
break;
end
end
end
```

APPENDIX B : MATLAB code for finding the numerical solutions of the SDE (3) with harvest function 2

```
rng default;
% Define all parameters and all functions for the ODE term with out harvest function
r = 0.8;
K0 = 708500;
echilon = 0.1;
K = @(t) K0*(1+(echilon)*cos((2*pi*t)./12));
F = @(x,t) r*x*(1-(x/K(t)));
% Define all parameters and the functions for the diffusion term
kappa = 0.5;
zeta = 30;
G = @(x) zeta*(x.^kappa);
% Define all normal times
% maximum time domain
T = 60;
% Time step for normal times
dt = 0.01;
t = 0:dt:T;
% Find all numerical solutions by using Euler-Maruyama method with harvest function 2
Path = 10000;
N = length(t)-1;
x_sim = zeros(Path,N+1);
x_sim(1:Path,1) = 900000;
H0 = 156100;
H = @(t) H0/2;
for i=1:Path
for j=1:length(t)-1
dw = randn();
```

```
            x_sim(i,j+1) = x_sim(i,j)+...
            (F(x_sim(i,j),t(j))-H(t(j)))*(dt)+...
            G(x_sim(i,j))*sqrt(dt)*dw;
            if x_sim(i,j+1) <= 0
            x_sim(i,j+1) = 0;
            break;
            end
            end
            end
```

APPENDIX C : MATLAB code for finding the numerical solutions of the SDE (3) with harvest function 3

```
rng default;
% Define all parameters and all functions for the ODE term
    with out harvest function
r = 0.8;
K0 = 708500;
echilon = 0.1;
K = @(t) K0*(1+(echilon)*cos((2*pi*t)./12));
F = @(x,t) r*x*(1-(x/K(t)));
% Define all parameters and the functions for the diffusion
    term
kappa = 0.5;
zeta = 30;
G = @(x) zeta*(x.^kappa);
% Define all normal times
% maximum time domain
T = 60;
% Time step for normal times
dt = 0.01;
t = 0:dt:T;
% Find all numerical solutions by using Euler-Maruyama method
     with harvest function 3
Path = 10000;
N = length(t)-1;
x_sim = zeros(Path,N+1);
x_sim(1:Path,1) = 900000;
H0 = 156100;
H = @(t) H0;
for i=1:Path
for j=1:length(t)-1
dw = randn();
```

```
if mod(t(j),12)>0 && mod(t(j),12)<=2
x_sim(i,j+1) = x_sim(i,j)+...
(F(x_sim(i,j),t(j))-H(t(j)))*(dt)+...
G(x_sim(i,j))*sqrt(dt)*dw;
elseif mod(t(j),12)>2 && mod(t(j),12)<=4
x_sim(i,j+1) = x_sim(i,j)+...
(F(x_sim(i,j),t(j)))*(dt)+...
G(x_sim(i,j))*sqrt(dt)*dw;
elseif mod(t(j),12)>4 && mod(t(j),12)<=6
x_sim(i,j+1) = x_sim(i,j)+...
(F(x_sim(i,j),t(j))-H(t(j)))*(dt)+...
G(x_sim(i,j))*sqrt(dt)*dw;
       elseif mod(t(j),12)>6 && mod(t(j),12)<=8
x_sim(i,j+1) = x_sim(i,j)+...
(F(x_sim(i,j),t(j)))*(dt)+...
G(x_sim(i,j))*sqrt(dt)*dw;
elseif mod(t(j),12)>8 && mod(t(j),12)<=10
x_sim(i,j+1) = x_sim(i,j)+...
(F(x_sim(i,j),t(j))-H(t(j)))*(dt)+...
G(x_sim(i,j))*sqrt(dt)*dw;
else
x_sim(i,j+1) = x_sim(i,j)+...
(F(x_sim(i,j),t(j)))*(dt)+...
G(x_sim(i,j))*sqrt(dt)*dw;
end
if x_sim(i,j+1) <= 0
x_sim(i,j+1) = 0;
break;
end
end
end
```

APPENDIX D : MATLAB code for finding the numerical solutions of the SDE (3) with harvest function 4

```
rng default;
% Define all parameters and all functions for the ODE term
    with out harvest function
r = 0.8;
K0 = 708500;
echilon = 0.1;
K = @(t) K0*(1+(echilon)*cos((2*pi*t)./12));
F = @(x,t) r*x*(1-(x/K(t)));
% Define all parameters and the functions for the diffusion
    term
kappa = 0.5;
zeta = 30;
G = @(x) zeta*(x.^kappa);
% Define all normal times
% maximum time domain
T = 60;
% Time step for normal times
dt = 0.01;
t = 0:dt:T;
% Find all numerical solutions by using Euler-Maruyama method
     with harvest function 4
Path = 10000;
N = length(t)-1;
x_sim = zeros(Path,N+1);
x_sim(1:Path,1) = 900000;
H0 = 156100;
H = @(t) H0*(1+(echilon)*cos((2*pi*t)./12));
for i=1:Path
for j=1:length(t)-1
dw = randn();
```

```
if mod(t(j),12)>0 && mod(t(j),12)<=6
x_sim(i,j+1) = x_sim(i,j)+...
(F(x_sim(i,j),t(j))-H(t(j)))*(dt)+...
G(x_sim(i,j))*sqrt(dt)*dw;
else
x_sim(i,j+1) = x_sim(i,j)+...
(F(x_sim(i,j),t(j)))*(dt)+...
G(x_sim(i,j))*sqrt(dt)*dw;
end
if x_sim(i,j+1) <= 0
x_sim(i,j+1) = 0;
break;
end
end
end
```

APPENDIX E : MATLAB code for finding the numerical solutions of the SDE (3) with harvest function 5

```
rng default;
% Define all parameters and all functions for the ODE term with out harvest function
r = 0.8;
K0 = 708500;
echilon = 0.1;
K = @(t) K0*(1+(echilon)*cos((2*pi*t)./12));
F = @(x,t) r*x*(1-(x/K(t)));
% Define all parameters and the functions for the diffusion term
kappa = 0.5;
zeta = 30;
G = @(x) zeta*(x.^kappa);
% Define all normal times
% maximum time domain
T = 60;
% Time step for normal times
dt = 0.01;
t = 0:dt:T;
% Find all numerical solutions by using Euler-Maruyama method with harvest function 5
Path = 10000;
N = length(t)-1;
x_sim = zeros(Path,N+1);
x_sim(1:Path,1) = 900000;
H0 = 156100;
H = @(t) (H0/2)*(1+(echilon)*cos((2*pi*t)./12));
for i=1:Path
for j=1:length(t)-1
dw = randn();
```

```
        x_sim(i,j+1) = x_sim(i,j)+...
        (F(x_sim(i,j),t(j))-H(t(j)))*(dt)+...
        G(x_sim(i,j))*sqrt(dt)*dw;
        if x_sim(i,j+1) <= 0
        x_sim(i,j+1) = 0;
        break;
        end
        end
        end
```

APPENDIX F : MATLAB code for finding the numerical solutions of the SDE (3) with harvest function 6

```
rng default;
% Define all parameters and all functions for the ODE term with out harvest function
clearvars;
rng default;
r = 0.8;
K0 = 708500;
echilon = 0.1;
K = @(t) K0*(1+(echilon)*cos((2*pi*t)./12));
F = @(x,t) r*x*(1-(x/K(t)));
% Define all parameters and the functions for the diffusion term
kappa = 0.5;
zeta = 30;
G = @(x) zeta*(x.^kappa);
% Define all normal times
% maximum time domain
T = 60;
% Time step for normal times
dt = 0.01;
t = 0:dt:T;
% Find all numerical solutions by using Euler-Maruyama method with harvest function 6
Path = 10000;
N = length(t)-1;
x_sim = zeros(Path,N+1);
x_sim(1:Path,1) = 900000;
H0 = 156100;
H1 = @(t) H0*(1+echilon);
H2 = @(t) H0*(1-echilon);
```

```
for i=1:Path
for j=1:length(t)-1
dw = randn();
if mod(t(j),12)>0 && mod(t(j),12)<=2
x_sim(i,j+1) = x_sim(i,j)+...
(F(x_sim(i,j),t(j))-H1(t(j)))*(dt)+...
G(x_sim(i,j))*sqrt(dt)*dw;
elseif mod(t(j),12)>2 && mod(t(j),12)<=4
x_sim(i,j+1) = x_sim(i,j)+...
(F(x_sim(i,j),t(j)))*(dt)+...
G(x_sim(i,j))*sqrt(dt)*dw;
elseif mod(t(j),12)>4 && mod(t(j),12)<=6
x_sim(i,j+1) = x_sim(i,j)+...
(F(x_sim(i,j),t(j))-H2(t(j)))*(dt)+...
G(x_sim(i,j))*sqrt(dt)*dw;
       elseif mod(t(j),12)>6 && mod(t(j),12)<=8
x_sim(i,j+1) = x_sim(i,j)+...
(F(x_sim(i,j),t(j)))*(dt)+...
G(x_sim(i,j))*sqrt(dt)*dw;
elseif mod(t(j),12)>8 && mod(t(j),12)<=10
x_sim(i,j+1) = x_sim(i,j)+...
(F(x_sim(i,j),t(j))-H1(t(j)))*(dt)+...
G(x_sim(i,j))*sqrt(dt)*dw;
else
x_sim(i,j+1) = x_sim(i,j)+...
(F(x_sim(i,j),t(j)))*(dt)+...
G(x_sim(i,j))*sqrt(dt)*dw;
end
if x_sim(i,j+1) <= 0
x_sim(i,j+1) = 0;
break;
end
end
```

```
        end
```

APPENDIX G : MATLAB code for finding the numerical solutions of the SDE with jumps (4) with harvest function 2

```
rng default;
% Define all parameters and all functions for the ODE term
    with out harvest function
r = 0.8;
K0 = 708500;
echilon = 0.1;
K = @(t) K0*(1+(echilon)*cos((2*pi*t)./12));
F = @(x,t) r*x*(1-(x/K(t)));
% Define all parameters and the functions for the diffusion
    term
kappa = 0.5;
zeta = 30;
G = @(x) zeta*(x.^kappa);
% Define all parameters and the functions for the jump terms
eta1 = 0.9;
eta2 = 0.9;
eta3 = 0.9;
eta4 = 0.9;
V1 = @(x) -(x.^eta1);
V2 = @(x) -(x.^eta2);
V3 = @(x) -(x.^eta3);
V4 = @(x) -(x.^eta4);
% Define all times and find jump times that diseases occur
% Define all normal times
% maximum time domain
T = 60;
% Time step for normal times
dt = 0.01;
t_normal = 0:dt:T;
% Find all jump times that Columnaris occur
```

```
% Define the bounded value for inhomogeneous compound poisson process to find all jump times that Columnaris occur
lambdat_max1 = 0.5;
% Define the intensity function to find the jump times that Columnaris
% occur
f11 = @(x) (0.4./(1+48*exp(-16+8*abs(mod(x,12)+0.5)))+0.1);
f12 = @(x) 0.1;
f13 = @(x) (0.2./(1+48*exp(-24+8*abs(mod(x,12)-6)))+0.1);
f14 = @(x) 0.1;
f15 = @(x) (0.4./(1+48*exp(-16+8*abs(mod(x,12)-12.5)))+0.1);
% Find all jump times that Columnaris occur by inhomogeneous compound poisson process
tjump1 = 0;
tjump_in1 = 0;
while tjump_in1(end) <= T
tjump1 = [tjump1 , tjump1(end) + exprnd(1/lambdat_max1)];
if mod(tjump1(end),12) < 2
if rand() <= f11(tjump1(end))/lambdat_max1
tjump_in1 = [tjump_in1 , tjump1(end)];
end
elseif mod(tjump1(end),12) < 2.5
if rand() <= f12(tjump1(end))/lambdat_max1
tjump_in1 = [tjump_in1 , tjump1(end)];
end
elseif mod(tjump1(end),12) < 9.5
if rand() <= f13(tjump1(end))/lambdat_max1
tjump_in1 = [tjump_in1 , tjump1(end)];
end
elseif mod(tjump1(end),12) < 10
if rand() <= f14(tjump1(end))/lambdat_max1
tjump_in1 = [tjump_in1 , tjump1(end)];
end
```

```
else
if rand() <= f15(tjump1(end))/lambdat_max1
tjump_in1 = [tjump_in1 , tjump1(end)];
end
end
end
tjump_in1(1)=[];
tjump_in1(end)=[];
% Define all parameters for the severity of Columnaris
alpha1 = 9;
beta1 = 15;
% Find all jump times that Ephitheliocystis occur
% Define the bounded value for inhomogeneous compound poisson process to find all jump times that Ephitheliocystis occur.
lambdat_max2 = 0.5;
% Define the intensity function to find the jump times that Ephitheliocystis occur
f2 = @(x) 0.5;
% Find all jump times that Ephitheliocystis occur by inhomogeneous compound poisson process
tjump2 = 0;
tjump_in2 = 0;
while tjump_in2(end) <= T
tjump2 = [tjump2 , tjump2(end) + exprnd(1/lambdat_max2)];
if rand() <= f2(tjump2(end))/lambdat_max2
tjump_in2 = [tjump_in2 , tjump2(end)];
end
end
tjump_in2(1)=[];
tjump_in2(end)=[];
% Define all parameters for the PDF function for the severity of Ephitheliocystis
```

```
sigma_Ephitheliocystis = 0.14;
mu_Ephitheliocystis = -2.67;
logit = @(x) log(x./(1-x));
fEphitheliocystis = @(x) (1./(sigma_Ephitheliocystis.*sqrt(2.*pi)))*...
(1./(x.*(1-x))).*exp(-((logit(x)-mu_Ephitheliocystis).^2)./...
(2.*sigma_Ephitheliocystis^2));
% Find all jump times that Red egg occur
% Define the bounded value for inhomogeneous compound poisson process to find all jump times that Red egg occur.
lambdat_max3 = 0.5;
% Define the intensity function to find the jump times that Red egg occur
f31 = @(x) (0.02);
f32 = @(x) (0.48./(1+48*exp(-16+8*abs(mod(x,12)-8.5)))+0.02);
f33 = @(x) (0.02);
% Find all jump times that Red egg occur by inhomogeneous compound poisson process
tjump3 = 0;
tjump_in3 = 0;
while tjump_in3(end) <= T
tjump3 = [tjump3 , tjump3(end) + exprnd(1/lambdat_max3)];
if mod(tjump3(end),12) < 6
if rand() <= f31(tjump3(end))/lambdat_max3
tjump_in3 = [tjump_in3 , tjump3(end)];
end
elseif mod(tjump3(end),12) < 11
if rand() <= f32(tjump3(end))/lambdat_max3
tjump_in3 = [tjump_in3 , tjump3(end)];
end
else
if rand() <= f33(tjump3(end))/lambdat_max3
```

```
tjump_in3 = [tjump_in3 , tjump3(end)];
end
end
end
tjump_in3(1)=[];
tjump_in3(end)=[];
% Define all parameters for the severity of Red egg
alpha3 = 14;
beta3 = 36;
% Find all jump times that Streptococcus occur
% Define the bounded value for inhomogeneous compound poisson process to find all jump times that Streptococcus occur
lambdat_max4 = 0.5;
% Define the intensity function to find the jump times that Streptococcus
% occur
f41 = @(x) (0.4./(1+48*exp(-16+8*abs(mod(x,12)+0.5)))+0.1);
f42 = @(x) 0.1;
f43 = @(x) (0.2./(1+48*exp(-24+8*abs(mod(x,12)-6)))+0.1);
f44 = @(x) 0.1;
f45 = @(x) (0.4./(1+48*exp(-16+8*abs(mod(x,12)-12.5)))+0.1);
% Find all jump times that Streptococcus occur by inhomogeneous compound poisson process
tjump4 = 0;
tjump_in4 = 0;
while tjump_in4(end) <= T
tjump4 = [tjump4 , tjump4(end) + exprnd(1/lambdat_max4)];
if mod(tjump4(end),12) < 2
if rand() <= f41(tjump4(end))/lambdat_max4
tjump_in4 = [tjump_in4 , tjump4(end)];
end
elseif mod(tjump4,12) < 2.5
if rand() <= f42(tjump4(end))/lambdat_max4
```

```
tjump_in4 = [tjump_in4 , tjump4(end)];
end
elseif mod(tjump4,12) < 9.5
if rand() <= f43(tjump4(end))/lambdat_max4
tjump_in4 = [tjump_in4 , tjump4(end)];
end
elseif mod(tjump4,12) < 10
if rand() <= f44(tjump4(end))/lambdat_max4
tjump_in4 = [tjump_in4 , tjump4(end)];
end
else
if rand() <= f45(tjump4(end))/lambdat_max4
tjump_in4 = [tjump_in4 , tjump4(end)];
end
end
end
tjump_in4(1)=[];
tjump_in4(end)=[];
% Define all parameters for the severity of Streptococcus
alpha4 = 22;
beta4 = 10;
% Combine all normal times and all jump times that disease
    occur
t = unique...
(sort([t_normal , tjump_in1 , tjump_in2 , tjump_in3 ,
    tjump_in4]));
% Find all numerical solutions by using Jump-adapted euler
    method with harvest function 2
Path = 10000;
N = length(t)-1;
x_sim = zeros(Path,N+1);
x_sim(1:Path,1) = 900000;
H0 = 156100;
```

```
H = @(t) H0/2;
for i=1:Path
for j=1:length(t)-1
k = 1;
cj1 = 0;
while k<=length(tjump_in1)
if t(j+1)==tjump_in1(k)
% Find severity of Columnaris
cj1 = cj1 + betarnd(alpha1,beta1);
end
k=k+1;
end
k = 1;
cj2 = 0;
while k<=length(tjump_in2)
if t(j+1)==tjump_in2(k)
% Find severity of Ephitheliocystis by Acceptance-rejection
u1 = 0.001 + (0.999-0.001)*rand();
u2 = 0.001 + (0.999-0.001)*rand();
while u2 > fEphitheliocystis(u1)/50
u1 = 0.001 + (0.999-0.001)*rand();
u2 = 0.001 + (0.999-0.001)*rand();
end
severity_of_Ephitheliocystis = u1;
cj2 = cj2 + severity_of_Ephitheliocystis;
end
k=k+1;
end
k = 1;
cj3 = 0;
while k<=length(tjump_in3)
if t(j+1)==tjump_in3(k)
% Find severity of Red egg
```

```
        cj3 = cj3 + betarnd(alpha3,beta3);
        end
        k=k+1;
        end
        k = 1;
        cj4 = 0;
        while k<=length(tjump_in4)
        if t(j+1)==tjump_in4(k)
        % Find severity of Streptococcus
        cj4 = cj4 + betarnd(alpha4,beta4);
        end
        k=k+1;
        end
        dw = randn();
        x_sim(i,j+1) = x_sim(i,j)+...
        (F(x_sim(i,j),t(j))-H(t(j)))*(t(j+1)-t(j))+...
        G(x_sim(i,j))*sqrt(t(j+1)-t(j))*dw;
        x_sim(i,j+1) = x_sim(i,j+1)+...
        V1(x_sim(i,j+1))*cj1+V2(x_sim(i,j+1))*cj2+...
        V3(x_sim(i,j+1))*cj3+V4(x_sim(i,j+1))*cj4;
        if x_sim(i,j+1) <= 0
        x_sim(i,j+1) = 0;
        break;
        end
        end
        end
```

APPENDIX H : MATLAB code for plotting a graph

```
hold on
plot(t,x_sim(1:10,:))
plot(t,mean(x_sim,1),'k','LineWidth', 1)
xlabel('t')
ylabel('X_{t} ','Rotation',0)
legend([plot(t,mean(x_sim,1),'k','LineWidth', 1)], 'Expectation')
xlim([0 T])
ylim([0 1000000])
hold off
```

BIOGRAPHY

Name Mr. Kanitin Sukchum

Date of Birth 4 August 1995

Place of Birth Bangkok, Thailand

Education B.Sc. (Mathematics),

Chulalongkorn University, 2017