People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of Saida Dr. Moulay Tahar
Faculty of Mathematics, Computer
Science and Telecommunication
Department of Mathematics

Thesis submitted for the Academic
Master's degree

Specialty: Stochastic Analysis, Process Statistics and Applications

Presented by

Chabane Djamila[1]

Supervised by

Dr. Mekkaoui Imen and **Dr. Kebiri Omar**

Theme:

Uncertainty-Aware Control of Epidemic Processes

Defended on June 18th, 2026 in front of the jury composed of:

Pr. Kandouci Abdeldjebbar	University of Saida Dr. Moulay Tahar	President
Dr. Mekkaoui Imen	University of Saida Dr. Moulay Tahar	Supervisor
Dr. Kebiri Omar	Brandenburg University of Technology	Co-supervisor
Dr. Bachir Cherif Khalida	University of Saida Dr. Moulay Tahar	Examiner

Academic Year: 2025/2026

[1]e-mail:djamila.chabane@stu.univ-saida.dz

Acknowledgments

In the name of Allah, the Most Gracious, the Most Merciful

All praise and thanks to Allah Almighty for giving me strength, perseverance, and knowledge to write this work.

I would like to express my sincere appreciation and gratitude to my supervisors, who have been the backbone of this research for with their continuous support and guidance. **Dr. Imene Mekkaoui** and **Dr. Omar Kebiri**, your constant encouragement, valuable advice and constructive criticism have greatly influenced this work. I am so grateful for their patience and dedication.

Special thanks to **Prof. Abdeldjabbar Kandouci** Director of the Laboratory of Stochastic Models, Statistics and Applications (LMSSA) at the University of Saida Dr Moulay Tahar for his support without interruption, his help without limits and his motivation without fail during this trip. His leadership and scientific insight have been an invaluable asset.

My deepest gratitude is to the distinguished members of the examination committee who kindly agreed to evaluate and appraise this dissertation:

Prof. Abdeldjabbar Kandouci
Dr.Khalida Bachir Cherif

I would like to thank all the respected professors and members of the Laboratory LMSSA at the University of Saida. Their collective support was stimulating and generous discussions.

Finally, I would like to thank everyone who helped me along the way during my five years at the university. Whether it be academic advice, administrative help or moral support, their contributions have not gone unnoticed. May Allah reward them in abundance.

Abstract

This thesis develops and analyzes a stochastic epidemic model incorporating Lévy jump processes to capture the discontinuous, which classical deterministic and Brownian motion driven models fail to represent.

The work begins by formulating a deterministic compartmental system of ordinary differential equations (ODEs), then progressively introduces continuous random fluctuations via a Wiener process, and finally adds discontinuous jumps through a Lévy process.

The core contribution lies in the optimal control problem, where the objective is to minimize the number of infected individuals while simultaneously reducing intervention costs. Two controls are considered: immigration control and preventive measures.

Using the stochastic maximum principle extended to Lévy processes, we derive necessary optimality conditions expressed through a Hamiltonian function and a coupled forward-backward system of stochastic differential equations (FBSDEs) for the state and adjoint variables.

Because analytical solutions are infeasible, we employ a numerical scheme based on the Least Squares Monte Carlo (LSMC) method combined with the Milstein method.

The simulation results are presented through graphs that demonstrate the effectiveness of the optimal control strategy. The findings confirm that incorporating Lévy jumps and solving the resulting optimal control problem yields robust, practical intervention policies for epidemic management.

Keywords: Epidemic model, Lévy process, optimal control, stochastic maximum principle, forward-backward SDEs, LSMC, Milstein scheme, immigration control, preventive measures.

List of Figures

List of Tables

List of Algorithms

Contents

Introduction

Effects of infectious diseases are not limited only to morbidity but have far-reaching effects on public health structures and financial stability as well. As evidenced by events in recent decades, insufficient containment of epidemics puts financial strain on the medical system, creating a chain of problems whose repercussions persist long after the initial crisis has passed [19]. The coronavirus epidemic was particularly revealing as a watershed event that highlighted the vulnerabilities in modern healthcare systems when managing new infections, prompting scientists from various fields to reconsider the basics of their knowledge about the spread of infections and the timing of necessary actions. In this regard, mathematics has become an invaluable tool for understanding infectious diseases rather than merely a language used to describe their effects.

The basic SIR (Susceptible-Infected-Recovered) model, which was formulated by Kermack and McKendrick (1927) [16], offered valuable insight into the pathogen propagation and control and the very foundation of mathematical epidemiology. The effectiveness of its use cannot be underestimated [9]. Nevertheless, such deterministic models rely on certain assumptions about the continuity, homogeneity, and predictability of the processes under investigation. In reality, outbreaks are rarely continuous. They can abruptly happen because of the emergence of a highly contagious new variant, an unannounced super-spreading event during an assembling, or a surprise introduction of a lockdown measure. All those events introduce unpredictable non-Gaussian discontinuous behavior which cannot be captured with the help of a differential equation. Therefore, even if useful in describing the average behavior of the epidemic, deterministic models cannot describe events which can shape the very course of the epidemic.

One way to add an element of randomness into an epidemic process is the use of stochastic differential equations (SDEs). The most common approach is adding a Wiener process to describe random diffusion in the equation of transmission or recovery [20]. Still, as many authors point out, the use of white noise is not enough since it cannot describe a large sudden jump in the case number. To overcome this problem, we should add the Lévy process to our equation. Indeed, this critical gap

in modeling capabilities is what the Lévy process helps fill in perfectly. Specifically, a Lévy process is a stochastic process with stationary and independent increments and allows describing both continuous (Brownian) and discontinuous behavior in the process studied. Hence, the use of such a process allows capturing the dynamics of the system under consideration very well. In particular, this process features Poisson jumps. It describes large-scale but sudden changes in the process. For example, positive jumps can correspond to a sudden increase in the number of infections due to a new variant, while negative jumps represent abrupt decreases due to policy interventions [25].

With Lévy noise, the problem of optimal control becomes mathematically challenging. To formulate this problem, we rely on the Hamiltonian function, which combines the cost functional, the system dynamics, and the adjoint variables. Applying the necessary condition for optimality in the presence of jumps leads to a system of forward-backward stochastic differential equations (FBSDEs): the state equations evolve forward in time, while the adjoint equations evolve backward, both affected by the jump component. Analytical solutions are generally impossible. Therefore, we turn to numerical methods.

This thesis contains three chapters

- In the first chapter, we introduce the necessary for the entire thesis by starting with a classical deterministic epidemic models (SIS, SIR, SIER, SIRS) expressed as a system of ordinary differential equations (ODEs), which captures the core mechanisms of disease transmission and recovery within a population. This formulation, despite its clarity and analytical tractability, assumes a perfectly smooth and predictable evolution of the disease. Recognizing that real epidemics are subject to random environmental fluctuations, we then extended the SIRS model by introducing a Wiener process to represent continuous, small scale stochastic perturbations. However, Brownian motion alone cannot capture abrupt, discontinuous events such as super-spreading episodes or sudden policy changes. To overcome this limitation, we further incorporated a Lévy jump process, which adds rare but large discontinuities to the system dynamics. Thus, Chapter 1 progressively built a hierarchy of models: from deterministic ODEs, to diffusion-driven stochastic differential equations, and finally to a jump-diffusion framework driven by a Lévy process. This final formulation serves as the mathematical foundation for the optimal control analysis developed in the second chapter

- In the next chapter, we formulate the problem of optimal control by considering two goals, minimizing the number of infected people and the total costs

of control interventions. The controls considered in this thesis are immigration control (restricting the influx of susceptible individuals) and preventive measures, such as social distancing, wearing masks, and vaccination. To derive optimal control, we apply the stochastic maximum principle for Lévy processes, which accounts for both continuous Brownian fluctuations and discontinuous jumps. We obtain a Hamiltonian involving state variables, control variables, and adjoint variables. The necessary optimality conditions lead to a system of adjoint (backward) stochastic differential equations with jumps, while the optimal controls are obtained by minimizing the Hamiltonian with respect to the control variables point-wise. The resulting forward-backward system must be solved numerically, which we address in Chapter 3.

- In the last chapter, we use The LSMC algorithm based on Milstein scheme in order to simulate the forward–backward equations from Chapter 2. We present the simulations via figures, which clearly illustrate how our optimal control has influenced the reduction of the susceptible and infected individuals over time. This clearly demonstrates the effectiveness of the proposed optimal control technique.

Chapter 1

Deterministic and stochastic epidemic models

1.1 Introduction and motivation

In recent years, some contagious diseases like SARS, Ebola virus, and COVID-19 have caused a serious impact on health aspects as well as economic systems all over the world. Studying the dynamics of contagious diseases in a population has therefore recently gained a great deal of attention in the scientific fields. In this context, mathematical modeling has a central role to play in studying contagious diseases. Among the different methods in mathematical modeling of contagious diseases are "Deterministic Models" and "Stochastic Models". Deterministic models are mathematical representations that model ideal scenarios regarding contagious diseases. On the contrary to deterministic models, stochastic models are mathematical representations that study contagious diseases in a more realistic context. The integration of information related to deterministic models and stochastic models can provide a deeper understanding related to contagious diseases.

1.2 Deterministic epidemic models

Epidemiologists typically employ deterministic models, such as the SI (Susceptible-Infected), SIR (Susceptible-Infected-Recovered), SIS and SEIR models to figure out how diseases spread in a group of people and how to stop them. These models make some educated approximations to help us understand how real-world epidemics act in a more straightforward way. The compartments are:

- **Susceptible(S):** Persons that are in good health and have not contracted the infection but are liable to get infected.

- **Infected (I):** The people that are infected and are capable of spreading the infection to others.
- **Exposed (E):** Compartment referred to the individuals that despite infected, do not exhibit obvious signs of infection and the abundance of the pathogen may be too low to allow further transmission.
- **Removed (R):** Persons who have recovered from a disease and are not susceptible to reinfection.

1.2.1 Hamer's simple deterministic model (SI model)

The Hamer model was first introduced by William Heaton Hamer in 1906, who also named the SI model. This is the first straightforward deterministic model in which the entire population N can be split into two groups: those who are susceptible to infection S and those who are infected I and hence contagious [5]. Hamer first assumed the following in order to demonstrate his model:
- There are no recoveries or deaths (fixed population number);
- The total population is constant: $N = S + I$;
- Once infected, a person becomes contagious and stays contagious for the remainder of their lives;
- Direct contact between susceptible people and infected people is how the infection spreads.

The differential equation system that describes this model is:

$$\begin{cases} \dfrac{dS(t)}{dt} = -\beta S(t)I(t) \\ \\ \dfrac{dI(t)}{dt} = \beta S(t)I(t) \end{cases} \tag{1.1}$$

- Where β is the transmission rate.

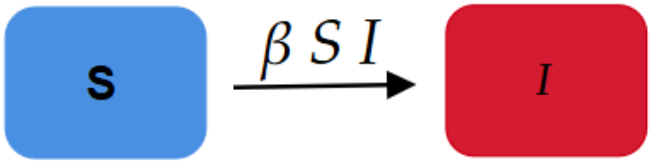

Figure 1.1: The SI scheme

1.2.2 The SIS model

The two kinds of persons in this model infected individuals (I) and healthy individuals (S) are the most basic for explaining how an epidemic spreads. There are two transformation processes in the model. In the first, a healthy person contracts the infection by coming into touch with other affected people. The graphic in figure 1.1 illustrates this process, which happens at a constant infection rate. In the second instance, an infected person recovers on their own. More specifically, an infected person who has recovered from the illness does not develop immunity and is instantly vulnerable once more [10]. The graphic in figure 1.2 illustrates this process, which proceeds at a steady rate of recoveryγ. This paradigm, which correlates to some illnesses like tuberculosis, does not offer immunity, so that leads us to the following system:

$$\begin{cases} \dfrac{dS}{dt} = -\beta SI + \gamma I \\ \dfrac{dI}{dt} = \beta SI - \gamma I \end{cases} \tag{1.2}$$

with the total population is constant: $N = S + I$

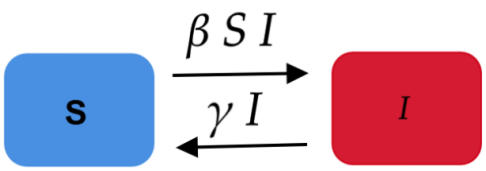

Figure 1.2: The SIS scheme

1.2.3 The SIR model Kermack–McKendrick epidemic model

The SIR model classifies hosts within a population as Susceptible, Infected, and Recovered [26] and was first thoroughly examined by Kermack and McKendrick. It depicts the processes of acute infections that, if cured, provide lifetime immunity. Diseases to which people develop lifelong immunity, and diseases like measles, smallpox, diphtheria, chickenpox, typhoid fever, mumps, can be treated using this strategy.
Two scenarios that differ in whether demographic factors are included or excluded will be examined.

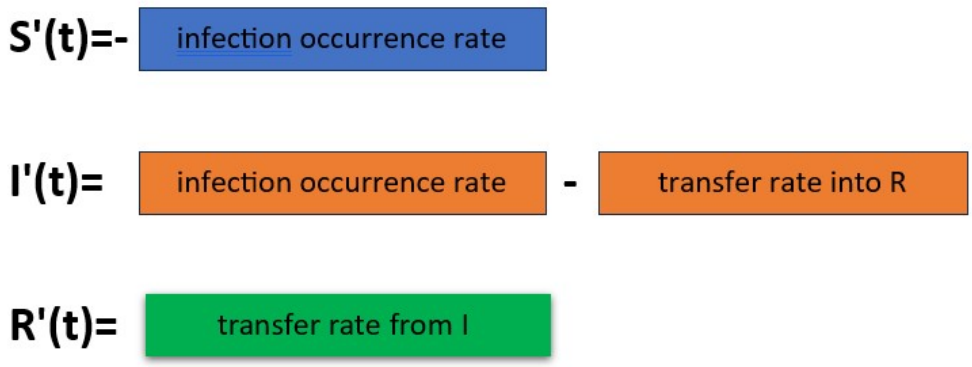

Figure 1.3: Differential equations of the SIR epidemic model.

1.2.3.1 The SIR model without demographic factors

After compartmentalizing the population, we need a set of equations that define the change over time in compartment sizes. The SIR model, excluding deaths and births, can be defined by the following system of differential equations:

$$\begin{cases} \frac{dS}{dt} = -\beta SI, \\ \frac{dI}{dt} = \beta SI - \gamma I, \\ \frac{dR}{dt} = \gamma I. \end{cases} \tag{1.3}$$

Figure 1.4: Diagram for SIR model without demography.

The total population is constant: N = S + I + R.
In addition, the transmission rate is β, and the recovery rate is γ, where $S(0) > 0, I(0) > 0$, and $R(0) > 0$.

1.2.3.2 SIR model with varying total population size

By including the movement of immigration and various demographic rates, which make the size of the entire population changeable, the deterministic model (1.3) can be further enhanced. We derive a modified SIR epidemic model based on these hypotheses. This model is appropriate for modeling a variety of diseases such as HBV

and Morbilli disease (measles). The SIR model with varying population assumes the following:

- Λ recruitment rate $\Lambda > 0$ corresponding to births and immigration.
- All people are born sensitive to infection.
- A disease-related mortality rate $\epsilon > 0$ in addition to the natural rate μ.

The system of differential equations for the model is as follows:

$$\begin{cases} \dfrac{dS(t)}{dt} = \Lambda - \beta S(t)I(t) - \mu S(t) \\ \dfrac{dI(t)}{dt} = \beta S(t)I(t) - (\gamma + \mu + \epsilon)I(t) \\ \dfrac{dR(t)}{dt} = \gamma I(t) - \mu R(t) \end{cases} \tag{1.4}$$

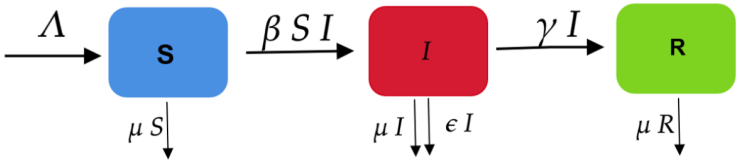

Figure 1.5: Diagram for SIR model with varying total population size.

1.2.4 SIRS model (SIR with temporary immunity)

The SIR epidemic model can be modified to incorporate provisional immunity, where members of the removed compartment may loss immunity over time and return to the susceptible class [7]. This extension is referred to as the SIRS epidemic model. By denoting the rate of immunity loss by l, the SIRS model is illustrated in figure 1.6 with the following corresponding system of differential equations:

$$\begin{cases} \dfrac{dS(t)}{dt} = \Lambda - \beta S(t)I(t) - \mu S(t) + lR(t) \\ \dfrac{dI(t)}{dt} = \beta S(t)I(t) - (\gamma + \mu + \epsilon)I(t) \\ \dfrac{dR(t)}{dt} = \gamma I(t) - (\mu + l)R(t) \end{cases} \tag{1.5}$$

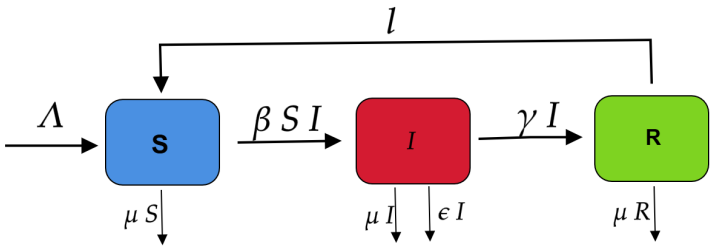

Figure 1.6: Diagram for SIRS model with temporary immunity.

1.2.5 SEIR model

By including an E (Exposed) compartment, which represents those who have been exposed to the disease but are not yet contagious, the SEIR model expands upon the SIR model. The SEIR framework, for instance, can be used to represent COVID-19 since it has an incubation phase during which people are exposed but not yet contagious [5]. Since $N = S + E + I + R$, the total population is constant. The following differential equations describe this model:

$$\begin{cases} \frac{dS}{dt} = -\beta SI, \\ \frac{dE}{dt} = \beta SI - \sigma E, \\ \frac{dI}{dt} = \sigma E - \gamma I, \\ \frac{dR}{dt} = \gamma I \end{cases} \tag{1.6}$$

where :
β : Transmission rate.
σ : the rate at which exposed individuals become infectious
γ : Recovery rate.

1.3 Basic reproduction number R_0

In epidemiology, the basic reproduction number R_0 is a quantity in epidemic modeling that shows how fast an infectious disease can spread in a population that is completely susceptible. In a formal way, R_0 is defined as the anticipated number of secondary cases generated by a single infectious individual during their infectious period, in the absence of immunity and control measures. In compartmental models such as SIS, SIR, and SEIR, R_0 comes naturally from the ratio of the transmission mechanism to the removal dynamics. For example, in the SIR family of models, it is given by $R_0 = \frac{\beta}{\gamma}$, where β is the effective contact or transmission rate and γ is

the recovery rate. In a broader sense, in multi-dimensional or structured epidemic models, R_0 is calculated as the spectral radius of the next-generation matrix, which distinguishes new-infection terms from transition dynamics. The condition $R_0 > 1$ means that the infection can spread through the population and cause an epidemic. On the other hand, $R_0 < 1$ indicates that the disease will eventually die out. In the realm of uncertainty-informed epidemic management, assessing and governing R_0 becomes difficult because of random changes, uncertainty about parameters, and not having all the information about how the gearbox works. So, a lot of modern control strategies try to keep the effective reproduction number R_0 below unity, even with these uncertainties, to make sure that disease spread is strongly stopped.

1.3.1 Basic reproduction number for SIRS model

The basic reproduction number R_0 is computed using the next-generation matrix method [8]. The only infected compartment is I. Let $\mathcal{F}$ denotes the rate of appearance of new infections and $\mathcal{V}$ the rate of transfer of individuals out of the infected class. From the infected equation of the SIRS model (1.5), we have:

$$\mathcal{F} = \beta SI, \qquad \mathcal{V} = (\mu + \epsilon + \gamma)I.$$

In the disease-free equilibrium, $I^* = R^* = 0$ in the system (1.5), and its first equation becomes $0 = \Lambda - \mu S^*$, and so $S^* = \dfrac{\Lambda}{\mu}$. We deduce that the equilibrium point of the system is $E^0 = \left(\dfrac{\Lambda}{\mu}, 0, 0\right)$. Evaluating the Jacobian matrices of $\mathcal{F}$ and $\mathcal{V}$ at E^0, we obtain

$$F = \left[\frac{\partial \mathcal{F}}{\partial I}\right]_{E^0} = \beta\frac{\Lambda}{\mu}, \qquad V = \left[\frac{\partial \mathcal{V}}{\partial I}\right]_{E^0} = \mu + \epsilon + \gamma.$$

The next-generation matrix is therefore $K = FV^{-1}$, and the basic reproduction number, defined as the spectral radius of K, is given by

$$R_0 = \rho(K) = \frac{\beta\Lambda}{\mu(\gamma + \mu + \epsilon)}.$$

1.3.2 Basic reproduction number for COVID 19

To illustrate the impact of the reproduction number R_0 on the spread of an epidemic, we consider the SEIR model (1.6) for COVID-19 in Algeria [4]. Let's consider a simplified estimation of R_0 for COVID-19:

- **Transmission rate (β):** The rate at which susceptible individuals come into contact with infected individuals and become infected. For COVID-19, early

estimates placed β around 0.41 per day.

- **Recovery rate (γ):** The rate at which infected individuals recover. In the case of COVID-19, with a mean recovery time of 10 days, γ is 0.1 per day.

Using the formula for R_0:

$$R_0 = \frac{\beta}{\gamma}$$

Substituting the values:

$$R_0 = \frac{0.41}{0.1} = 4.1$$

This suggests that, in the absence of immunity or interventions, each infected person could potentially spread the virus to **4 other individuals** on average.

Interpretation:

- If $R_0 > 1$, the disease is likely to spread rapidly. With a $R_0 = 4.1$, COVID-19 was able to cause rapid outbreaks.
- If $R_0 = 1$, the disease would only remain at a stable level.
- If $R_0 < 1$, the disease would eventually die out.

In the beginning, figuring out R_0 for COVID-19 showed how important it was to take steps to lower R_0 and stop the virus from spreading, such as staying away from other people, wearing masks, and being vaccinated.

Simulation of the SEIR Model for COVID-19 with $R_0 = 4.1$

The graphic below shows the simulation of how COVID-19 spreads under the SEIR model with an estimated $R_0 = 4.1$.

- The susceptible population $S(t)$ decreases over time as individuals come into contact with infected individuals and transition into the exposed compartment.
- The exposed population $E(t)$ increases as susceptible individuals are exposed to the virus but are not yet infectious.
- The infected population $I(t)$ grows rapidly as exposed individuals transition to infectious individuals, before peaking and eventually decreasing as they recover or are removed.
- The recovered (or removed) population $R(t)$ increases as infected individuals recover from the disease or pass away, depending on the model's assumptions.

This graph illustrates the dynamics of the epidemic and shows how the infection peaks and then subsides as interventions or natural immunity kick in.

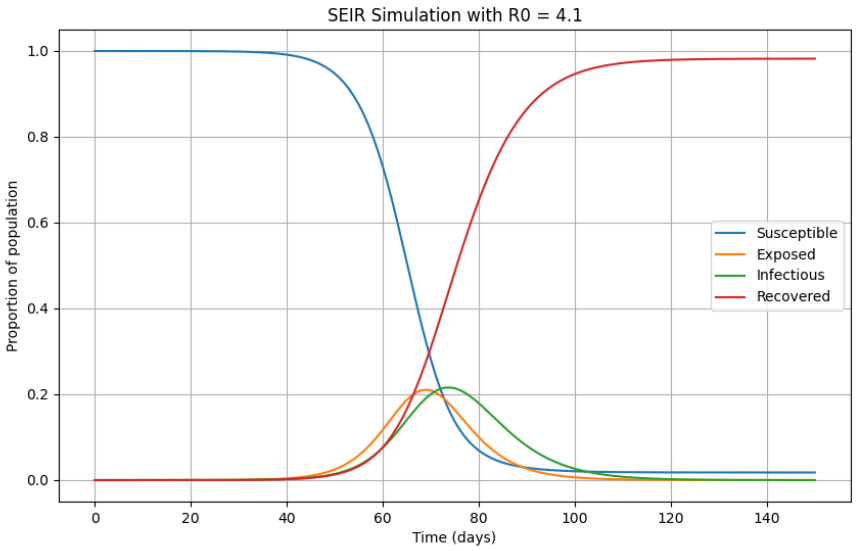

Figure 1.7: SEIR Model Simulation for COVID-19 With $R_0 = 4.1$

1.4 Equilibrium in epidemic models

In epidemic models, equilibrium points are steady states of the system where the number of people in each compartment stays the same over time. These equilibria offer significant understanding of the long-term dynamics of diseases and the possible consequences of an epidemic.

1.4.1 Disease-free equilibrium (DFE)

The disease-free equilibrium denotes the condition in which the disease is non existent within the population. In compartmental models like the SIRS model, this balance is shown by a population of people who are susceptible and no one who is infected. In mathematical terms, the disease-free equilibrium requires:

$$I^* = 0$$

while all other compartments must have non negative values that are consistent with the total population size. The basic reproduction number R_0 usually determines how stable the disease-free equilibrium is. If $R_0 < 1$, the disease-free equilibrium is locally asymptotically stable. This means that the disease will eventually die out.

1.4.2 Endemic equilibrium

The endemic equilibrium means that the disease is always present in the population. At this equilibrium point, the number of people who are infected stays positive over time. This happens when the disease can spread on its own, which is usually when $R_0 > 1$. The endemic equilibrium is a key idea for understanding how diseases can last a long time. It shows how new infections and recovery or removal processes are balanced.

1.5 Stochastic epidemic model

Epidemic modeling represents a fundamental methodology used in assessments of the dynamics of disease spread within populations. Although traditional epidemic models such as the SIRS model provide insights into the spread of infectious diseases, they do not take into consideration the intrinsic randomness and sudden,large-scale events that may drastically alter the course of an epidemic. In this section, we extend the classical deterministic SIRS model with stochastic fluctuations and jump processes, which account for sudden environmental changes such as mass infections, super-spreader events, or environmental shocks. As discussed by May and Mao et al. [22], random fluctuations have a great influence on the dynamics of ecological and epidemiological systems. To account for these fluctuations, Jiang et al. [14] added random fluctuations to the standard deterministic SIR model through the inclusion of Brownian motions. However, to capture the sudden large events that can cause discontinuities in epidemic models, like pandemics created by super-spreaders, or sudden changes in the environment, such as volcanic eruptions and hurricanes, Gaussian noise is no longer be appropriate. To transcend this deficiency, Zhang and Wang [32] proposed a Lévy-type jump process within the stochastic SIR model in order to capture such discontinuous events.

Remark 1. *To more accurately reflect epidemiological realities, our study focuses on the SIRS model. This analytical framework accounts for dynamic fluctuations in the total number of individuals through demographic shifts and disease-induced mortality, providing a more robust perspective on the epidemic's time evolution.*

1.5.1 Stochastic SIRS Model with Brownian noise

Applications of Brownian motion in epidemiology

- Small Scale Epidemics: When the number of people or the geographical area in which the epidemic occurs is small, the role of randomness can be substantial.

Brownian motion is a more refined model of the phenomenon if the dynamics are described by deterministic models that overestimate the rate of the spread of the disease or ignore the sudden changes in the course of the epidemic.

- Threshold Values of Stochastic Epidemics: Brownian motion can be used to model the threshold of an epidemic, the point beyond which the epidemic can die or spread exponentially. In fact, the stochastic variations may result in the change of the threshold values, allowing us to have a clearer view of the likelihood of the epidemic turning into a pandemic.

- variations in the spread of diseases: The Brownian motion process may help model the 'noise' caused by variations in the rate of spread, interventions, or changed behaviors in the spread of diseases.

Modeling epidemic dynamics with Brownian motion and stochastic differential equations:

In modeling epidemic dynamics, the introduction of Brownian motion offers a way to introduce randomness and uncertainty in the system. Brownian motion is the stochastic process with continuous paths, having independent and stationary increments; making it ideal for modeling random fluctuations that affect disease transmission. To represent such randomness in epidemic models, we use Stochastic Differential Equations (SDEs), which describe how the state of a system evolves over time under both deterministic and random influences. A general SDE of epidemic dynamics may take the form:

$$dX_t = \mu dt + \sigma dB_t$$

where B_t is the Brownian motion factor that introduces noise to the system, and μ, σ are drift and volatility, respectively. We make stochastic epidemic models by adding Brownian motion to epidemic models. These models let infection and recovery rates change randomly, which is how the disease spreads in the real world. These models help us understand how randomness can change the likelihood of an epidemic ending or becoming endemic. Stochastic models also take into consideration the fact that even small random changes can have a big effect on the course of the disease, which can lead to outcomes that are different from what deterministic models predict. This unpredictability is important for modeling epidemic outbreaks that are more accurate since it reflects the natural uncertainty in how diseases spread in the real world [6].

Derivation of the stochastic SIRS model with Brownian motion:

To include random fluctuations, we add stochastic components to the terms of the deterministic model that represent random noise [21]. The modified system (1.5) with Brownian noise is as follows:

$$\begin{cases} dS(t) &= (\Lambda - \beta S(t)I(t) - \mu S(t) + lR(t))\,dt + \sigma_1 S(t)dB_1(t), \\ dI(t) &= (\beta S(t)I(t) - (\gamma + \mu + \epsilon)I(t))\,dt + \sigma_2 I(t)dB_2(t), \\ dR(t) &= (\gamma I(t) - (\mu + l)R(t))\,dt + \sigma_3 R(t)dB_3(t), \end{cases} \tag{1.7}$$

where $B_1(t)$, $B_2(t)$, and $B_3(t)$ are independent Brownian motions, and σ_1, σ_2, and σ_3 represent the noise intensities associated with each compartment. The Brownian motions show how the dynamics of people who are susceptible, infected, or recovered change randomly. The noise terms model the randomness in spread and recovery of the disease are.

1.5.2 Incorporating jump processes in SIRS model

The Brownian motion depicts continuous fluctuations, but it fails to accurately represent abrupt changes or discontinuities, such as rapid mass infections or significant environmental shocks. Lévy-type jump processes are superior for modeling these events. To account for these discontinuities, we need to improve the stochastic SIRS model by incorporating a lévy jump process. This process features Poisson jumps, which signify abrupt large-scale occurrences [23].

Lévy processes:

To analyze epidemics, it is important to consider the likelihood of a significant amplitude surge in the number of infected or deaths. This is supported by both theoretical and actual research. To mathematically quantify these occurrences, jump processes provide an appropriate environment for describing and forecasting the epidemic's future. In this section, we describe a Lévy process, named after French mathematician Paul Lévy, and show how diverse a category of processes they create. We will provide a quick summary of the theory, as detailed explanations may be found in various books and documents. For more information, please refer to the book [1].

Definition 1.1. *Let $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\geq 0}, \mathbb{P})$ be a filtered probability space. A process $Z(t) = \{Z_t\}_{t\geq 0}$ defined on $\mathbb{R}$ is called a Lévy process if it satisfies the following conditions:*

1. $Z(0) = 0$ *almost surely.*

2. $Z(t)$ *has independent increments, meaning that the increment* $Z(t+s) - Z(t)$ *is independent of* $\mathcal{F}_t$ *(the filtration up to time* t*).*

3. $Z(t)$ *has stationary increments, i.e., the distribution of* $Z(t+h) - Z(t)$ *depends only on* h *and not on* t*.*

4. $Z(t)$ *is stochastically continuous, meaning for all* $t \geq 0$ *and* $x > 0$*, the probability* $\mathbb{P}(|Z(t+\epsilon) - Z(t)| > x) \to 0$ *as* $\epsilon \to 0$*.*

Let $Z(t)$ be a Lévy process. Then, $Z(t)$ has a cadlàg version (right-continuous with left limits), which is also a Lévy process [31]. Furthermore, let $\mathcal{B}_0$ be the collection of Borel sets $D \subset \mathbb{R}$ whose closure $\bar{D}$ does not contain zero. By denoting the jump of $Z(t)$ at time $t > 0$ as $\Delta Z(t) = Z(t) - Z(t^-)$, we associate the counting measure $N(t, D)$ to $Z(t)$ as follows:

$$N(t, D) = N(t, D, \omega) = \sum_{0 \leq s \leq t} \mathbb{1}_D(\Delta Z(s)), \tag{1.8}$$

for all $D \in \mathcal{B}_0$, where $\mathbb{1}_D$ is the indicator function for the set D.
This counting measure $N(t, D)$ represents the total number of jumps of size $\Delta Z(t)$ in D that occur before or at time t.
In general, the expression above corresponds to the Poisson random measure, which counts the number of jumps of $Z(t)$ in a Borel set D up to time t. The process is cadlàg, meaning the number of jumps $N(t, D)$ is almost surely finite for all $D \in \mathcal{B}_0$. A classic example of a Lévy process is Brownian motion, denoted $B(t)$, which has stationary and independent increments. Therefore, $B(t)$ is a continuous Lévy process. Another important example is the Poisson process.

Poisson process

The Poisson process $\theta(t)$ with associated intensity $\lambda > 0$ is a Lévy process taking values in $\mathbb{N} \cup \{0\}$ such that:

$$P(\theta(t) = n) = \frac{(\lambda t)^n e^{-\lambda t}}{n!}, \quad n = 0, 1, 2, \ldots$$

Returning to the definition given in equation (1.8), the set function $D \mapsto N(t, D, \omega)$ is a σ-finite measure on $\mathcal{B}_0$ for each fixed t and ω, with the associated differential form $N(t, D, \omega)$. Furthermore, for each fixed ω, the set function $[a, b] \times D \mapsto N(A, D, \omega)$ defines a σ-finite measure with the associated differential form $N(t, D, \omega)$ for $D \in \mathcal{B}_0$.

Now, considering the mean measure:

$$\pi(D) = \mathbb{E}(N(1, D)),$$

this defines a σ-finite measure on $\mathcal{B}_0$, known as the Lévy measure associated with $Z(t)$. By fixing $D \in \mathcal{B}_0$, the process:

$$\theta_D(t) = N(t, D, \omega)$$

is a Poisson process with intensity $\lambda = \pi(D)$.

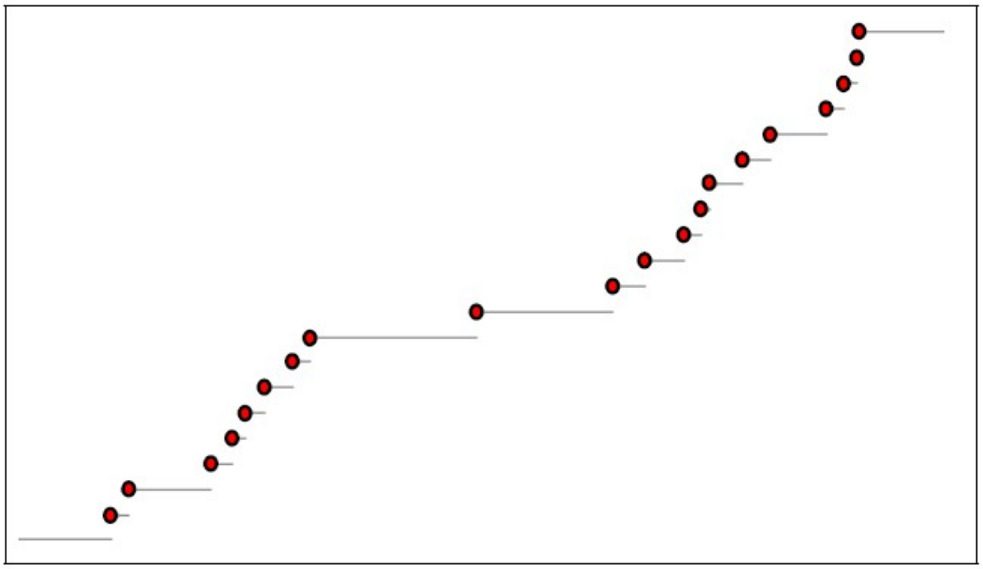

Figure 1.8: Visualization of the Poisson process.

Compound Poisson process

Let $X(n)$, $n \in \mathbb{N}$, be a sequence of i.i.d. random variables in $\mathbb{R}$ with a common finite distribution μ_X. Let $\theta(t)$ be a Poisson process with intensity λ, independent of all $X(n)$. The compound Poisson process $X^\theta(t)$ is expressed as:

$$X^\theta(t) = X(1) + \cdots + X(\theta(t)), \quad t > 0.$$

An increment of this process is defined by:

$$X^\theta(s) - X^\theta(t) = \sum_{k=\theta(t+1)}^{\theta(s)} X(k), \quad s > t.$$

The distribution of $X^\theta(t)$ depends only on the difference $(s-t)$, which implies that $X^\theta(t)$ is a Lévy process.[27] The Lévy measure π of $X^\theta(t)$ is given by

$$\pi(dz) = \lambda\, \mu_Z(dz),$$

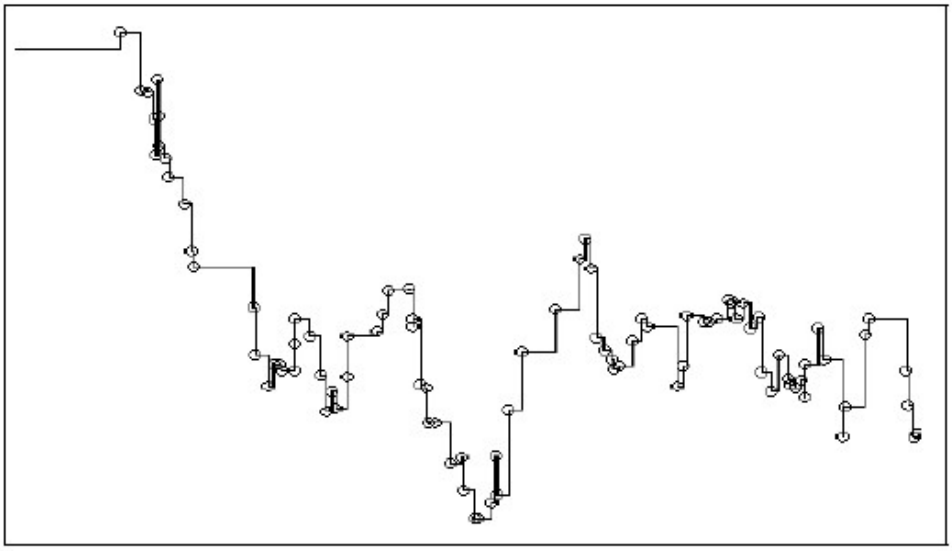

Figure 1.9: Illustration of the compound Poisson process.

Figures 1.9 and 1.8 are taken from the dissertation[27].

Proposition 1.1. *(Itô-Lévy decomposition) [1]: Let $Z(t)$ be a Lévy process. Then, $Z(t)$ admits the following unique representation:*

$$Z(t) = \alpha t + \sigma B(t) + \int_{|z|<r} z\, \tilde{N}(t, dz) + \int_{|z|\geq r} z\, N(t, dz)$$

where $\alpha \in \mathbb{R}$, $\sigma \in \mathbb{R}$, and $r \in [0, \infty)$, and the process $\tilde{N}(dt, dz)$ is the compensated Poisson random measure of $Z(t)$ such that:

$$\tilde{N}(dt, dz) = N(dt, dz) - \pi(dz)dt$$

It is assumed that the standard Brownian motion $B(t)$ is independent of $\tilde{N}$. Note that for each $D \in \mathcal{B}_0$, the process $M(t) = \tilde{N}(t, D)$ is a martingale. Specifically, if $\alpha = 0$ and $r = \infty$, we call $Z(t)$ a Lévy martingale. Furthermore, if $\mathbb{E}[|Z(t)|] < \infty$ for all $t > 0$, we can choose $r = \infty$. Hence, $Z(t)$ can be written in compact form:

$$Z(t) = \alpha t + \sigma B(t) + \int_{\mathbb{R}} z\, \tilde{N}(t, dz).$$

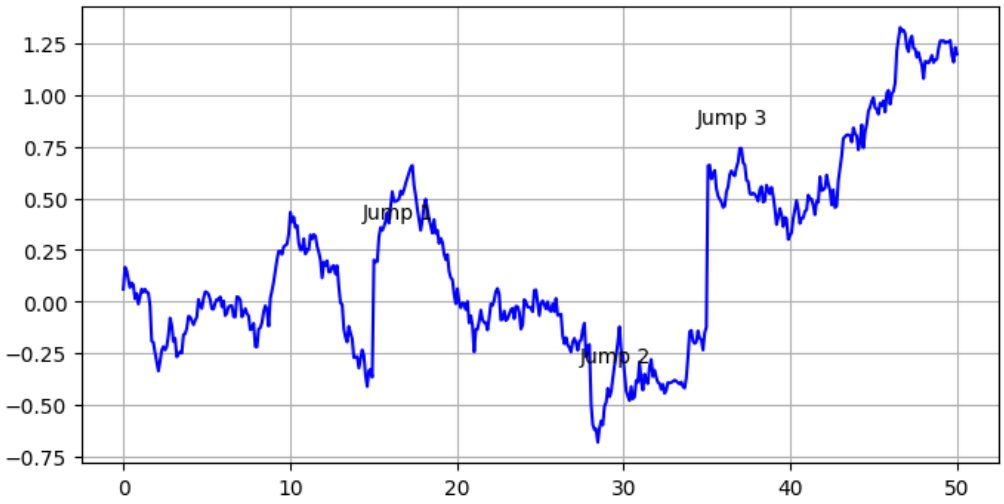

Figure 1.10: Illustration of the Itô-Lévy jumps process paths.

Theorem 1.1. *[24] A Lévy process is a semi-martingale.*

SIRS model with lévy Jumps

lévy processes are ideal for representing discontinuous events as the increments are independent and do not vary with time. They provide a framework within which the sudden changes in how epidemics work, such as:

- Super-spreader events: A small number of people infect a lot of other people in a short amount of time.
- Environmental shocks : sudden environmental changes, such the arrival of a novel disease variation or a natural disaster, that produce big changes in the rates of disease transmission.

Adding jump processes to our model lets us study how such severe events affect the spread of the disease, making the model more useful in real-life.
The altered system with jump processes is:

$$\begin{cases} dS(t) &= (\Lambda - \beta S(t)I(t) - \mu S(t) + lR(t))\,dt + \sigma_1 S(t)dB_1(t) + \int_{\mathbb{R}^*} C_1(z)S(t^-)\tilde{N}(dt,dz), \\ dI(t) &= (\beta S(t)I(t) - (\gamma + \mu + \epsilon)I(t))\,dt + \sigma_2 I(t)dB_2(t) + \int_{\mathbb{R}^*} C_2(z)I(t^-)\tilde{N}(dt,dz), \\ dR(t) &= (\gamma I(t) - (\mu + l)R(t))\,dt + \sigma_3 R(t)dB_3(t) + \int_{\mathbb{R}^*} C_3(z)R(t^-)\tilde{N}(dt,dz), \end{cases} \tag{1.9}$$

where:

- $\tilde{N}(dt,dz)$ is the compensated Poisson process with:

$$\tilde{N}(dt,dz) = N(dt,dz) - \pi(dz)dt.$$

- $C_1(z), C_2(z), C_3(z)$ are the magnitudes of the jumps in the susceptible, infected, and recovered compartments, respectively.
- The integral term $\int_{\mathbb{R}^*}$ sums the possible jump magnitudes, and the jumps happen randomly, modeled by the Poisson process $N(dt, dz)$. This allows for the representation of abrupt changes in the number of susceptible, infected, or recovered resulting from shocks.

1.6 Existence and uniqueness of the positive solution

In this section, using the Lyapunov analysis method, we prove that jump processes can suppress the explosion and that the system (1.9) admits a unique global positive solution [32].
Assume that for each $m > 0$ there exists a constant $L_m > 0$ such that

- (H1) $\int_{\mathbb{R}^*} |H_i(x,z) - H_i(y,z)|^2 \pi(dz) \leq L_m |x-y|^2$, $i = 1, 2, 3$,
  where $H_1(x,z) = C_1(z)S(t^-)$, $H_2(x,z) = C_2(z)I(t^-)$ and $H_3(x,z) = C_3(z)R(t^-)$ with $|x| \vee |y| \leq m$.
- (H2) $|\ln(1 + C_i(z))| \leq K_i$, for $C_i(z) > -1$, where K_i is positive constant, for $i = 1, 2, 3$.

Theorem 1.2. *Under assumptions* (H1) – (H2)*, and for any initial value:*

$$(S(0), I(0), R(0)) \in \mathbb{R}_+^{*3},$$

the system (1.9) *admits a unique global positive solution almost surely.*

Proof. By (H1) the drift and the diffusion are local Lipchitz, and for any initial value $(S(0), I(0), R(0)) \in \mathbb{R}_+^{*3}$, there exists a unique local solution $(S(t), I(t), R(t))$ in $[0, \tau_e)$, where τ_e is the explosion time. To show that the solution is global, i.e., $\tau_e = \infty$, almost surely, we use a Lyapunov function argument.
Define $F : \mathbb{R}_+^{*3} \to [0, \infty)$ by

$$F(S, I, R) = \left(S - a - a \ln \frac{S}{a}\right) + (I - 1 - \ln I) + (R - 1 - \ln R),$$

where $a > 0$ is a constant that will be chosen later. This function is $\mathcal{C}^2$ and tends to $+\infty$ as any component approaches 0 or $+\infty$.
Let $m_0 > 0$ be such that $S(0), I(0), R(0) \in [1/m_0, m_0]$. For any integer $m \geq m_0$, define the stopping time:

$$\tau_m = \inf \left\{ t \in [0, \tau_e) : S(t) \notin \left(\frac{1}{m}, m\right) \text{ or } I(t) \notin \left(\frac{1}{m}, m\right) \text{or } R(t) \notin \left(\frac{1}{m}, m\right) \right\}.$$

Clearly, τ_m is increasing as $m \to \infty$ a.s. Set $\tau_\infty = \lim_{m\to\infty} \tau_m$, whence, $\tau_\infty \leq \tau_e$ a.s. If we can show that $\tau_\infty = \infty$ is true, then $\tau_e = \infty$ and $(S(t), I(t), R(t)) \in \mathbb{R}^3_+$ a.s. If this statement is false, then there exist a pair of constants $T > 0$ and $0 < \eta < 1$ such that

$$\mathbb{P}(\tau_\infty \leq T) \geq \eta. \tag{1.10}$$

Now we apply Ito's formula for jump diffusion to $F(S(t), I(t), R(t))$. The process (S, I, R) has continuous part given by the drift and Brownian terms, and jumps given by the integrals with respect to $\widetilde{N}$. For a $\mathcal{C}^2$ function F:

$$\begin{aligned} dF(S(t), I(t), R(t)) = \mathcal{L}F(S(t), I(t), R(t))dt + (S(t) - a)\sigma_1 dB_1(t) + (I(t) - 1)\sigma_2 dB_2(t) \\ + (R(t) - 1)\sigma_2 dB_3(t) + \int_{\mathbb{R}^*} C_1(z)S(t^-) - a\ln(1 + C_1(z))\tilde{N}(dt, dz) \\ + \int_{\mathbb{R}^*} C_2(z)I(t^-) - \ln(1 + C_2(z))\tilde{N}(dt, dz) \\ + \int_{\mathbb{R}^*} C_3(z)R(t^-) - \ln(1 + C_3(z)]\tilde{N}(dt, dz) \end{aligned}$$

where $\mathcal{L}F$ is the generator or the drift part, which is given by:

$$\begin{aligned} \mathcal{L}F(S, I, R) = \Lambda + (\gamma + \mu + \epsilon) + \mu + l + a\mu - (\beta + \mu)S + (a\beta - \mu - \epsilon)I \\ - \mu R - \frac{a\Lambda}{S} - \frac{alR}{S} - \frac{\gamma I}{R} + \frac{a\sigma_1^2}{2} + \frac{\sigma_2^2}{2} + \frac{\sigma_3^2}{2} + \int_{\mathbb{R}^*} a\big(C_1(z) - \ln(1 + C_1(z))\big)\pi(dz) \\ + \int_{\mathbb{R}^*} \big(C_2(z) - \ln(1 + C_2(z))\big)\pi(dz) + \int_{\mathbb{R}^*} \big(C_3(z) - \ln(1 + C_3(z))\big)\pi(dz). \end{aligned}$$

Then

$$\begin{aligned} \mathcal{L}F(S, I, R) \leq \Lambda + (\gamma + \mu + \epsilon) + \mu + l + a\mu + (a\beta - \mu - \epsilon)I + \frac{a\sigma_1^2}{2} + \frac{\sigma_2^2}{2} + \frac{\sigma_3^2}{2} \\ + \int_{\mathbb{R}^*} a\big(C_1(z) - \ln(1 + C_1(z))\big)\pi(dz) + \int_{\mathbb{R}^*} \big(C_2(z) - \ln(1 + C_2(z))\big)\pi(dz) \\ + \int_{\mathbb{R}^*} \big(C_3(z) - \ln(1 + C_3(z))\big)\pi(dz). \end{aligned}$$

We choose a small enough so that $a\beta - \mu - \epsilon \leq 0$. For instance:

$$a = \frac{\mu + \epsilon}{2\beta}$$

then

$$\begin{aligned} \mathcal{L}F(S, I, R) \leq \Lambda + (\gamma + \mu + \epsilon) + \mu + l + a\mu + \frac{a\sigma_1^2}{2} + \frac{\sigma_2^2}{2} + \frac{\sigma_3^2}{2} \\ + \int_{\mathbb{R}^*} a\big(C_1(z) - \ln(1 + C_1(z))\big)\pi(dz) + \int_{\mathbb{R}^*} \big(C_2(z) - \ln(1 + C_2(z))\big)\pi(dz) \\ + \int_{\mathbb{R}^*} \big(C_3(z) - \ln(1 + C_3(z))\big)\pi(dz). \end{aligned}$$

By assumption (H2) and the inequality:

$$x - \ln(x+1) \geq 0, \quad x > -1,$$

we obtain:

$$\mathcal{L}F(S,I,R) \leq \Lambda + (\gamma + \mu + \epsilon) + \mu + l + a\mu + \frac{a\sigma_1^2}{2} + \frac{\sigma_2^2}{2} + \frac{\sigma_3^2}{2} + 3k = k_1$$

where

$$\begin{aligned} k = \max \Bigg\{ & \int_{\mathbb{R}^*} \left[aC_1(z) - a\ln\left(1 + C_1(z)\right) \right] \pi(dz), \\ & \int_{\mathbb{R}^*} \left[C_2(z) - \ln\left(1 + C_2(z)\right) \right] \pi(dz), \\ & \int_{\mathbb{R}^*} \left[C_3(z) - \ln\left(1 + C_3(z)\right) \right] \pi(dz) \Bigg\}. \end{aligned}$$

Therefore

$$\begin{aligned} \int_0^{\tau_m \wedge T} dF(S(t), I(t), R(t)) \leq & \int_0^{\tau_m \wedge T} k_1 \, dt \\ & + \int_0^{\tau_m \wedge T} \Big[\int_{\mathbb{R}^*} C_1(z) S(t^-) - a\ln(1 + C_1(z)) \tilde{N}(dt, dz) \\ & + \int_{\mathbb{R}^*} C_2(z) I(t^-) - \ln(1 + C_2(z)) \tilde{N}(dt, dz) \\ & + \int_{\mathbb{R}^*} C_3(z) R(t^-) - \ln(1 + C_3(z)] \tilde{N}(dt, dz) \Big] \end{aligned}$$

Taking expectation yields:

$$\mathbb{E}F\big(S(\tau_m \wedge T), I(\tau_m \wedge T), R(\tau_m \wedge T)\big) \leq F(S(0), I(0), R(0)) + k_1 T. \tag{1.11}$$

Set $\Omega_m = \{\tau_m \leq T\}$ for $m \geq m_0$ and by (1.10), we have $\mathbb{P}(\Omega_m) \geq \eta$. Note that for every $\omega \in \Omega_m$, there is $S(\tau_m, \omega)$ or $I(\tau_m, \omega)$ or $R(\tau_m, \omega)$ equal to m or $\dfrac{1}{m}$ and by (1.11), it yields:

$$\begin{aligned} F(S(0), I(0), R(0)) + k_1 T \geq & \, \mathbb{E}[\mathbb{1}_{\Omega_m} F(S(\tau_m, \omega), I(\tau_m, \omega), R(\tau_m, \omega))] \\ & \geq \varepsilon \left(m - a - a\ln\frac{m}{a} \right) \wedge \left(\frac{1}{m} - a - a\ln\frac{1}{am} \right) \\ & \quad \wedge (m - 1 - \ln m) \wedge \left(\frac{1}{m} - 1 - \ln\frac{1}{m} \right), \end{aligned}$$

where $\mathbb{1}_{\Omega_m}$ is the indicator function of Ω_m. Letting $m \to \infty$, leads to the contradiction $\infty > F(S(0), I(0), R(0)) + k_1 T = \infty$, then we must have $\tau_\infty = \infty$ a.s. Therefore, it implies that $S(t)$, $I(t)$ and $R(t)$ will not explode in finite time with probability one. □

Chapter 2

Optimal control for system of stochastic differential equations with Lévy jumps

In this chapter, we study the properties of the SIRS-type epidemic model incorporating Lévy jumps, modeled by Poisson processes. This type of stochastic modeling allows for a better description of the actual dynamics of epidemics, often marked by sudden variations. We first present the SIRS model without control with Lévy jumps, then introduce optimal control strategies based on controlled immigration and preventive measures like masks and social distancing. Finally, we formulate and analyze an optimal control problem based on the notion of the Hamiltonian, and establish the associated adjoint equations using stochastic maximum principle to then derive the expressions of the controls.

2.1 Optimal control background

Optimal Control (OC) is the study of how to identify the best control inputs and state trajectories for dynamic systems over a specific time period in order to improve a particular performance index. OC derives from the calculus of variations, which was created in the seventeenth century. The brachistochrone problem that Johann Bernoulli and other early mathematicians came up with made Newton, Leibniz, Jacob Bernoulli, L'Hôpital, and the Bernoulli family make considerable progress. See [26] and the references therein. These works were on how to identify the best ways to go when there are physical limits. Euler and Lagrange formalized the calculus of variations, resulting in the concept of least action and other essential requirements for optimality. In the late 1920s and early 1930s, economists like Roos, Evans, Hotelling, and Ramsey used these math tools to help them make decisions about the future and how to use resources. The transition from classical variational problems to modern optimal control theory was significantly influenced by military and aerospace applications after the 1950s. Two key advances are Bellman's Dynamic Pro-

gramming and Pontryagin and his colleagues' work on the Pontryagin Maximum Principle [17]. We might use these methods to tackle optimization problems that require to obey the principles of differential equations in a more ordered fashion. Adding control variables to state variables made it easier to deal with complicated dynamic systems. The theory of optimal control has also been broadened to include problems that involve restrictions, unpredictable dynamics, and systems where classical variational approaches don't function well or at all [26, 29].

2.2 Optimal control for the SIRS model with Lévy jumps

Let $T > 0$ be a fixed final time and let:

$$\mathcal{U} = \Big\{(u(\cdot), v(\cdot)) \;\Big|\; u, v \text{ are measurable},\ 0 \le u(t) \le u_{\max},\ 0 \le v(t) \le v_{\max},\ \forall t \in [0, T]\Big\}$$

be the set of admissible controls, where $u(t)$ represents the immigration control and $v(t)$ represents preventive measures such as masks and social distancing. We can take $u_{\max} = u_{\min} = 1$ for simplicity reasons. For any $(u, v) \in \mathcal{U}$, consider the controlled SIRS model with Lévy jumps:

$$\begin{cases} dS(t) &= (u(t)\Lambda - \beta(1 - v(t))S(t)I(t) - \mu S(t) + lR(t))\,dt + \sigma_1 S(t)dB_1(t) \\ &\quad + \displaystyle\int_{\mathbb{R}^*} C_1(z)S(t^-)\tilde{N}(dt, dz), \\ dI(t) &= (\beta(1 - v(t))S(t)I(t) - (\gamma + \mu + \epsilon)I(t))\,dt + \sigma_2 I(t)dB_2(t) \\ &\quad + \displaystyle\int_{\mathbb{R}^*} C_2(z)I(t^-)\tilde{N}(dt, dz), \\ dR(t) &= (\gamma I(t) - (\mu + l)R(t))\,dt + \sigma_3 R(t)dB_3(t) \\ &\quad + \displaystyle\int_{\mathbb{R}^*} C_3(z)R(t^-)\tilde{N}(dt, dz), \end{cases} \tag{2.1}$$

with initial condition $(S(0), I(0), R(0)) \in \mathbb{R}_+^{*3}$. Let the cost functional be defined by

$$J(u, v) = \mathbb{E}\left[\int_0^T f(t, X(t), u(t), v(t))dt + g(X(T))\right] \tag{2.2}$$

where $X(t) = (S(t), I(t), R(t))^\top$, $f : [0, T] \times \mathbb{R}^n \times U \to \mathbb{R}$ is a continuous function called *running cost*, $g : \mathbb{R}^n \to \mathbb{R}$ is $\mathcal{C}^1$ function called *terminal cost*, and $T < \infty$ is a positive fixed deterministic time.

In our case, the cost function will be:

$$J(u,v) = \mathbb{E}\left[\int_0^T \left(I(t) + \frac{\gamma_1}{2}u(t)^2 + \frac{\gamma_2}{2}v(t)^2\right) dt + I(T)\right], \tag{2.3}$$

where $\gamma_1, \gamma_2 > 0$ are weighting constants, and

$$f(t, X(t), u(t), v(t)) = I(t) + \frac{\gamma_1}{2}u(t)^2 + \frac{\gamma_2}{2}v(t)^2$$

and

$$g(X(T)) = I(T).$$

A pair of admissible controls $(u^*, v^*) \in \mathcal{U}$ is said to be an *optimal control* for the SIRS model with Lévy jumps if:

$$J(u^*, v^*) = \inf_{(u,v)\in\mathcal{U}} J(u, v).$$

The corresponding state trajectory $(S^*(t), I^*(t), R^*(t))$ solution to (2.1) associated with (u^*, v^*) is called the *optimal state*, and the pair $\big((S^*, I^*, R^*), (u^*, v^*)\big)$ is called an *optimal pair*.

2.3 Maximum principle

In the deterministic case this principle was first introduced by Pontryagin and his collaborators formulated the essential first-order requirements for optimal control. This result is regarded as one of the most significant achievements in Mathematics throughout the 20th century [28]. Pontryagin proposed an idea of adjoint functions to incorporate the differential equation to the objective functional. Adjoint functions provide a purpose similar to Lagrange multipliers in multivariate calculus, which incorporate constraints into the function of many variables to be optimized. A corresponding maximum principle for Ito diffusions was formulated by Kushner, Bismut and subsequently further developed by Bensoussan, Haussmann and others. For jump diffusions, a sufficient maximum principle has recently been formulated in [12].

2.3.1 Hamiltonian of the model

We define the Hamiltonian $H : [0,T] \times \mathbb{R}^n \times \mathcal{U} \times \mathbb{R}^n \times \mathbb{R}^{n\times m} \times \mathbb{R}^n \to \mathbb{R}$ by:

$$\begin{aligned} H(t, X, U, p, q, r) = f(t, X, U) + \sum_{i=1}^{n} b_i(t, X, U)p_i + tr(\sigma(t, X, U)q) \\ + \sum_{i=1}^{n} \int_{\mathbb{R}^*} \gamma_i(t, X, U, z)r_i(t, z)\pi(dz) \end{aligned} \tag{2.4}$$

where:

- $X \in \mathbb{R}^n$ vector of state variable

- $U = (u, v) \in \mathcal{U} \subset \mathbb{R}^k$ controls variable
- $f(t, X, U)$ running cost
- $b(t, X, U) \in \mathbb{R}^n$ drift coefficient
- $p \in \mathbb{R}^n$ adjoint variable associated with drift
- $\gamma_i(t, x, U, z)$ jump coefficient for the i-th component
- $r(t, z) \in \mathbb{R}^n$ adjoint variable associated with the jump part
- $\pi(dz)$ Lévy measure(jump intensity measure).

In our case, the Brownian adjoint process is a full matrix:

$$q(t) = \begin{pmatrix} q_{11}(t) & q_{12}(t) & q_{13}(t) \\ q_{21}(t) & q_{22}(t) & q_{23}(t) \\ q_{31}(t) & q_{32}(t) & q_{33}(t) \end{pmatrix} \in \mathbb{R}^{3\times 3}.$$

The diffusion matrix is diagonal:

$$\sigma(X) = \begin{pmatrix} \sigma_1 S & 0 & 0 \\ 0 & \sigma_2 I & 0 \\ 0 & 0 & \sigma_3 R \end{pmatrix}$$

The Hamiltonian associated to the control problem (2.1)–(2.3) is defined by:

$$\begin{aligned} H(t, X, u, v, p, q, r) = I &+ \frac{\gamma_1}{2}u^2 + \frac{\gamma_2}{2}v^2 + p_1\big(u\Lambda - \beta(1-v)SI - \mu S + lR\big) \\ &+ p_2\big(\beta(1-v)SI - (\gamma + \mu + \epsilon)I\big) + p_3\big(\gamma I - (\mu + l)R\big) \\ &+ \sigma_1 S q_{11} + \sigma_2 I q_{22} + \sigma_3 R q_{33} \\ &+ \int_{\mathbb{R}^*} \Big[C_1(z)Sr_1(z) + C_2(z)Ir_2(z) + C_3(z)Rr_3(z)\Big]\pi(dz). \end{aligned} \tag{2.5}$$

where $X(t) = (S(t), I(t), R(t))^\top$, $p = (p_1, p_2, p_3)^\top$ are the adjoint processes, and $q(t)$ and $r(t, z)$ are the Brownian and jump adjoint terms, respectively.

2.3.2 Derivation of the adjoint model

The adjoint equation corresponding to (u, v) and $X(u)$ is the unknown processes $p(t) \in \mathbb{R}^n, q(t) \in \mathbb{R}^{n\times m}$ and $r(t, z) \in \mathbb{R}^{n\times \ell}$ that describe the following backward stochastic differential equation:

$$\begin{cases} dp(t) = -\nabla_x H(t, X(t), u(t), p(t), q(t), r(t, \cdot))\, dt \\ \qquad\quad +q(t)\, dB(t) + \displaystyle\int_{\mathbb{R}^*} r(t^-, z)\tilde{N}(dt, dz); \quad t < T \\ p(T) = \nabla_x g(X(T)) = \nabla_x I(T). \end{cases}$$

The adjoint processes satisfy the backward stochastic differential equations with jumps:

$$\begin{cases} dp_1(t) = -\dfrac{\partial H}{\partial S}dt + q_{11}(t)dB_1(t) + q_{12}(t)dB_2(t) + q_{13}(t)dB_3(t) + \displaystyle\int_{\mathbb{R}^*} r_1(t,z)\tilde{N}(dt,dz), \\ dp_2(t) = -\dfrac{\partial H}{\partial I}dt + q_{21}(t)dB_1(t) + q_{22}(t)dB_2(t) + q_{23}(t)dB_3(t) + \displaystyle\int_{\mathbb{R}^*} r_2(t,z)\tilde{N}(dt,dz), \\ dp_3(t) = -\dfrac{\partial H}{\partial R}dt + q_{31}(t)dB_1(t) + q_{32}(t)dB_2(t) + q_{33}(t)dB_3(t) + \displaystyle\int_{\mathbb{R}^*} r_3(t,z)\tilde{N}(dt,dz), \end{cases} \tag{2.6}$$

with

$$p_1(T) = 0,\ p_2(T) = 1,\ p_3(T) = 0$$

where:

$$\begin{cases} \dfrac{\partial H}{\partial S} = -p_1\beta(1-v)I - p_1\mu + p_2\beta(1-v)I + \sigma_1 q_{11} + \displaystyle\int_{\mathbb{R}^*} C_1(z)r_1(z)\pi(dz), \\ \dfrac{\partial H}{\partial I} = 1 - p_1\beta(1-v)S + p_2\beta(1-v)S - p_2(\lambda+\mu+\epsilon) + p_3\gamma + \sigma_2 q_{22} + \displaystyle\int_{\mathbb{R}^*} C_2(z)r_2(z)\pi(dz), \\ \dfrac{\partial H}{\partial R} = p_1 l - p_3(\mu+l) + \sigma_3 q_{33} + \displaystyle\int_{\mathbb{R}^*} C_3(z)r_3(z)\pi(dz). \end{cases}$$

Then the adjoint model is:

$$\begin{aligned} dp_1(t) &= \left[p_1\beta I(1-v) + p_1\mu - p_2\beta(1-v)I - \sigma_1 q_{11} - \int_{\mathbb{R}^*} C_1(z)r_1(z)\pi(dz)\right]dt \\ &+ q_{11}(t)dB_1(t) + q_{12}(t)dB_2(t) + q_{13}(t)dB_3(t) + \int_{\mathbb{R}^*} r_1(t,z)\tilde{N}(dt,dz), \\ dp_2(t) &= \left[-1 + p_1\beta(1-v)S - p_2\beta(1-v)S + p_2(\lambda+\mu+\epsilon) - p_3\gamma - \sigma_2 q_{22} - \int_{\mathbb{R}^*} C_2(z)r_2(z)\pi(dz)\right]dt \\ &+ q_{21}(t)dB_1(t) + q_{22}(t)dB_2(t) + q_{23}(t)dB_3(t) + \int_{\mathbb{R}^*} r_2(t,z)\tilde{N}(dt,dz), \\ dp_3(t) &= \left[-p_1 l + p_3(\mu+l) - \sigma_3 q_{33} - \int_{\mathbb{R}^*} C_3(z)r_3(z)\pi(dz)\right]dt \\ &+ q_{31}(t)dB_1(t) + q_{32}(t)dB_2(t) + q_{33}(t)dB_3(t) + \int_{\mathbb{R}^*} r_3(t,z)\tilde{N}(dt,dz), \end{aligned} \tag{2.7}$$

with

$$p_1(T) = 0,\ p_2(T) = 1,\ p_3(T) = 0.$$

2.3.3 Optimal controls $u^*(t), v^*(t)$ of the model

The derivatives of the Hamiltonian (2.5) with respect to the controls u and v are:

$$\begin{aligned} \frac{\partial H}{\partial u} &= \gamma_1 u + p_1\Lambda \\ \frac{\partial H}{\partial v} &= \gamma_2 v + p_1\beta SI - p_2\beta SI. \end{aligned}$$

From $\frac{\partial H}{\partial u} = 0$ and $\frac{\partial H}{\partial v} = 0$. The optimal controls satisfy

$$\boxed{\begin{aligned} u^*(t) &= \max\left(0, \min\left(1, \frac{-\Lambda p_1(t)}{\gamma_1}\right)\right), \\ v^*(t) &= \max\left(0, \min\left(1, \frac{\beta S(t) I(t)(p_2(t) - p_1(t))}{\gamma_2}\right)\right). \end{aligned}} \tag{2.8}$$

Chapter 3

Numerical simulations for the controlled epidemic model

3.1 Introduction

In most practical applications, stochastic differential equations (SDEs) do not admit explicit analytical solutions. This is particularly true for epidemic models perturbed by stochastic noise, where randomness arises from environmental fluctuations, demographic variability, or sudden external shocks. As a result, numerical approximation schemes play a fundamental role in the qualitative and quantitative analysis of stochastic models.
This chapter is first devoted to the numerical approximation of SDEs. We present two widely used numerical methods: the Euler–Maruyama method and the Milstein method. A comparison between the two methods is then carried out in order to evaluate their accuracy and performance. The more efficient method is then chosen to be used in the next parts of the chapter. Also, we introduce the Least-Squares Monte Carlo method that we have used to solve the forward-backward stochastic system resulted from the Pontryagin's maximum principle for our controlled epidemic model. The last part of the chapter presents numerical simulations of the jump model, the optimal controls and their effects on the dynamic of the epidemic model in different situations. These results are performed in Python. The Python code is also provided at the end of this chapter.

3.2 Preliminaries on numerical methods for stochastic differential equations

Let $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, \mathbb{P})$ be a complete filtered probability space supporting a standard Brownian motion $B(t)$. A one-dimensional stochastic differential equation is written as

$$dX(t) = a(X(t), t)dt + b(X(t), t)dB(t), \quad X(0) = X_0. \tag{3.1}$$

A numerical scheme aims to approximate the true solution $X(t)$ at discrete time points $t_n = n\Delta t$, where $\Delta t = T/N$ is the time step.

3.2.1 Euler–Maruyama method

Definition

The Euler–Maruyama method is the simplest numerical approximation for SDEs and can be viewed as a stochastic extension of the classical Euler method for ordinary differential equations [3]. Given the SDE (3.1), the Euler–Maruyama approximation is defined by:

$$X_{n+1} = X_n + a(X_n, t_n)\Delta t + b(X_n, t_n)\Delta B_n, \tag{3.2}$$

where $\Delta B_n = B(t_{n+1}) - B(t_n)$ and $\Delta B_n \sim \mathcal{N}(0, \Delta t)$. The algorithm of this method is given by:

Algorithm 1 Euler–Maruyama Method

1: **Input:** Drift function $a(t, X)$, diffusion function $b(t, X)$, initial value X_0, initial time t_0, final time T, step size Δt
2: Set $t = t_0$
3: Set $X = X_0$
4: **while** $t < T$ **do**
5: Generate $\Delta B \sim \mathcal{N}(0, \Delta t)$
6: $X = X + a(t, X)\Delta t + b(t, X)\Delta B$
7: $t = t + \Delta t$
8: **end while**
9: **Output:** Approximate solution X

Convergence of Euler–Maruyama method

The Euler–Maruyama method converges strongly with order 1/2, [11] meaning that:

$$\mathbb{E}\big[|X(T) - X_N|\big] = O(\Delta t^{1/2}).$$

This relatively slow convergence rate motivates the use of higher-order methods such as the Milstein scheme.

3.2.2 Milstein method

Definition

The Milstein method improves the Euler–Maruyama scheme by incorporating additional information from the diffusion term [18]. It is given by:

$$X_{n+1} = X_n + a(X_n, t_n)\Delta t + b(X_n, t_n)\Delta B_n + \frac{1}{2}b(X_n, t_n)b_x(X_n, t_n)\big((\Delta B_n)^2 - \Delta t\big), \quad (3.3)$$

where b_x denotes the derivative of b with respect to x.

Convergence of the Milstein method

The Milstein method converges strongly with the order 1, [18] i.e.

$$\mathbb{E}\big[|X(T) - X_N|\big] = O(\Delta t).$$

This higher order of convergence leads to improved precision compared to the Euler–Maruyama method. The algorithm of the Mistein method is given by:

Algorithm 2 Milstein Method

1: **Input:** Drift $a(t, X)$, diffusion $b(t, X)$, derivative $b_x(t, X)$, initial value X_0, initial time t_0, final time T, step size Δt
2: Set $t \leftarrow t_0$
3: Set $X \leftarrow X_0$
4: **while** $t < T$ **do**
5: Generate $\Delta B \sim \mathcal{N}(0, \Delta t)$
6: $X = X + a(t, X)\Delta t + b(t, X)\Delta B + \frac{1}{2}b(t, X)b_x(t, X)\left[(\Delta B)^2 - \Delta t\right]$
7: $t = t + \Delta t$
8: **end while**
9: **Output:** Approximate solution X

3.2.3 Numerical example: Linear stochastic differential equation

Consider the linear SDE:

$$dX(t) = \mu X(t)dt + \sigma X(t)dB(t), \quad X(0) = X_0 = 1. \quad (3.4)$$

Analytic solution

To get an analytic solution to the SDE, we follow the steps:

- Step 1. Apply Itô's formula: We apply Itô's Lemma to the function $f(t, X_t) = \ln(X_t)$:

$$\mathrm{d}f = \left(\frac{\partial f}{\partial t} + \frac{\partial f}{\partial X_t}\mu X_t + \frac{1}{2}\frac{\partial^2 f}{\partial X_t^2}\sigma^2 X_t^2\right)\mathrm{d}t + \frac{\partial f}{\partial X_t}\sigma X_t\,\mathrm{d}B_t.$$

With the relevant partial derivatives:

$$\frac{\partial f}{\partial t} = 0\ ;\ \frac{\partial f}{\partial X_t} = \frac{1}{X_t}\ ;\ \frac{\partial^2 f}{\partial X_t^2} = -\frac{1}{X_t^2}.$$

Substituting these into Itô's formula, yields a simpler SDE for $Y_t = \ln(X_t)$:

$$\mathrm{d}Y_t = \left(\mu - \frac{\sigma^2}{2}\right)\mathrm{d}t + \sigma\,\mathrm{d}B_t.$$

- Step 2. Integrate the New SDE: Integrating both sides from 0 to t:

$$\int_0^t \mathrm{d}Y_s = \int_0^t \left(\mu - \frac{\sigma^2}{2}\right)\mathrm{d}s + \int_0^t \sigma\,\mathrm{d}B_s \iff Y_t - Y_0 = \left(\mu - \frac{\sigma^2}{2}\right)t + \sigma B_t.$$

Using the initial condition $X_0 = 1$, we have $Y_0 = \ln(1) = 0$.

$$Y_t = \left(\mu - \frac{\sigma^2}{2}\right)t + \sigma B_t.$$

- Step 3. Exponentiate to find X_t: Since $Y_t = \ln(X_t)$, we solve for X_t:

$$X_t = e^{Y_t} = e^{\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma B_t}.$$

The exact solution is given by:

$$X(t) = X_0 \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma B(t)\right). \tag{3.5}$$

Euler–Maruyama approximation

Applying the Euler–Maruyama method to (3.4), we obtain:

$$X_{n+1}^{EM} = X_n^{EM} + \mu X_n^{EM}\Delta t + \sigma X_n^{EM}\Delta B_n.$$

The approximation error at time T is measured by:

$$E^{EM} = |X(T) - X_N^{EM}|.$$

The Python code corresponding to this method is given by:

```
# Step 0: Import required libraries
```

```
import numpy as np           # For numerical computations
import matplotlib.pyplot as plt  # For plotting
mu = 1.0# Drift coefficient
sigma = 0.5 # Diffusion coefficient
x0 = 1.0 # Initial condition X_0
T= 1.0 # Final time
N = 1000 # Number of time steps
dt = T / N # Time step size dt
t = np.linspace(0, T, N + 1)  # Time grid
np.random.seed(1)# Fix the random seed for reproducibility

# Brownian increments dB_n ~ N(0, dt)
dB = np.sqrt(dt) * np.random.randn(N)

# Brownian path B_t = cumulative sum of increments
B = np.concatenate(([0.0], np.cumsum(dB)))
# Exact solution of the SDE:

X_exact = x0 * np.exp((mu - 0.5 * sigma**2) * t + sigma * B)

# Initialize array for Euler Maruyama solution
X_EM = np.zeros(N + 1)

# Initialize the array for Euler-Maruyama solution X_EM = np.
    zeros (N + 1) # Set initial value
X_EM[0] = x0
# Time-stepping loop
for n in range(N):
    X_EM[n + 1] = (
        X_EM[n]
        + mu * X_EM[n] * dt         # Drift term
        + sigma * X_EM[n] * dB[n]   # Diffusion term
    )
plt.figure()
plt.plot(t, X_exact, 'k', linewidth=2, label='Exact solution')
plt.plot(t, X_EM, 'r--', label='Euler Maruyama')
plt.xlabel('Time t')
plt.ylabel('X(t)')
plt.title('Euler Maruyama solution superposed with Exact
    Solution')
plt.legend()
```

```
plt.grid(True)
plt.show()
```

Listing 3.1: Euler–Maruyama method for a linear SDE

The simulation result is given in the figure below:

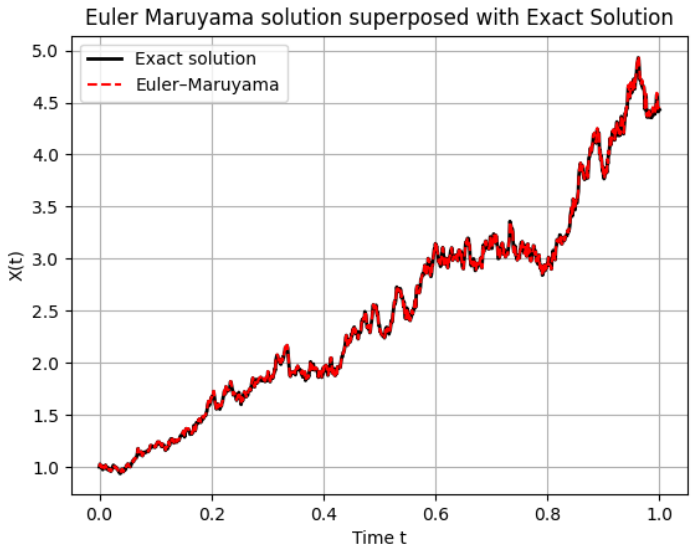

Figure 3.1: Numerical solution by Euler–Maruyama method.

Milstein Approximation

Applying the Milstein method yields:

$$X_{n+1}^M = X_n^M + \mu X_n^M \Delta t + \sigma X_n^M \Delta B_n + \frac{1}{2}\sigma^2 X_n^M \big((\Delta B_n)^2 - \Delta t\big).$$

The corresponding error is:

$$E^M = |X(T) - X_N^M|.$$

The Python code corresponding to this method is given by:

```

import numpy as np
import matplotlib.pyplot as plt
mu = 1.0
sigma = 0.5
x0 = 1.0
T = 1.0
N = 1000
dt = T / N
t = np.linspace(0, T, N + 1)
```

```
np.random.seed(1)
# Brownian increments dB_n ~ N(0, dt)
dB = np.sqrt(dt) * np.random.randn(N)

# Brownian path B_t = cumulative sum of increments
B= np.concatenate(([0.0], np.cumsum(dB)))
# Exact solution of the SDE:
X_exact = x0 * np.exp((mu - 0.5 * sigma**2) * t + sigma * B)
# Initialize array for Milstein solution
X_MIL = np.zeros(N + 1)
# Set initial value
X_MIL[0] = x0
for n in range(N):
    X_MIL[n + 1] = (
        X_MIL[n]
        + mu * X_MIL[n] * dt
        + sigma * X_MIL[n] * dB[n]
        + 0.5 * sigma**2 * X_MIL[n] * (dB[n]**2 - dt) )
plt.figure()
plt.plot(t, X_exact, 'k', linewidth=2, label='Exact solution')
plt.plot(t, X_MIL, 'b-.', label='Milstein')
plt.xlabel('Time t')
plt.ylabel('X(t)')
plt.title('Milstein solution  superposed with Exact Solution')
plt.legend()
plt.grid(True)
plt.show()
```

Listing 3.2: Milstein method for a linear SDE

The simulation result is given in the figure below:

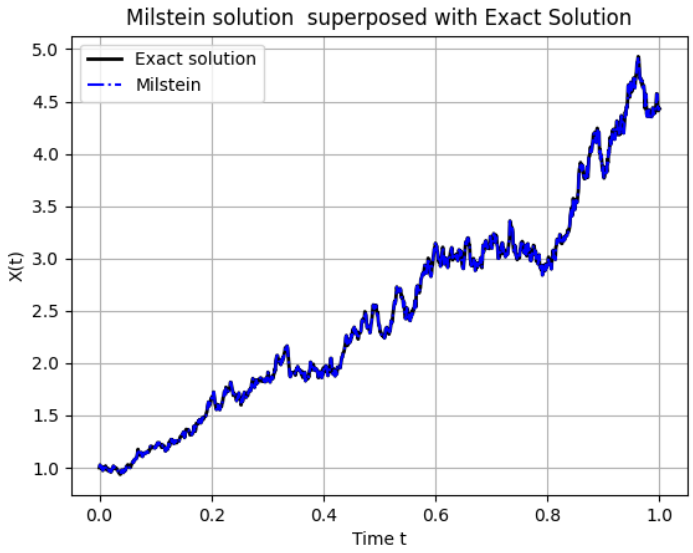

Figure 3.2: Numerical solution by Milstein method.

Comparison of methods

Numerical simulations show that the Milstein method provides significantly higher accuracy than the Euler–Maruyama method for the same time step. While the Euler–Maruyama method is simpler to implement, its convergence order is lower. The Milstein method, although slightly more complex, achieves faster convergence and better approximation of the exact solution. The figure below indicates the error values of both methods.

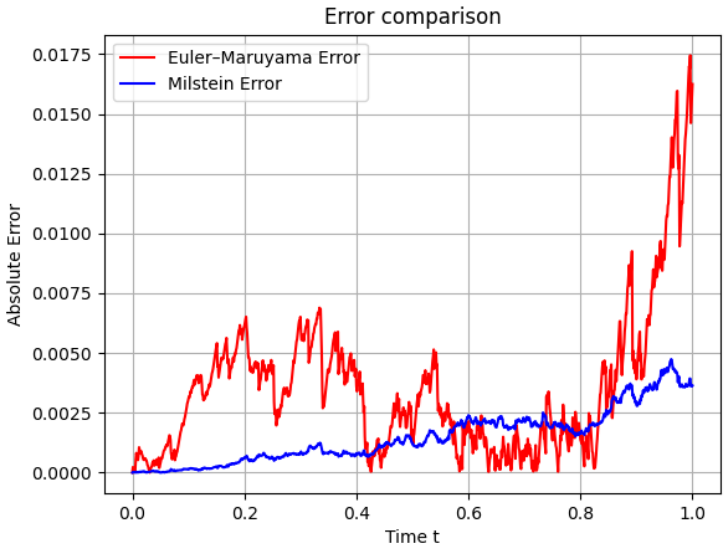

Figure 3.3: The approximation error

Final-time absolute error:

The Milstein scheme is much more accurate (3.64×10^{-3}) than the Euler–Maruyama method (1.62×10^{-2}). This improvement is due to Milstein's greater strong convergence order of 1.0, while Euler–Maruyama's is just 0.5. Adding the second-order Itô corrective term to the Milstein approach makes the discretization error at the last time step much smaller.

Remark 2. *Based on these results, which demonstrate superior accuracy and convergence of the Milstein method. This latter will be implemented as the primary numerical approach in our study of stochastic epidemic models in the next parts of this chapter.*

3.3 Least-Squares Monte Carlo method

The forward-backward stochastic system obtained from the Pontryagin's Maximum Principle involves conditional expectations that cannot be computed analytically. To overcome this difficulty, we employ a Least-Squares Monte Carlo (LSMC) method, which combines Monte Carlo simulation with regression techniques to approximate conditional expectations [15].

3.3.1 General framework

Let X_t be the state process governed by a stochastic differential equation with Lévy jumps:

$$dX_t = b(X_t, u_t, v_t)dt + \sigma(X_t)dB_t + \int_{\mathbb{R}^*} \gamma(X_{t^-}, z)\tilde{N}(dt, dz), \tag{3.6}$$

and let p_t be the adjoint process satisfying a backward stochastic differential equation. Define the σ-algebra generated by the discrete Brownian motion:

$$\mathcal{F}_t = \sigma(\tilde{B}_k : 0 \leq k \leq n),$$

such that $\tilde{B} = \sqrt{\Delta t} \sum_{i \leq n} \xi_i$ and $(\xi_i)_{i>1}$ is an independent and identically distributed sequence of normalised Gaussian random variables.
The backward equation typically has the form

$$p_t = \mathbb{E}\left[p_{t+\Delta t} + \Delta t\, F(X_t, u_t, v_t, q_{t+\Delta t}, p_{t+\Delta t}) \mid \mathcal{F}_t\right], \tag{3.7}$$

where F represents the drivers. The key numerical challenge lies in approximating the conditional expectation.

3.3.2 Time discretization

Let $0 = t_0 < t_1 < \cdots < t_N = T$ with $\Delta t = T/N$.

Representation of the forward model:

The forward dynamics are simulated using the Milstein method with jumps:

$$X_{n+1} = X_n + b(X_n, u_n, v_n)\Delta t + \sigma(X_n)\Delta B_n + \frac{1}{2}\sigma(X_n)\sigma'(X_n)((\Delta B_n)^2 - \Delta t) + \sum_{k=1}^{K_n} \gamma(X_n, Z_{n,k}) \tag{3.8}$$

where:

- $B_n \sim \mathcal{N}(0, \Delta t)$,
- $K_n \sim \text{Poisson}(\lambda \Delta t)$,
- $Z_{n,k}$ are jump amplitudes.

We simulate M independent trajectories:

$$\begin{aligned}
S_{n+1}^{(m)} &= S_n^{(m)} + [u_n^{(m)}\Lambda - \beta(1 - v_n^{(m)})S_n^{(m)}I_n^{(m)} - \mu S_n^{(m)} + lR_n^{(m)}]\Delta t \\
&\quad + \sigma_1 S_n^{(m)}\Delta B_{1,n}^{(m)} + \frac{1}{2}(\sigma_1^2 S_n^{(m)}((\Delta B_{1,n}^{(m)})^2 - \Delta t) + \sum_{j=1}^{J} C_1(z_j)S_n^{(m)}\Delta\tilde{N}_n^{(m)}(z_j) \\
I_{n+1}^{(m)} &= I_n^{(m)} + [\beta(1 - v_n^{(m)})S_n^{(m)}I_n^{(m)} - (\gamma + \mu + \epsilon)I_n^{(m)}]\Delta t \\
&\quad + \sigma_2 I_n^{(m)}\Delta B_{2,n}^{(m)} + \frac{1}{2}(\sigma_1^2 I_n^{(m)}((\Delta B_{1,n}^{(m)})^2 - \Delta t) + \sum_{j=1}^{J} C_2(z_j)I_n^{(m)}\Delta\tilde{N}_n^{(m)}(z_j) \\
R_{n+1}^{(m)} &= R_n^{(m)} + [\gamma I_n^{(m)} - (\mu + l)R_n^{(m)}]\Delta t \\
&\quad + \sigma_3 R_n^{(m)}\Delta B_{3,n}^{(m)} + \frac{1}{2}(\sigma_1^2 R_n^{(m)}((\Delta B_{1,n}^{(m)})^2 - \Delta t) + \sum_{j=1}^{J} C_3(z_j)R_n^{(m)}\Delta\tilde{N}_n^{(m)}(z_j)
\end{aligned}$$

where $\Delta B_{i,n}^{(m)} \sim \mathcal{N}(0, \Delta t)$ and $\Delta\tilde{N}_n^{(m)}(z_j)$ are the increments of the compensated Poisson process.

Representation by conditional expectation of backward equations:

Taking the conditional expectation given $\mathcal{F}_{t_n}$ of the discretized backward equations, we obtain the following expressions for the adjoint model:

$$p_1(t_n) = \mathbb{E}\left[p_1(t_{n+1}) + \Delta t\mathcal{G}_1(t_n) \mid \mathcal{F}_{t_n}\right] \tag{3.9}$$

$$p_2(t_n) = \mathbb{E}\left[p_2(t_{n+1}) + \Delta t\mathcal{G}_2(t_n) \mid \mathcal{F}_{t_n}\right] \tag{3.10}$$

$$p_3(t_n) = \mathbb{E}\left[p_3(t_{n+1}) + \Delta t\mathcal{G}_3(t_n) \mid \mathcal{F}_{t_n}\right] \tag{3.11}$$

where the drivers $\mathcal{G}_i$ are given by:

$$\begin{aligned}\mathcal{G}_1(t_n) &= p_1(t_{n+1})\beta(1-v_n^{(m)})I_n + p_1(t_{n+1})\mu - p_2(t_{n+1})\beta(1-v_n^{(m)})I_n \\ &\quad - \sigma_1 q_{11}(t_{n+1}) - \int_{\mathbb{R}^*} C_1(z) r_1(t_{n+1}, z)\pi(dz) \qquad (3.12)\\ \mathcal{G}_2(t_n) &= -1 + p_1(t_{n+1})\beta(1-v_n^{(m)})S_n - p_2(t_{n+1})\beta(1-v_n^{(m)})S_n + p_2(t_{n+1})(\gamma+\mu+\epsilon) \\ &\quad - p_3(t_{n+1})\gamma - \sigma_2 q_{22}(t_{n+1}) - \int_{\mathbb{R}^*} C_2(z) r_2(t_{n+1}, z)\pi(dz) \qquad (3.13)\\ \mathcal{G}_3(t_n) &= -p_1(t_{n+1})l + p_3(t_{n+1})(\mu+l) - \sigma_3 q_{33}(t_{n+1}) - \int_{\mathbb{R}^*} C_3(z) r_3(t_{n+1}, z)\pi(dz) \qquad (3.14)\end{aligned}$$

Regression approximation:

We approximate adjoint processes and controls as linear combinations of K basis functions $\{\phi_k\}_{k=1}^K$:

$$p_i(t_n, S, I, R) \approx \sum_{k=1}^{K} \alpha_{i,k}(t_n)\phi_k(S, I, R) \qquad (3.15)$$

$$q_{ij}(t_n, S, I, R) \approx \sigma_i(S, I, R) \sum_{k=1}^{K} \alpha_{i,k}(t_n) \frac{\partial \phi_k}{\partial X_j}(S, I, R) \qquad (3.16)$$

$$r_i(t_n, z, S, I, R) \approx C_i(z) \sum_{k=1}^{K} \alpha_{i,k}(t_n)\phi_k(S, I, R) \qquad (3.17)$$

where X_j represents the corresponding state variables (S for $j = 1$, I for $j = 2$, R for $j = 3$). At each time step, the coefficients $\alpha(t_n)$ are obtained by solving the least-squares problem:

$$\alpha(t_n) = \arg\min_{\alpha \in \mathbb{R}^3 \times \mathbb{R}^K} \sum_{m=1}^{M} \left| \sum_{k=1}^{K} \alpha_k \phi_k(X_n^{(m)}) - b_n^{(m)} \right|^2, \qquad (3.18)$$

where

$$b_n^{(m)} = p_{n+1}^{(m)} + \Delta t\, \mathcal{G}(X_n^{(m)}, u_n^{(m)}, v_n^{(m)}, p_{n+1}^{(m)}). \qquad (3.19)$$

In matrix form, this leads to:

$$(A_n^\top A_n + \lambda I)\alpha = A_n^\top b_n, \qquad (3.20)$$

$$\alpha(t_n) = (A_n^\top A_n + \lambda I)^{-1} A_n^\top b_n, \qquad (3.21)$$

where $A_n = \phi_k(X_n^{(m)})_{m=1,\dots,M;k=1,\dots,K}$.

Choice of basic functions:

We take as radial basis functions (RBF), the following formula:

$$\phi_k(S, I, R) = \frac{1}{\delta\sqrt{2\pi}} \exp\left(-\frac{(S-\mu_{S,k})^2 + (I-\mu_{I,k})^2 + (R-\mu_{R,k})^2}{2\delta^2}\right)$$

The centers $(\mu_{S,k}, \mu_{I,k}, \mu_{R,k})$ can be chosen adaptively using the positions of the trajectories at each time step.

Numerical treatment of jump terms:

The integrals over the jump space are discretized as follows:

$$\int_{\mathbb{R}^*} C_i(z) r_i(t, z) \pi(dz) \approx \sum_{j=1}^{J} C_i(z_j) r_i(t, z_j) \pi_j$$

where $\{z_j\}_{j=1}^{J}$ are representative jump amplitudes and π_j are the corresponding intensities.

3.3.3 Algorithm of the LSMC method

The LSMC procedure can be summarized as follows:

1. Choose time discretization parameters N and Δt.
2. Generate M forward sample paths.
3. Initialize the terminal adjoint variable p_N.
4. For $n = N - 1$ down to 2:
   - Construct regression matrix A,
   - Solve the least-squares problem,
   - Update p_n,
   - Update controls u_n and v_n.
5. Iterate until convergence.

The adapted Least Square Monte Carlo algorithm is given by:

Algorithm 3 Adapted LSMC for SIRS model with two controls and jumps

Forward initialization: Set $S_0^{(m)} = 0.8,\ I_0^{(m)} = 0.15,\ R_0^{(m)} = 0.05.$

Backward initialization: Set $p_{1,N}^{(m)} = 0, p_{2,N}^{(m)} = 1, p_{3,N}^{(m)} = 0,\ q_{ij,N}^{(m)} = 0,\ r_{i,N}^{(m)} = 0$ for $i, j = 1, 2, 3.$
for $z = 1$ to N-iteration **do**
Forward simulation: Generate M paths $(S_n^{(m)}, I_n^{(m)}, R_n^{(m)})$ using the Milstein scheme including jump terms.
Backward simulation: for $n = N - 1$ to 2 **do**

- For each $i = 1, 2, 3$:
  - Build matrix $A_{i,n}$ with entries $\phi_k(S_n^{(m)}, I_n^{(m)}, R_n^{(m)})$.
  - Build data vector $b_{i,n}$ using the appropriate driver $\mathcal{G}_i$ evaluated at time t_{n+1}.
  - Find $\hat{\alpha}_i(t_n) = (A_{i,n}^T A_{i,n} + \lambda I)^{-1} A_{i,n}^T b_{i,n}$ (with regularisation).
  - Update

$$p_i(t_n)^{(m)} = \sum_k \hat{\alpha}_{i,k}(t_n)\phi_k(S_n^{(m)}, I_n^{(m)}, R_n^{(m)}),$$
$$q_{ij}(t_n)^{(m)} = \sigma_i(S, I, R) \sum_k \hat{\alpha}_{i,k}(t_n) \frac{\partial \phi_k}{\partial X_j}(S_n^{(m)}, I_n^{(m)}, R_n^{(m)}),$$
$$r_i(t_n, z_j)^{(m)} = C_i(z_j) \sum_k \hat{\alpha}_{i,k}(t_n)\phi_k(S_n^{(m)}, I_n^{(m)}, R_n^{(m)}).$$

end for
end for

- Compute the optimal controls:

$$u_n^{(m)} = \max\left(0, \min\left(1, -\frac{\Lambda}{\gamma_1} p_1(t_n)^{(m)}\right)\right)$$
$$v_n^{(m)} = \max\left(0, \min\left(1, \frac{\beta S_n^{(m)} I_n^{(m)}}{\gamma_2}\left(p_2(t_n)^{(m)} - p_1(t_n)^{(m)}\right)\right)\right)$$

3.3.4 Convergence and complexity of LSMC method

The total computational complexity is dominated by solving the regression system at each time step. If K denotes the number of basis functions and M the number of Monte Carlo trajectories, the overall complexity is approximately [13]:

$$\mathcal{O}(NK^2 + NMK).$$

Under suitable regularity conditions, the LSMC estimator converges as:

$$\text{Error} \sim \mathcal{O}\left(\frac{1}{\sqrt{M}}\right) + \mathcal{O}(\Delta t).$$

3.3.5 Advantages of LSMC method

The LSMC method offers several advantages:

- It avoids solving high-dimensional partial differential equations.
- It is well-suited for stochastic control problems with jump dynamics.
- The computational cost grows polynomially with the state dimension.

3.4 Numerical simulation of the controlled epidemic model with jumps

3.4.1 Simulation of stochastic SIRS model with jump

Since the differential equation system governing the model is difficult to solve analytically, we used the Milstein method described in the previous section to obtain numerical solutions of the stochastic system (1.9) by simulating multiple possible trajectories. The results of these simulations are illustrated in the following figure:

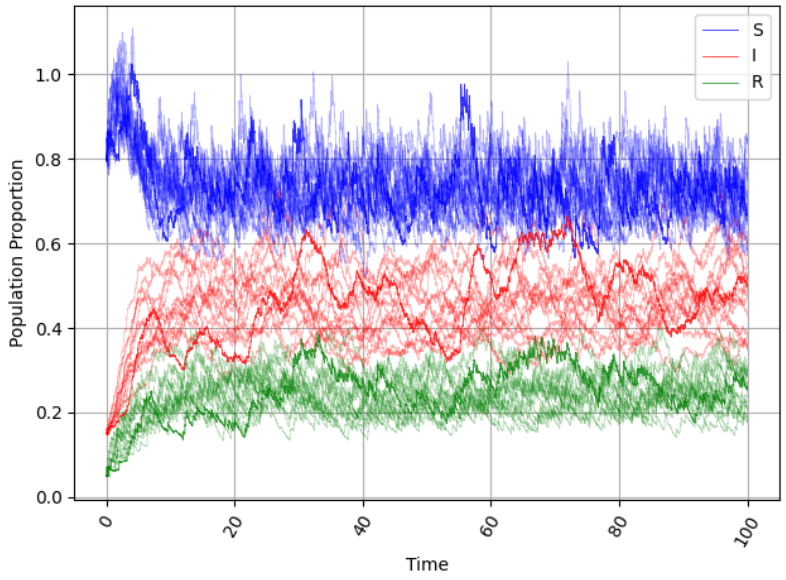

Figure 3.4: Solutions of system (1.9). The initial condition is $(S_0 = 0.8, I_0 = 0.15, R_0 = 0.05)$, with the parameter values $\beta = 0.8, \mu = 0.3, \Lambda = 0.4, \epsilon = 0.1, \gamma = 0.2, l = 0.2$.

3.4.2 The Lévy jumps effect on the epidemic dynamics

To illustrate the Lévy jumps on the model trajectories, we show one path in the following figure:

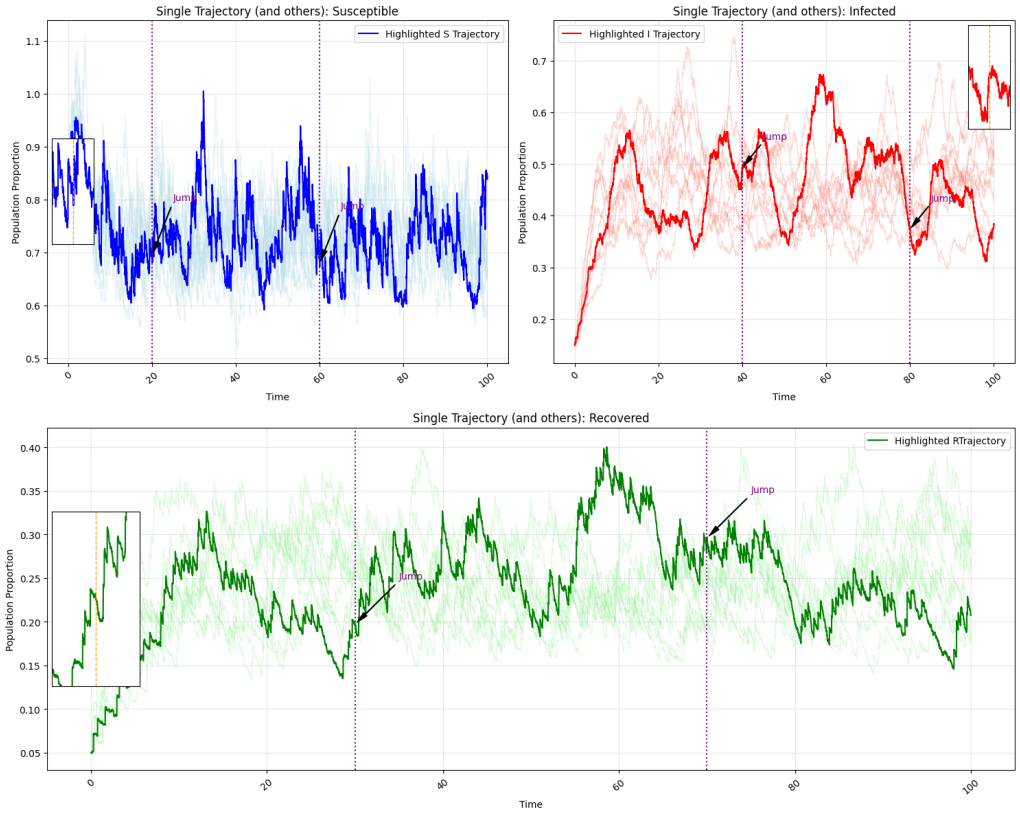

Figure 3.5: Paths of S(t), I(t) and R(t) for the stochastic model (1.9) (one trajectory).

Interpretation:

The jumps observed in the magnified segments of the graph are interpreted as real life discontinuities that cannot be modeled by a smooth model, like for example:

- Super-spreader event: A vertical jump in the infected curve indicates a real life situation such as a concert, an organized sport, or a festival that results in a surge of new cases almost instantaneously.
- Migration: The abrupt decline in the susceptible population may be a consequence of migration that takes place in a matter of days, which changes the density of vulnerable individuals overnight.
- Policy Changes: Another example where the graph's jumps can illustrate an abrupt change in reality is when there is an urgent change in the policy. Such changes could be the result of ending a lockdown or the arrival of a new variant.

So, the presence of these jumps in the graph makes our model more realistic.

3.4.3 Numerical Approximation of optimal control for SIRS models

We use the Least-Squares Monte Carlo Method mentioned in the previous section based on the Milstein method to solve the adjoint equations (2.7) and obtain the optimal controls which are shown the following figure:

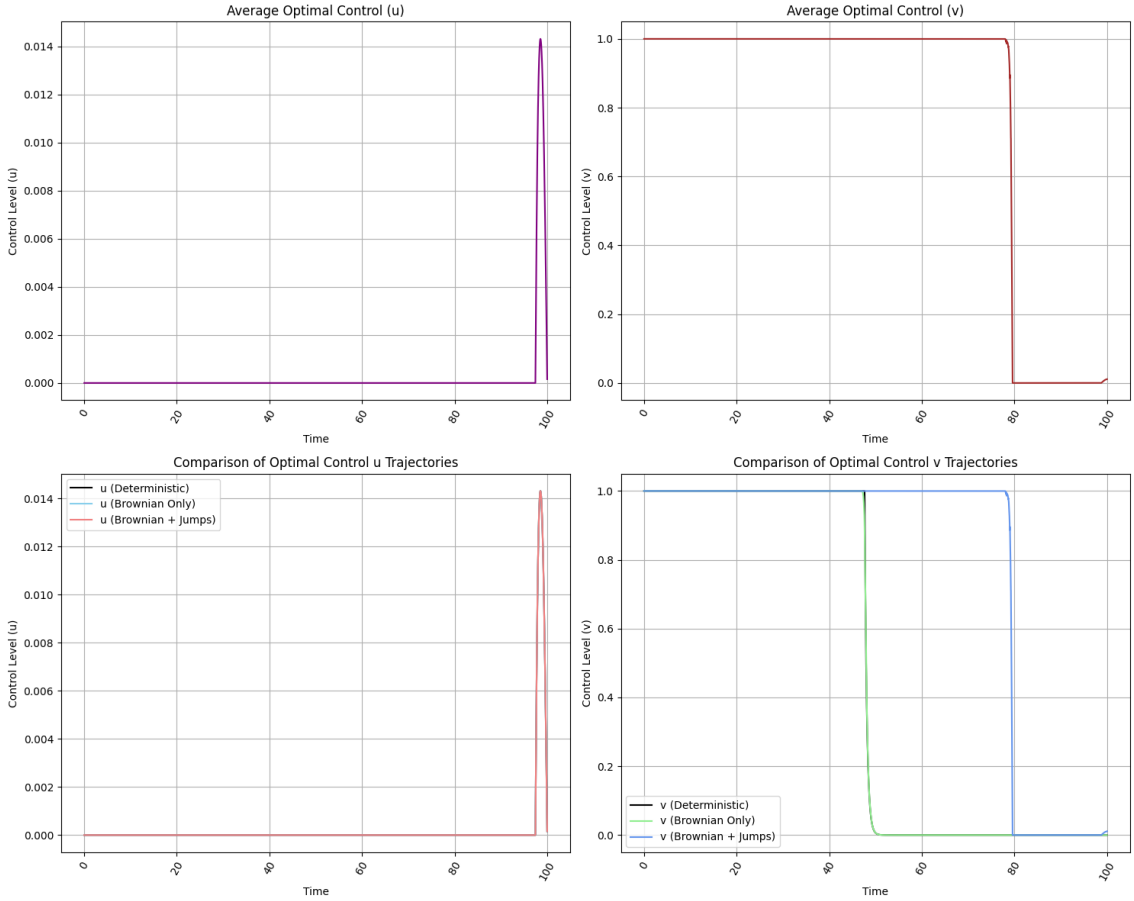

Figure 3.6: Stochastic optimal control trajectories.

Figure 3.6 represents the evolution of optimal control trajectories over time for deterministic and stochastic cases. At the beginning, both controls are maximum (borders closed and strong preventive measures. Over time the control strategy relaxed as the epidemic is being reduced.

The figure also illustrates incorporating jumps through a Lévy process significantly enhances the effectiveness of the optimal control strategy compared to a purely deterministic framework. In the deterministic case, the control v (social distancing, vaccination and masks) follows a smooth path that gradually declines over time. However, when Lévy jumps are introduced representing abrupt, large-scale events such as super-spreader outbreaks or the emergence of a new variant, the optimal control reacts immediately and forcefully. The figure shows that the stochastic trajectory with jumps maintains a per-

sistently higher level of v than its deterministic counterpart. This elevated control value provides a buffer that allows the system to respond more precisely and robustly when jumps in the infection rate occur (by keeping v sufficiently high). The controller is always ready to mitigate sudden surges in transmission, leading to more accurate and resilient disease management. Thus, the visual comparison confirms that adding Lévy jumps is not merely a theoretical refinement, but it directly enables tighter and more adaptive control actions under real-world uncertainty.

Mathematically, immigration control acts as a long-term strategic buffer. Unlike preventive measures v that must react tactically to daily spikes. The control u focuses on the "marginal value of control immigration". It remains stable because it aims to reduce the overall pool of susceptible individuals over time, building institutional resilience that "filters out" the immediate volatility caused by random stochastic shocks.

3.4.4 Impact of control measures on SIRS model dynamics

The figure 3.7 illustrates the effect of applying optimal control to the SIRS model. In the upper right graph, the evolution of the S, I, R populations is presented without control (uncontrolled case), while the upper left graph shows the behavior of the system after applying control strategies (controlled case). The results represent the effect of the controls, which play an important role in reducing the number of infected individuals through preventive measures such as social distancing, wearing masks, and vaccination. These strategies help decreasing the transmission rate of the disease and limit its spread over time.

The bottom plot compares the infected numbers from both scenarios (controlled and uncontrolled cases). It can be clearly observed that the application of control significantly reduces the proportion of infected individuals, demonstrating the effectiveness of the proposed strategy in controlling the epidemic.

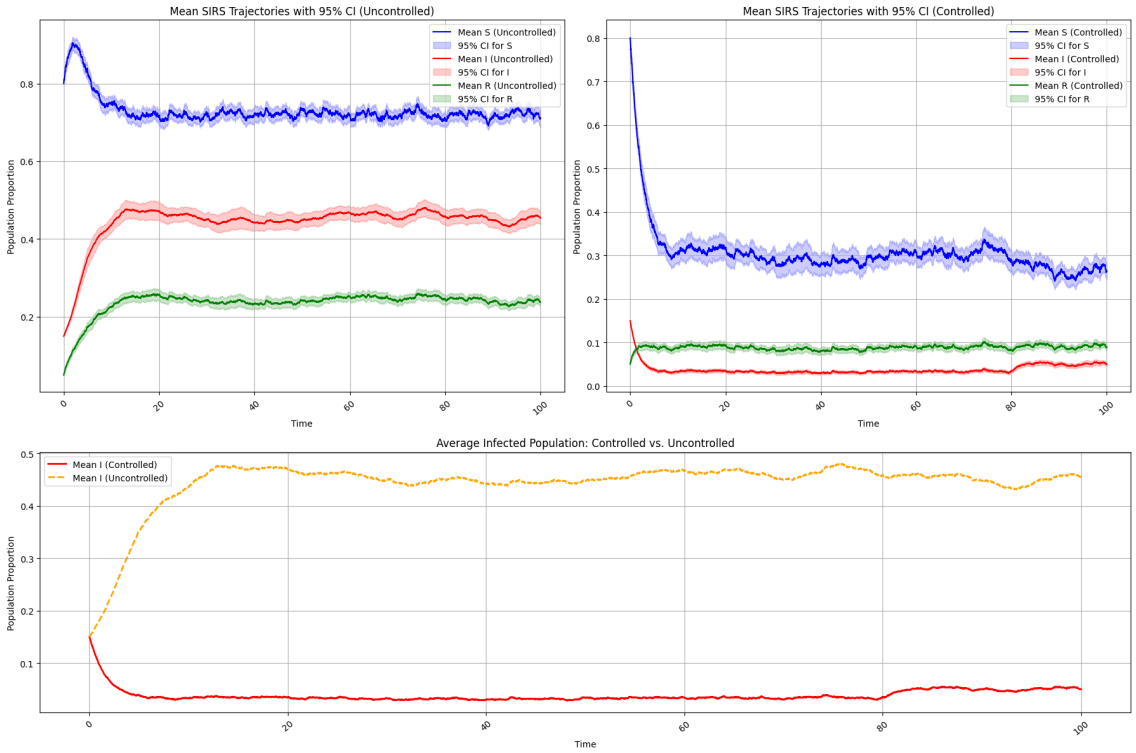

Figure 3.7: Optimal control effects on SIRS Model changes

In addition, as shown in figure 3.8, the control variable u contributes to reducing the number of susceptible individuals by regulating the migration rate, which further helps to limit the spread of the disease and improve the overall stability of the model.

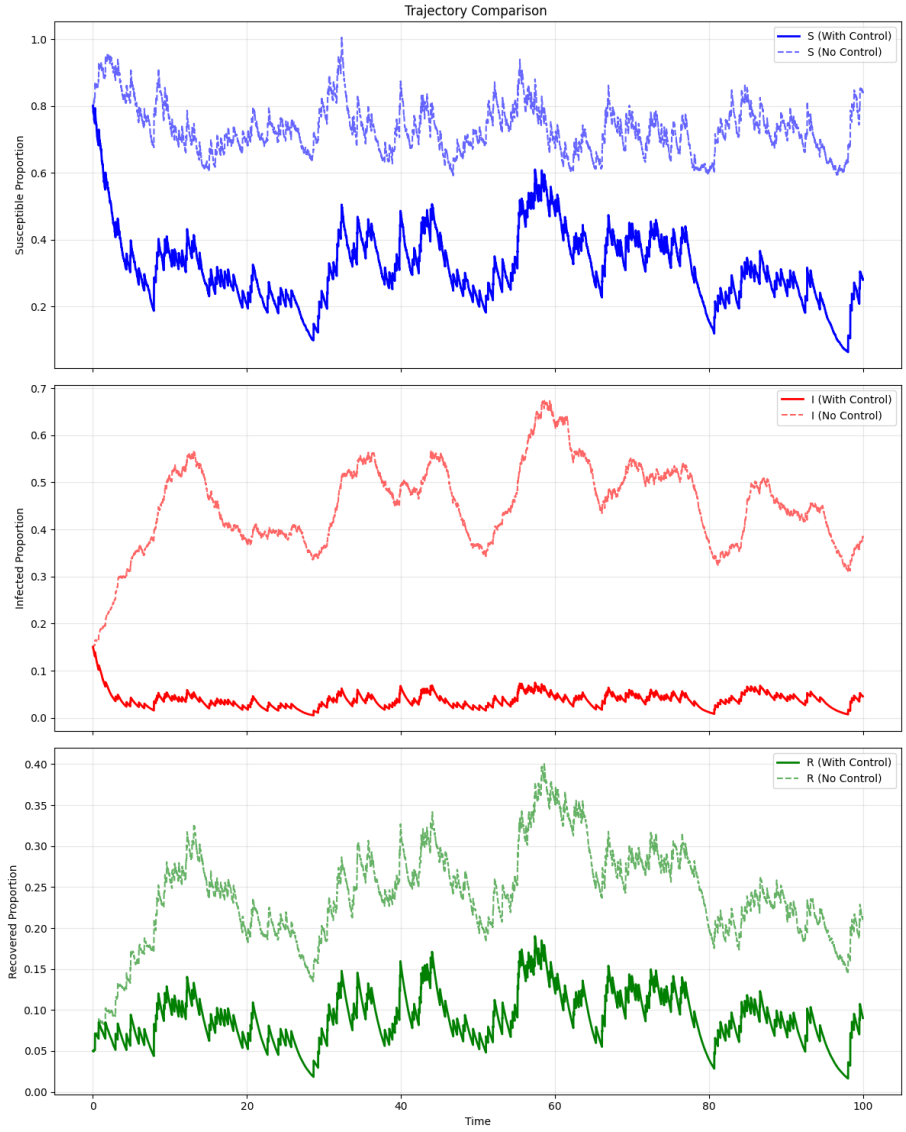

Figure 3.8: Single stochastic realization highlighting optimal control effect on outbreak.

3.4.5 Effect of different levels of randomness on the epidemic model

We show in the figure below how different levels of randomness might change the SIRS trajectories. The black line is a deterministic model and the light (blue, green, gold) lines are continuous random fluctuations (Brownian motion). The red, cornflower blue and medium orchid lines show the effect of the jumps, leading to more realistic and volatile scenarios of disease spread over the population.

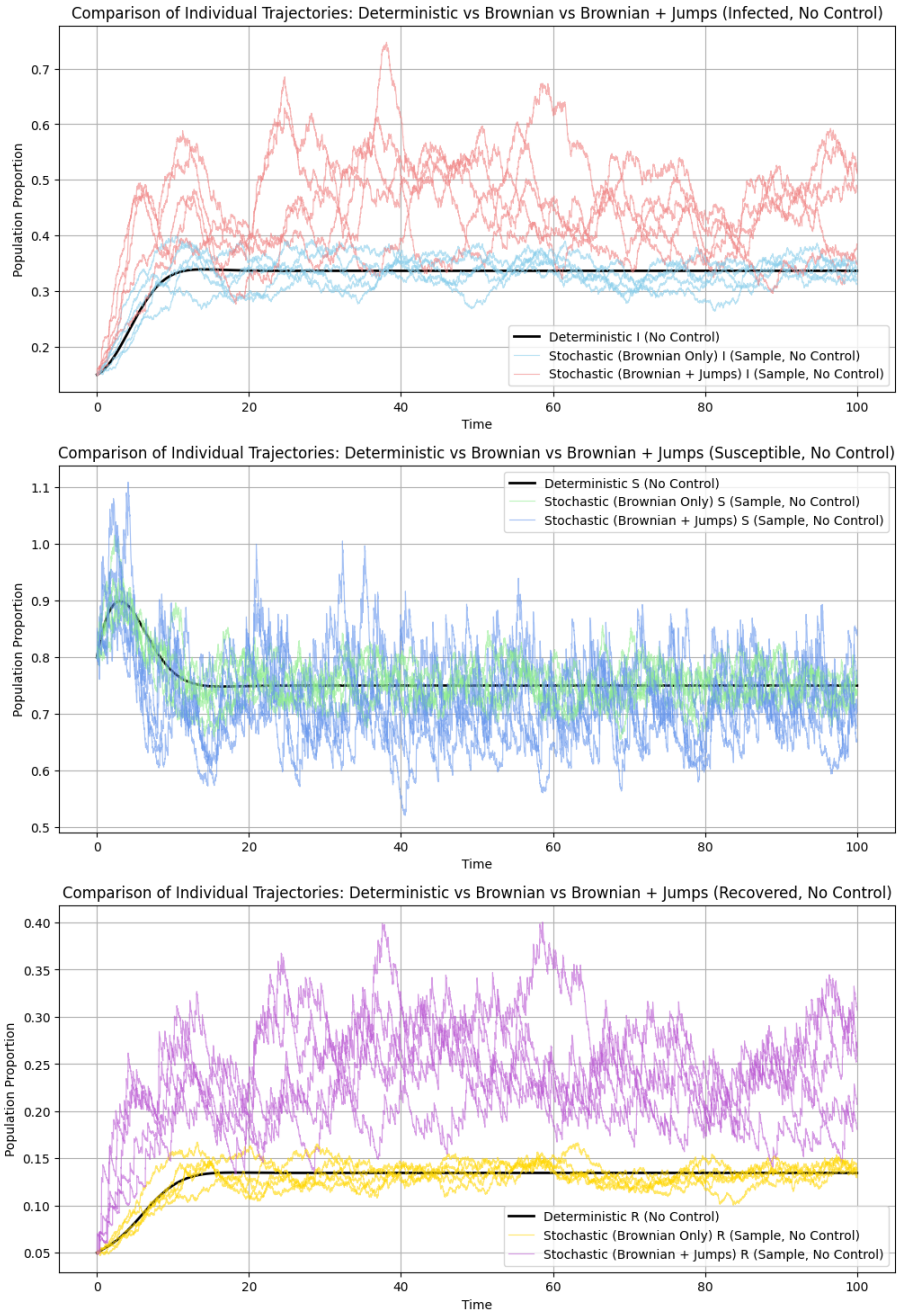

Figure 3.9: Impact of randomness (deterministic vs. stochastic trajectories (without and with jumps).)

3.4.6 Stochastic threshold for the SIRS model with Lévy jumps

Consider the SIRS model with Lévy jumps. Following [34], we define the stochastic threshold without control:

$$\tilde{R}_0 = R_0 - \frac{\beta_2}{\gamma + \mu + \epsilon}$$

where $R_0 = \dfrac{\beta\Lambda}{\mu(\gamma+\mu+\epsilon)}$ is the basic reproduction number of the deterministic model (1.5), and

$$\beta_2 = \frac{1}{2}\sigma_2^2 + \int_{\mathbb{R}^*} \big[C_2(u) - \ln(1+C_2(u))\big]\pi(du)$$

accounts for the white noise σ_2 and Lévy jumps C_2. Then:

- If $\tilde{R}_0 < 1$: the disease goes extinct almost surely.
- If $\tilde{R}_0 > 1$: the disease persists in mean.

With controls (immigration control u and transmission reduction v), the controlled threshold becomes:

$$\tilde{R}_0^{\text{ctrl}} = R_0^{\text{ctrl}} - \frac{\beta_2}{\gamma+\mu+\epsilon}, \quad R_0^{\text{ctrl}} = \frac{\beta(1-v)\,(u\Lambda)}{\mu(\gamma+\mu+\epsilon)}.$$

The table below summarizes the reproduction numbers under various control strategies:

Scenario	u(immigration control)	v(transmission control)	R_0 (Deter.)	$\tilde{R}_0$ (Stoch.)
No Control	1.0000	0.0000	1.7777	1.7771
With Optimal Control (Actual Mean)	0.0002	0.7931	0.000	0.000
Immigration control Only	0.7000	0.0000	1.2444	1.2437
transmission control Only	1.0000	0.2500	1.3333	1.3326
Combined (Example)	0.800	0.200	1.1377	1.13712

Table 3.1: Threshold comparison in different control scenarios.

Impact and efficiency analysis of optimal control:
The calculated of optimal control strategy is highly effective by reducing the immigration factor u to near zero and maintaining transmission control $v \approx 0.79$. Stochastic vs deterministic: In all scenarios, $\tilde{R}_0$ is slightly lower than R_0. This confirms that environmental noise (white noise and Lévy jumps) actually assists in destabilizing the infected population, providing a "noise-induced extinction" advantage compared to the deterministic model.
Individual vs combined control: Neither immigration control ($u = 0.7$) nor transmission control ($v = 0.25$) alone is sufficient to bring the reproduction number less than 1. This highlights the necessity of the Multi-Control approach optimized by the Least-Squares Monte Carlo method mentioned in Chapter 2.

3.5 Python code for the SIRS epidemic model

The Python code that we have programmed to perform our simulations of the controlled SIRS model is given bellow:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import solve
import pandas as pd
# Parameters
beta = 0.8      #the transmission rate
l=0.2           #the rate of immunity loss
mu = 0.3         #death rate
epsilon = 0.1   #disease-induced death rate
gamma = 0.2      #the recovery rate
imm_rate =0.4   # recruitment rate
gamma1 = 1.0     #cost weight for u
gamma2 = 1.0     #cost weight for v
sigma = np.array([0.1, 0.1, 0.1])    # Volatilities
lambda_jump = 2       #  Jump Intensity
C = np.array([0.05, 0.01, 0.02])      #  Constant Amplitudes  [
    C_S, C_I, C_R]

T = 100
dt = 0.1                # Time step
M = 50                  # Number of Monte Carlo trajectories
K = 3                    # Number of basis functions
reg_lambda = 1e-4        #  Ridge regularization
N = int(T / dt)          # Number of time steps

S0, I0, R0 = 0.8, 0.15, 0.05    # Initial Conditions

tol = 1e-3               # Tolerance for convergence
max_iter = 10            #  Maximum  Number of iteration
ep=0.01

#    adaptive RBF Functions
def compute_centers(S, I, R):

    min_S, max_S = np.min(S), np.max(S)
    min_I, max_I = np.min(I), np.max(I)
    min_R, max_R = np.min(R), np.max(R)
    centers = np.array([
        [min_S+ep, min_I+ep, min_R+ep],
        [max_S+ep, max_I+ep, max_R+ep],
```

```
        [(min_S+max_S)/2+ep, (min_I+max_I)/2+ep, (min_R+max_R)/2+ep]
    ])
    return centers

def phi(X, centers, width=10):

    M = X.shape[0]
    K = centers.shape[0]
    Phi = np.zeros((M, K))
    for k in range(K):
        diff = X - centers[k, :]
        r2 = np.sum(diff**2, axis=1)
        Phi[:, k] = (1/(width*np.sqrt(np.pi*2)))*np.exp(-r2 / (2 * width**2))
    return Phi

def grad_phi(X, centers, width=10):
    M = X.shape[0]
    K = centers.shape[0]
    grad_S = np.zeros((M, K))
    grad_I = np.zeros((M, K))
    grad_R = np.zeros((M, K))
    for k in range(K):
        diff = X - centers[k, :]
        r2 = np.sum(diff**2, axis=1)
        phi_k = (1/(width*np.sqrt(np.pi*2)))*np.exp(-r2 / (2 * width**2))
        grad_S[:, k] = -(1/(width*np.sqrt(np.pi*2)))*(diff[:, 0] / width**2 )* phi_k
        grad_I[:, k] = -(1/(width*np.sqrt(np.pi*2)))*(diff[:, 1] / width**2) * phi_k
        grad_R[:, k] = -(1/(width*np.sqrt(np.pi*2)))*(diff[:, 2] / width**2) * phi_k
    return grad_S, grad_I, grad_R
#  Forward simulation  with given controls
def forward_simulation(u_func, v_func):
    S = np.zeros((M, N+1))
    I = np.zeros((M, N+1))
    R = np.zeros((M, N+1))
    S[:, 0] = S0
```

```
    I[:, 0] = I0
    R[:, 0] = R0

    # Generate the jumps
    np.random.seed(123)
    jump_inc = np.zeros((M, N, 3))
    for m in range(M):
        n_jumps = np.random.poisson(lambda_jump * T)
        jump_times = np.random.uniform(0, T, n_jumps)
        for t_jump in jump_times:
            idx = int(np.floor(t_jump / dt))
            if idx < N:
                jump_inc[m, idx, :] += C

    for n in range(N-1):
        S_n = S[:, n]
        I_n = I[:, n]
        R_n = R[:, n]
        # Control at instant  n
        u = u_func(n*dt, S_n, I_n, R_n)
        v = v_func(n*dt, S_n, I_n, R_n)
        # Brownian increments
        dW1 = np.sqrt(dt) * np.random.randn(M)
        dW2 = np.sqrt(dt) * np.random.randn(M)
        dW3 = np.sqrt(dt) * np.random.randn(M)

        # Drift
        drift_S = u * imm_rate - beta*(1-v) * S_n * I_n - mu *
            S_n+ l*R_n - lambda_jump * C[0] * S_n
        drift_I = beta*(1-v) * S_n * I_n - (gamma + mu +
           epsilon) * I_n - lambda_jump * C[1] * I_n
        drift_R = gamma * I_n - mu * R_n -l*R_n - lambda_jump
           * C[2] * R_n

        # Diffusion
        diff_S = sigma[0] * S_n
        diff_I = sigma[1] * I_n
        diff_R = sigma[2] * I_n

        # Milstein
        milstein_S = 0.5 * sigma[0]**2 * S_n * (dW1**2 - dt)
```

```
        milstein_I = 0.5 * sigma[1]**2 * I_n * (dW2**2 - dt)
        milstein_R = 0.5 * sigma[2]**2 * R_n * (dW3**2 - dt)

        # jump
        jump_S = jump_inc[:, n, 0]
        jump_I = jump_inc[:, n, 1]
        jump_R = jump_inc[:, n, 2]

        S[:, n+1] = S_n + drift_S * dt + diff_S * dW1 +
            milstein_S + jump_S
        I[:, n+1] = I_n + drift_I * dt + diff_I * dW2 +
            milstein_I + jump_I
        R[:, n+1] = R_n + drift_R * dt + diff_R * dW3 +
            milstein_R + jump_R

        S[:, n+1] = np.maximum(S[:, n+1], 0)
        I[:, n+1] = np.maximum(I[:, n+1], 0)
        R[:, n+1] = np.maximum(R[:, n+1], 0)

    return S, I, R
#Backward Simulation (LSMC) to calculate adjoints
def backward_simulation(S, I, R):

    M, Np1 = S.shape
    N = Np1 - 1
    p1 = np.zeros((M, N+1))
    p2 = np.zeros((M, N+1))
    p3 = np.zeros((M, N+1))

    # Terminal conditions
    p1[:, N] = 0
    p2[:, N] = 1
    p3[:, N] = 0

    alpha1 = np.zeros((K, N+1))
    alpha2 = np.zeros((K, N+1))
    alpha3 = np.zeros((K, N+1))

    for n in range(N-1, 0, -1):
        S_n = S[:, n]
        I_n = I[:, n]
```

```
151         R_n = R[:, n]
152 
153         # Adaptive Centers
154         centers = compute_centers(S_n, I_n, R_n)
155         X_n = np.column_stack((S_n, I_n, R_n))
156         A = phi(X_n, centers)
157 
158         # Values at the next step  (for the drivers)
159         p1_np1 = p1[:, n+1]
160         p2_np1 = p2[:, n+1]
161         p3_np1 = p3[:, n+1]
162 
163         # Calculation of  q and r  at the next step
164         if n < N-1:
165             centers_np1 = compute_centers(S[:, n+1], I[:, n
                +1], R[:, n+1])
166             X_np1 = np.column_stack((S[:, n+1], I[:, n+1], R
                [:, n+1]))
167             A_np1 = phi(X_np1, centers_np1)
168             grad_S_np1, grad_I_np1, grad_R_np1 = grad_phi(
                X_np1, centers_np1)
169 
170 
171             q1_np1 = sigma[0] * S[:, n+1] * np.sum(alpha1[:, n
                +1] * grad_S_np1, axis=1)
172             q2_np1 = sigma[1] * I[:, n+1] * np.sum(alpha2[:, n
                +1] * grad_I_np1, axis=1)
173             q3_np1 = sigma[2] * R[:, n+1] * np.sum(alpha3[:, n
                +1] * grad_R_np1, axis=1)
174 
175             r1_np1 = C[0] * (A_np1 @ alpha1[:, n+1])
176             r2_np1 = C[1] * (A_np1 @ alpha2[:, n+1])
177             r3_np1 = C[2] * (A_np1 @ alpha3[:, n+1])
178         else:
179             q1_np1 = np.zeros(M)
180             q2_np1 = np.zeros(M)
181             q3_np1 = np.zeros(M)
182             r1_np1 = np.zeros(M)
183             r2_np1 = np.zeros(M)
184             r3_np1 = np.zeros(M)
185 
```

```
        # Approximation of  v for the driver
        v_approx = (S_n*I_n / gamma2) * (p1_np1 - p2_np1)
        v_approx = np.clip(v_approx, 0, 1)

        #jump integrals
        int_C1r1 = lambda_jump * C[0] * r1_np1
        int_C2r2 = lambda_jump * C[1] * r2_np1
        int_C3r3 = lambda_jump * C[2] * r3_np1

        # Drivers
        G1 = (p1_np1 * beta * I_n*(1-v_approx) + p1_np1 * mu
           -
              p2_np1 * beta*(1-v_approx) * I_n -sigma[0] *
                 q1_np1 -int_C1r1)

        G2 = (-1 +p1_np1 * beta*(1-v_approx) * S_n - p2_np1 *
           beta*(1-v_approx) * S_n +
              p2_np1 * (gamma + mu + epsilon) -p3_np1 * gamma
                 -
              sigma[1] * q2_np1 -int_C2r2)

        G3 = (-p1_np1*l+ p3_np1 * (mu +v_approx)-sigma[2] *
           q3_np1 - int_C3r3)

        b1 = p1_np1 + dt * G1
        b2 = p2_np1 + dt * G2
        b3 = p3_np1 + dt * G3

        #  Ridge regression
        AtA = A.T @ A + reg_lambda * np.eye(K)
        alpha1[:, n] = solve(AtA, A.T @ b1, assume_a='pos')
        alpha2[:, n] = solve(AtA, A.T @ b2, assume_a='pos')
        alpha3[:, n] = solve(AtA, A.T @ b3, assume_a='pos')

        # Reconstruction  of p
        p1[:, n] = A @ alpha1[:, n]
        p2[:, n] = A @ alpha2[:, n]
        p3[:, n] = A @ alpha3[:, n]

    # At  the first time n=0
    p1_mean_time = np.mean(p1[:, 1:], axis=1)    # shape (M,)
```

```
    p2_mean_time = np.mean(p2[:, 1:], axis=1)
    p3_mean_time = np.mean(p3[:, 1:], axis=1)
    p1[:, 0] = p1_mean_time
    p2[:, 0] = p2_mean_time
    p3[:, 0] = p3_mean_time

    # Regression for coefficients at n=0 (using these
       path_specific targets)
    S_0 = S[:, 0]
    I_0 = I[:, 0]
    R_0 = R[:, 0]
    centers_0 = compute_centers(S_0, I_0, R_0)
    X_0 = np.column_stack((S_0, I_0, R_0))
    A0 = phi(X_0, centers_0)

    b1_0 = p1_mean_time   # already a vector of length M
    b2_0 = p2_mean_time
    b3_0 = p3_mean_time

    AtA0 = A0.T @ A0 + reg_lambda * np.eye(K)
    alpha1[:, 0] = solve(AtA0, A0.T @ b1_0, assume_a='pos')
    alpha2[:, 0] = solve(AtA0, A0.T @ b2_0, assume_a='pos')
    alpha3[:, 0] = solve(AtA0, A0.T @ b3_0, assume_a='pos')
    return alpha1, alpha2, alpha3, p1, p2, p3
# Iterates until convergence
def u0(t, S, I, R):
    return np.ones_like(S)
def v0(t, S, I, R):
    return np.zeros_like(S)

u_func = u0
v_func = v0

prev_p1_mean = None
converged = False

for it in range(max_iter):
    print(f"Iteration {it+1}")
    # Forward
    S, I, R = forward_simulation(u_func, v_func)

```

```
    # Backward
    alpha1, alpha2, alpha3, p1, p2, p3 = backward_simulation(S
        , I, R)

    # Construction of new control functions
    def make_control_func(alpha1, alpha2, alpha3):
        def u_func_new(t, S_arr, I_arr, R_arr):
            n = int(np.round(t / dt))
            n = max(0, min(n, N-1))
            centers = compute_centers(S_arr, I_arr, R_arr)
            X = np.column_stack((S_arr, I_arr, R_arr))
            A = phi(X, centers)
            p1_curr = A @ alpha1[:, n]
            u = -(imm_rate / gamma1) * p1_curr
            return np.clip(u, 0, 1)
        def v_func_new(t, S_arr, I_arr, R_arr):
            n = int(np.round(t / dt))
            n = max(0, min(n, N-1))
            centers = compute_centers(S_arr, I_arr, R_arr)
            X = np.column_stack((S_arr, I_arr, R_arr))
            A = phi(X, centers)
            p1_curr = A @ alpha1[:, n]
            p2_curr = A @ alpha2[:, n]
            p3_curr = A @ alpha3[:, n]
            v = (beta*S_arr *I_arr/ gamma2) * (p2_curr -
                p1_curr)
            return np.clip(v, 0, 1)
        return u_func_new, v_func_new

    u_func, v_func = make_control_func(alpha1, alpha2, alpha3)

    # Convergence verification  (on the averge of p1 )
    p1_mean = np.mean(p1[:, :N], axis=0)
    if prev_p1_mean is not None:
        diff = np.linalg.norm(p1_mean - prev_p1_mean) / (np.
            linalg.norm(prev_p1_mean) + 1e-8)
        print(f"Diff relative p1: {diff:.6f}")
        if diff < tol:
            print("Convergence reached")
            converged = True
            break
```

```
300     prev_p1_mean = p1_mean
301
302 if not converged:
303     print("Muximum number of iterations reached")
```

Listing 3.3: Python code: Numerical simulation of SIRS epidemic model with controls and jumps.

Remark 3. *The program execution took* 52 *seconds to complete the full simulation on an HP Elitebook PC, Intel Core i5 vPro 8th Gen Processor.*

Conclusion

In this thesis, we developed a hierarchical epidemic model starting from a deterministic ODE system, then incorporating Brownian motion, and finally introducing Lévy jumps to capture abrupt, real-world events in a SIRS epidemic model.
Using the stochastic maximum principle for Lévy processes, we formulated an optimal control problem with two controls (immigration restriction and preventive measures) to reduce infected individuals while minimizing costs. The necessary optimality conditions were expressed via a Hamiltonian and a forward-backward system of stochastic differential equations for the state and adjoint variables.
Numerically, we applied the Least Squares Monte Carlo (LSMC) method and the Milstein scheme to solve our problem. Simulation results showed that the optimal control strategy significantly reduces both susceptible and infected populations compared to uncontrolled scenarios.
Future work may extend the model to incorporate vaccination explicitly as an additional differential equation rather than merely a control variable. Furthermore, to achieve better numerical accuracy and efficiency, real-world data could be employed based on more advanced numerical methods such as Deep Learning to approximate stochastic control epidemic model.

Bibliography

[1] D. Applebaum. lévy processes and stochastic calculus. *Cambridge University Press. Second Edition*, 2009.

[2] J. Bao, X. Mao, G. Yin et al. Competitive Lotka–Volterra population dynamics with jumps. *Nonlinear Analysis: Theory, Methods & Applications*, 74(17):6601–6616, 2011.

[3] M. Bayram, T. Partal, G. Buyukoz. Numerical methods for simulation of stochastic differential equations. *Advances in Difference Equations*, Vol. 2018, p. 17, 2018.

[4] S. Bentout, A. Chekroun, T. Kuniya. Parameter Estimation and Prediction for Coronavirus Disease Outbreak 2019 (COVID-19) in Algeria. *AIMS Public Health*, 7(2):306–318, 2020.

[5] F. Brauer. Mathematical epidemiology: Past, present, and future. *Infectious Disease Modelling*, 2(2): 113-127, 2017.

[6] F. Brauer, Carlos Castillo-Chavez. Mathematical models in population biology and epidemiology. *Springer New York. Second Edition*, 2012.

[7] Y. Cai, Y. Kang, W. Wang. A stochastic SIRS epidemic model with nonlinear incidence rate. *Applied Mathematics and Computation*, 305(772): 221–240, 2017.

[8] O. Diekmann, J.A.P. Heesterbeek. Mathematical Epidemiology of Infectious Diseases: Model Building, Analysis and Interpretation. *Wiley Series in Mathematical and Computational Biology*, Chichester, Wiley, 2000.

[9] T. Easlick, W.A. Sun. Unified stochastic SIR model driven by Lévy noise with time-dependency, arXiv:2201.03406v3, 2024.

[10] A.A. Essak. Stochastic Differential Equations In Epidemiology. University of Leicester. PhD Thesis, 2023.

[11] B.D. Ewald. Error and convergence of two numerical schemes for stochastic differential equations. *Numerical Methods for Partial Differential Equations*, 22(5):1247–1253, 2006.

[12] N.C. Framstad, B. Øksendal, A. Sulem. Sufficient Stochastic Maximum Principle for the Optimal Control of Jump Diffusions and Applications to Finance. *Journal of Optimization Theory and Applications*, Vol. 121, pp. 77–98, 2004.

[13] E. Gobet, J.-P. Lemor, X. Warin, A regression-based Monte Carlo method to solve backward stochastic differential equations. *The Annals of Applied Probability*, 15(3):2172–2202, 2005.

[14] C. Ji, D. Jiang, N. Shi. The Behavior of an SIR Epidemic Model with Stochastic Perturbation. *Stochastic Analysis and Applications*, 30(5):755–773, 2012.

[15] O. Kebiri, L. Neureither, C. Hartmann. Adaptive Importance Sampling with Forward-Backward Stochastic Differential Equations. Stochastic Dynamics Out of Equilibrium. *Springer Proceedings in Mathematics & Statistics*. Vol. 282, pp. 265–281. Springer Cham, 2019.

[16] W.O. Kermack, A.G. McKendrick. Contributions to the mathematical theory of epidemics, part I, Proc. R. Soc. Lond. Ser. A 115, 700–721, 1927.

[17] D.E. Kirk, Optimal control theory: An introduction. Courier Corporation, *Dover Publications*, 2004.

[18] PE. Kloeden, A. Neuenkirch. The Path-wise Convergence of Approximation Schemes for Stochastic Differential Equations. *LMS Journal of Computation and Mathematic*, 10:235–253, 2007.

[19] S. Kumar. The global impact of pandemics on world economy and public health response (Chap3), Computational Approaches for Novel Therapeutic and Diagnostic Designing to Mitigate SARS-CoV-2 Infection, *Academic Press*, pp:43–48, 2022.

[20] M. Lefebvre. Modeling and optimal control of infectious diseases. *Mathematics*, 12(13):2139, 2024.

[21] S. Li, SIR Epidemic Model with General Nonlinear Incidence Rate and Lévy Jumps. *Mathematics*, 12(2), 215, 2024.

[22] X. Mao, G. Marion. Environmental Brownian noise suppresses explosions in population dynamics. *Stochastic Processes and their Applications*, 97(1):95–110, 2002.

[23] M. Mehdaoui. Well-posedness results for a new class of stochastic spatio-temporal SIR-type models driven by proportional pure-jump Lévy noise. *Applied Mathematical Modelling*, Vol. 126, 2023.

[24] B. Øksendal, A. Sulem. Applied Stochastic Control of Jump Diffusions, *Springer Cham, Third Edition*, 2019.

[25] N. Privault, L. Wang. Stochastic SIR lévy jump model with heavy tailed increments, 2019, https://doi.org/10.48550/arXiv.1911.12924

[26] H.S.F. Rodrigues, Optimal control and numerical optimization applied to epidemiological models. *PhD Thesis*, Universidade de Aveiro, Portugal, 2012.

[27] Y. Sabbar. Mathematical analysis of some stochastic infectious disease models with white noises and Lévy jumps, PhD Thesis, Sidi Mohamed Ben Abdellah University of Fès, Morocco, 2021.

[28] A. Seierstad, K. Sydsaeter. Sufficient conditions in optimal control theory. *International Economic Review*, 18(2):367–391, 1977.

[29] H.J. Sussmann, J.C. Willems. 300 years of optimal control: from the brachystochrone to the maximum principle. Historical perspectives, *IEEE Control Systems Magazine*, 17(3):32–44, 1997.

[30] F. Tröltzsch. Optimal control of partial differential equations: theory, methods, and applications. *American Mathematical Society*, Vol. 112. pp. 399, 2010.

[31] J. Wu, H. Feng, D. Yao. Optimal Control of a lévy Inventory System: The Optimality of Control Band Policy, 2016. https://doi.org/10.48550/arXiv.1607.08671

[32] X. Zhang and K. Wang, Stochastic SIR model with jumps, *Applied Mathematics Letters*, 26(8):867–874, 2013.

[33] S. Zhao, Q. Lin, J. Ran, et al. Preliminary estimation of the basic reproduction number of novel coronavirus (2019-nCoV) in China, from 2019 to 2020: A data-driven analysis in the early phase of the outbreak. *International Journal of Infectious Diseases*, 92:214–217, 2020.

[34] Y. Zhou, W. Zhang. Threshold of a stochastic SIR epidemic model with Lévy jumps. *Physica A: Statistical Mechanics and its Applications*, Elsevier, 446(C):204–216, 2016.