Backward Simulation of Stochastic Process Using a Time Reverse Monte Carlo Method

SHINICHI TAKAYANAGI

Doctor of Philosophy

Department of Statistical Science
School of Multidisciplinary Sciences
SOKENDAI
(The Graduate University for Advanced Studies)

Backward Simulation of Stochastic Process Using a Time Reverse Monte Carlo Method

Shinichi Takayanagi

Department of Statistical Science
School of Multidisciplinary Sciences
SOKENDAI (The Graduate University for Advanced Studies)

This dissertation is submitted for the degree of
Doctor of Philosophy

March 2020

Acknowledgements

It has been a truly life-changing experience for me while I undertake this Ph.D. It would have been impossible to do without the support and guidance that I received from many people.

At first, I would like to express my gratitude to my supervisor, Prof. Iba, for his useful advice, which helped me to improve the quality of my research throughout my entire Ph.D. course at The Graduate University for Advanced Studies (SOKENDAI) and The Institute of Statistical Mathematics (ISM). Without his help, I would not have acquired the skills required to tackle our research challenges. I feel gratitude to the members of the dissertation committee, Genta Ueno, Hiroshi Fujisaki, and Shinsuke Koyama, who gave me many valuable comments.

I also thank colleagues at HOXO-M Inc for supporting me spiritually throughout writing this thesis. These brilliant colleagues always inspired me over many years, even though I went through hard times. Without their dedicated cooperation, I could not have completed this thesis.

Abstract

In this thesis, we discuss "backward simulation," which traces a time-reversed path from a target region to the initial configuration. If the outputs of the original simulation ("forward simulation") are easily restored from those obtained by backward dynamics, we can use backward simulation as a computational tool. In particular, the time required to calculate the probability to reach a target region from the initial configurations can be significantly reduced when the target region is small, but the initial distribution is broad. An example is a computation of the probability that a typhoon will hit the Tokyo area exactly under a given stochastic model.

It is, however, difficult to design backward dynamics with the desired properties. The naïve method described later defines a natural candidate for backward dynamics. Surprisingly, this naïve method does not, however, work as expected; it does not reproduce the correct probabilities defined by the forward simulation, and the calculation of factors required to correct the bias is often computationally expensive.

The aim of this thesis is to draw attention to these facts and propose an algorithm that partially resolves the problem. We named this algorithm the time reverse Monte Carlo method (TRMC). TRMC is based on the ideas of sequential importance sampling (SIS) and sequential Monte Carlo (SMC). Time-reversed dynamics itself was discussed in several studies, mostly from a theoretical viewpoint. On the other hand, related computational problems are found in data science, especially in time-series analysis using state-space models. Our problem can formally be regarded as a limiting case of the "smoothing" part of these algorithms, where only one observation ("target") is available at the end of the time series. There are, however, important differences from our problem, which are discussed in this thesis. TRMC essentially involves introducing simplified backward dynamics with a weight. This weight enables the bias of estimators to be corrected. In this algorithm, we introduce a backward transition probability. We can choose an arbitrary probability density as a backward transition probability, while the computation efficiency strongly depends on it.

To give concrete examples, we also present numerical results. Forward simulations are used to check the consistency and computational efficiency of our result. TRMC with SIS was tested for a stochastic difference equation and a stochastic typhoon model and the Lorenz 96 model; it converges more efficiently than forward simulations in some of these examples. Three types of improved versions of TRMC are also introduced. The first one is a higher-order approximation in backward dynamics. The second one is TRMC with resampling for simulations with a larger number of steps. The third one is TRMC with an external field. In these improvements, TRMC provides unbiased estimates of the probabilities without expensive computation. These three types of improved schemes are shown to be advantageous.

We also discussed the limitation. The examples provided in this thesis show that backward simulations using TRMC provide unbiased estimates of the probabilities and can be more efficient than forward simulations. In these examples, the computational efficiencies of TRMC are higher than those obtained by forward simulation. Note that TRMC can be used to calculate the probability for an arbitrarily small target region; this would be impossible by using forward simulation. There are, however, cases in which TRMC is inefficient. First, TRMC is not advantageous if the time-reversed paths rarely encounter a region in which the initial density is high; this can occur when the initial density is not broad. Another case in which TRMC can be inefficient is when the weight is highly time dependent. If paths with smaller weights in the initial stage of backward simulation acquire larger weights in the latter stage, resampling of the path (particle splitting) in SMC may not be effective. In this case, if TRMC with SIS is ineffective, TRMC with SMC also shows poor performance.

At last, we show the possible improvement of TRMC and its relation to the Bayes formula. It is useful to introduce optimal backward dynamics. Although it is not easy to obtain these dynamics *a priori*, the formal definition is derived. This formulation appears similar to the formulas used in Bayesian inference when the probability obtained by forward simulations is regarded as an analog of the prior distribution. It is also considered as the optimal backward dynamics.

Contents

1 Introduction 1

1.1 Background . . . 1

1.2 Failure of Naïve Method . . . 3

1.2.1 Derivation of Eq. (1.12) . . . 6

2 Theoretical Background 7

2.1 Sequential Importance Sampling . . . 7

2.2 Sequential Monte Carlo . . . 9

3 Time Reverse Monte Carlo Method 13

3.1 Time Reverse Monte Carlo Method . . . 13

3.2 Implementation for Stochastic Difference Equation . . . 15

3.2.1 Derivation of Eq. (3.9) . . . 16

4 Applications 18

4.1 Stochastic Difference Equation . . . 19

4.2 Stochastic Typhoon Model . . . 23

4.3 Lorenz 96 Model . . . 29

5 Improved Schemes 33

5.1 Higher-order Approximation . . . 33

5.2 Resampling . . . 35

5.3 External Field . . . 39

6 Discussion and Concluding Remarks 46

6.1 Discussion 46

6.2 Concluding Remarks 49

Appendix 50

A Time Reversal of Diffusions 50

B Application to Options Portfolio Valuation 54

List of Figures

1.1 Forward and backward simulations. The forward simulation is inefficient when the target region A is much smaller than the support of the initial distribution $p(x_0)$. The backward simulation simulates paths from the target region A to the support of the initial distribution $p(x_0)$. 2

1.2 Change in the infinitesimal volume in the state space along each path. 5

2.1 Graphical example of resampling. Particles with large weights are replaced with multiple copies of them, and particles with small weights are removed. 10

4.1 Convergence of TRMC for the stochastic difference equation (4.3). The estimated probabilities are converged to those obtained by FS, as the number of Monte Carlo paths increases. Error bars indicate approximate ±1 standard error confidence intervals by TRMC. The horizontal solid line indicates the estimated probability by FS. The horizontal dashed line represents ±1 standard error confidence intervals by FS. FS has the same number of Monte Carlo paths $M = 10^7$ as TRMC. 20

4.2 Convergence of TRMC for the stochastic difference equation (4.3). Smaller probability case. The estimated probabilities are converged to those probabilities by the forward simulation, as the number of Monte Carlo paths increases. Error bars indicate approximate ±1 standard error confidence intervals by TRMC. The horizontal solid line indicates the estimated probability by FS. The horizontal dashed line represents ±1 standard error confidence intervals by FS. FS has the same number of Monte Carlo paths $M = 10^7$ as TRMC. 21

4.3 Example of Monte Carlo paths generated by the stochastic typhoon model originating from the northwestern part of the Pacific Ocean. Each line corresponds to a path generated by the forward simulation. The black rectangular region shows the possible initial position of typhoons in the northwestern part of the Pacific Ocean. The initial positions of typhoons are uniformly distributed. 24

4.4 Example of Monte Carlo paths generated by TRMC starting from Tokyo. Each line corresponds to a path generated by TRMC. The black rectangular region corresponds to the possible initial position of typhoons in the northwestern part of the Pacific Ocean. 24

4.5 Convergence of TRMC for the stochastic typhoon model. The estimated probabilities converge to those obtained by FS as the number of Monte Carlo paths increases. Error bars indicate approximate ±1 standard error confidence intervals for TRMC. The horizontal solid line indicates the estimated probability by FS. The horizontal dashed line represents ±1 standard error confidence intervals for FS. FS has the same number of Monte Carlo paths, $M = 10^8$, as TRMC. 25

4.6 Example of Monte Carlo paths generated by TRMC starting from Tokyo. The velocity distribution is restricted to the tendency to move southward. Each line corresponds to a path generated by TRMC. The black rectangular region corresponds to the possible initial position of typhoons in the northwestern part of the Pacific Ocean. 27

4.7 Convergence of TRMC for the stochastic typhoon model in the smaller probability case. The estimated probabilities converge to those obtained by FS as the number of Monte Carlo paths increases. Error bars indicate approximate ±1 standard error confidence intervals for TRMC. The horizontal solid line indicates the estimated probability by FS. The horizontal dashed line represents ±1 standard error confidence intervals for FS. FS has the same number of Monte Carlo paths, $M = 10^8$, as TRMC. 28

4.8 A conceptual diagram of the Lorenz 96 Model. $x = \{x_k; k = 1 \dots K\}$ is regarded as some atmospheric quantity in K sectors of a latitude circle. 29

4.9 Convergence of TRMC for the Lorenz 96 model. The estimated probabilities converge to those obtained by FS as the number of Monte Carlo paths increases. Error bars indicate approximate ±1 standard error confidence intervals for TRMC. The horizontal solid line indicates the estimated probability by FS. The horizontal dashed line represents ±1 standard error confidence intervals for FS. FS has the same number of Monte Carlo paths, $M = 10^7$, as TRMC. 31

5.1 Distribution of weight $\prod_{i=0}^{N-1} W_i$ in TRMC and TRMC (HO) for the stochastic difference equation(4.3). The variance of the distribution by the TRMC (HO) is much smaller than that by TRMC. 34

5.2 Distribution of weight $\prod_{i=0}^{N-1} W_i$ in TRMC with different numbers of time steps. The vertical and horizontal lines indicate the weight density and the value of weights, respectively. The weight distributions with a large number of time steps have a heavy-tailed distribution. . . 36

5.3 Comparison among FS (Forward), TRMC, and TRMC(RS) for the Lorenz 96 model. $\alpha = \alpha_0\%$ means TRMC(RS) with $\alpha = \frac{\alpha_0}{100}$. 38

5.4 Distribution of the weight $\prod_{i=0}^{N-1} W_i$ in TRMC and TRMC (RS) for the Lorenz 96 model. We set the threshold α to 0.5 for TRMC (RS). The variance of the distribution is much smaller for TRMC (RS) than for TRMC. 38

5.5 Comparison of the estimated probabilities among FS (Forward), TRMC, and TRMC (EF) for the stochastic typhoon model. $\alpha = \alpha_0$ means TRMC (EF) with $\alpha = \alpha_0$. TRMC (EF) is more efficient than TRMC in a wide range of threshold values. Error bars indicate approximate ± 1 standard error confidence intervals. TRMC (EF, $\alpha = 1$) does not reproduce the unbiased estimates of the probability because of the strong external field. 41

5.6 Comparison among FS (Forward), TRMC, and TRMC (EF) for the stochastic typhoon model. $\alpha = \alpha_0$ means TRMC (EF) with $\alpha = \alpha_0$. TRMC (EF) is more efficient than TRMC in a wide range of threshold values. 42

5.7 Distribution of the weight $\prod_{i=0}^{N-1} W_i$ in TRMC and TRMC (EF) for the stochastic typhoon model. We set the threshold α to 10 for TRMC (EF). The variance of the distribution is much smaller for TRMC (EF) than for TRMC. 43

5.8 The number of a hit when α changes for the stochastic typhoon model. $\alpha = \alpha_0$ means TRMC (EF) with $\alpha = \alpha_0$. The number of a hit tends to increase when α decreases as we expected. 43

5.9 Example of Monte Carlo paths generated by TRMC starting from Tokyo. Each line corresponds to a path generated by TRMC. 44

5.10 Example of Monte Carlo paths generated by TRMC (EF) starting from Tokyo. Each line corresponds to a path generated by TRMC (EF, $\alpha = 10$). 44

5.11 The α dependency of M_{eff} for the stochastic typhoon model. $\alpha = \alpha_0$ means TRMC (EF) with $\alpha = \alpha_0$. The effective sample size becomes the largest when $\alpha = 10$, which is consistent with the tendency of computational efficiency. 45

5.12 The α dependency of the effective sample size for the stochastic typhoon model. $\alpha = \alpha_0$ means TRMC (EF) with $\alpha = \alpha_0$. 45

B.1 The shape of the payoff function, which we use here. The non-zero region of this payoff function corresponds to the target region A in TRMC. 56

B.2 Comparison among TRMC and the exact solution by BS formula for options portfolio valuation. Error bars indicate approximate 1 standard error confidence intervals for TRMC. 57

B.3 TRMC is more efficient than FS in a wide range of underlying security prices. ρ_2 values in a wide range of underlying security prices. 57

List of Tables

4.1 Comparison between TRMC, TRMC (no weight), and FS for a stochastic difference equation. 22

4.2 Comparison among TRMC, TRMC (restricted), TRMC (no weight), and FS for stochastic typhoon model. 26

4.3 Comparison among TRMC, TRMC (no weight), and FS for the Lorenz 96 model. 32

5.1 Comparison between TRMC, TRMC (HO), and the forward simulation for the stochastic difference equation (4.3) 34

5.2 Comparison among TRMC, TRMC (RS, α=50%), and FS for the Lorenz 96 model. 37

5.3 Comparison among TRMC and TRMC (EF) for the stochastic typhoon model. 40

Chapter 1

Introduction

1.1 Background

In this thesis, we discuss “backward simulation,” which traces a time-reversed path from a target region A to the initial configuration (Fig. 1.1). If the outputs of the original simulation (“forward simulation”) are easily restored from those obtained by backward dynamics, we can use backward simulation as a computational tool. In particular, the time required to calculate the probability to reach A from the initial configurations can be significantly reduced when the target region A is small but the initial distribution is broad. An example is a computation of the probability that a typhoon will hit the Tokyo area exactly under a given stochastic model (Sect. 4.2).

It is, however, difficult to design backward dynamics with the desired properties. Specifically, consider the forward dynamics of a D-dimensional stochastic process X defined by

$$X_{i+1} = g(X_i) + \eta_i, \tag{1.1}$$

where η_i is an independent noise that obeys an arbitrary distribution and the function $g : \mathbb{R}^D \to \mathbb{R}^D$ describes noiseless forward dynamics. Then, a naïve way to derive a time-reversed equation is to rearrange Eq. (1.1) as

$$X_i = g^{-1}(X_{i+1} - \eta_i). \tag{1.2}$$

Here, we assume that function g is a one-to-one and onto function and denotes the inverse function of g as g^{-1}. We can construct a time-reversed path iteratively, using

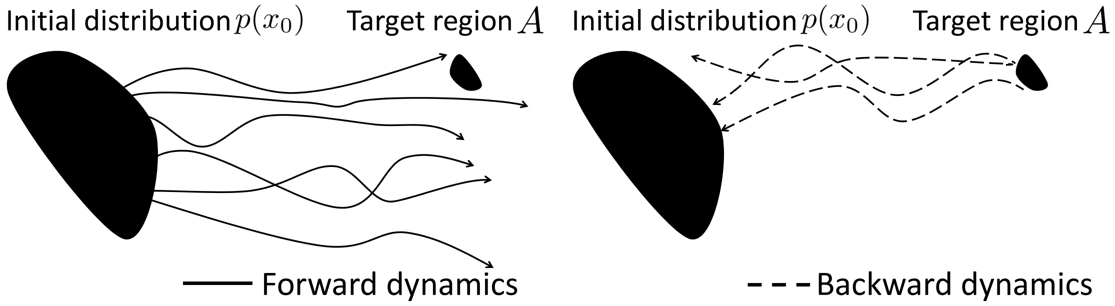


Figure 1.1: Forward and backward simulations. The forward simulation is inefficient when the target region A is much smaller than the support of the initial distribution $p(x_0)$. The backward simulation simulates paths from the target region A to the support of the initial distribution $p(x_0)$.

Eq. (1.2) and the independent realization of η_i, starting from the target region. It defines an apparently natural candidate for backward dynamics.

Surprisingly, this naïve method does not work as expected; it does not reproduce the correct probabilities defined by the forward simulation, and the calculation of factors required to correct the bias is often computationally expensive. This becomes clear in Sect. 1.2. Furthermore, the computation of g^{-1} in Eq. (1.2) is time-consuming and reduces the efficiency of the computation.

The aim of this thesis is to draw attention to these facts and propose an algorithm that partially resolves the problem. We named this algorithm the time reverse Monte Carlo method (TRMC). TRMC is based on the ideas of sequential importance sampling (SIS) [1, 2] and sequential Monte Carlo (SMC) [1, 3, 4, 5]. We discuss TRMC based on SIS in Chap. 4 and its improved version based on resampling and an external field in Chap. 5. There have been several studies using "path reweighting" in computational chemistry [6, 7]. These studies used reweighting for different purposes.

TRMC based on SIS is tested for a stochastic difference equation, a stochastic typhoon model and the Lorenz 96 model in Chap. 4. Three types of the improved version of TRMC are also introduced in Chap. 5. The first one is a higher-order approximation in backward dynamics. The second one is TRMC with resampling for simulations with a larger number of steps. The third one is TRMC with an external field. These three types of improved schemes are shown to be advantageous. In these improvements, TRMC provides unbiased estimates of the probabilities without expensive computation. In Sect. 6.1, we discuss its relation to the Bayes formula, as well as the possible improvement and limitations of TRMC.

Time-reversed dynamics itself was discussed in several studies [8, 9, 10], mostly from a theoretical viewpoint. On the other hand, related computational problems are found in data science, especially in time-series analysis using state-space models [11, 12, 13]. Our problem can formally be regarded as a limiting case of the "smoothing" part of these algorithms, where only one observation ("target") is available at the end of the time series. There are, however, important differences from our problem, which are discussed in Sect. 6.1. Studies related to the statistical inference on a discrete state stochastic process, such as genepropagation [2, 14] and information source detection [15] were also reported. These studies, however, did not consider dynamical systems of continuous variables.

1.2 Failure of Naïve Method

Here, we provide a detailed discussion of the naïve method and its drawbacks, which form the motivation for our algorithm. Before providing details, we formulate the problem. Let $S_T = \{0 = t_0 \leq t_1 \cdots \leq t_N = T\}$ be a partition of the interval $[0, T]$, and let step size $\Delta t = t_{i+1} - t_i$ be a constant; x_i is used to represent the value of stochastic process X at time point t_i. The transition probability density from x_i to x_{i+1} defined by Eq. (1.1) is denoted as $p\left(x_{i+1}|x_i\right)$. We consider an estimation of the probability $P(X_N \in A)$ that X_N hits a small target region A in the D-dimensional space. The probability is formally written as

$$P(X_N \in A) = \int dx_{0:N} \mathbf{1}_{x_N \in A} \left\{ \prod_{i=0}^{N-1} p\left(x_{i+1}|x_i\right) \right\} p(x_0), \tag{1.3}$$

where $\mathbf{1}_{x \in A}$ is the indicator function that takes value 1 when $x \in A$, and 0 otherwise, and $p(x_0)$ is the initial distribution of the forward simulation. Hereafter, $dx_{k:l}$

indicates $dx_k dx_{k+1} \cdots dx_l$ for $k \leq l$.

A naïve method is defined as a repeated simulation with a uniformly distributed initial condition in the target region A using Eq. (1.2). Initially, it appears sufficient to evaluate $P(X_N \in A)$ as $\frac{1}{M}\sum_{j=1}^{M} p(x_0^{(j)})$. However, there are two problems with this naïve method. First, the exact computation of g^{-1} in Eq. (1.2) is not easy. Computing g^{-1} using numerical root-finding techniques such as the Newton-Raphson method is computationally intensive and its severity increases as the dimension increases.

Second, this computation does not reproduce the correct probability $P(X_N \in A)$ even with the exact g^{-1}. To understand this problem, we show the difference between the forward simulation and the naïve method. Let us define

$$Y_i = X_i - \eta_{i-1} = g(X_{i-1}); i \in [1, \ldots, N]. \tag{1.4}$$

Using this definition, we can rewrite Eq. (1.2) as

$$Y_i + \eta_{i-1} = g^{-1}(Y_{i+1}). \tag{1.5}$$

Equation (1.5) can be simplified into

$$Y_i = g^{-1}(Y_{i+1}) - \eta_{i-1}. \tag{1.6}$$

The probability calculated using Eq. (1.6) corresponds to the equation

$$\int dy_{1:N} dx_N \frac{\mathbf{1}_{x_N \in A}}{V_A} \tilde{p}_f\left(y_N | x_N\right) \left\{ \prod_{i=1}^{N-1} \tilde{p}\left(y_i | y_{i+1}\right) \right\} p(g^{-1}(y_1)) V_A, \tag{1.7}$$

where $y_i = g(x_{i-1})$, $\tilde{p}_f\left(y_N | x_N\right)$ is the transition probability density from x_N to y_N defined by Eq. (1.4) with $i = N$ and $\tilde{p}\left(y_i | y_{i+1}\right)$ is the transition probability density from y_{i+1} to y_i defined by Eq. (1.6). An initial condition x_N is uniformly distributed in the target region A and V_A is the volume of target region A.

We have to introduce the Jacobian of function g so that Eq. (1.7) is consistent with Eq. (1.3). To show this, Eq. (1.3) is rewritten using equations

$$p(x_i | x_{i-1}) dx_i = \left|\det(J_{g^{-1}}(y_{i+1}))\right| \tilde{p}\left(y_i | y_{i+1}\right) dy_{i+1}, \tag{1.8}$$

$$i \in [1, \cdots, N-1]$$

$$p(x_0) dx_0 = \left|\det(J_{g^{-1}}(y_1))\right| p\left(g^{-1}(y_1)\right) dy_1, \tag{1.9}$$

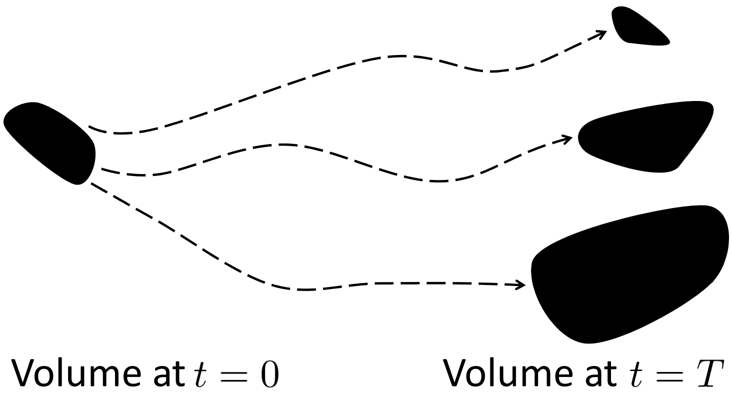


Figure 1.2: Change in the infinitesimal volume in the state space along each path.

where $|\det(J_{g^{-1}}(y_i))|$ is the absolute value of the Jacobian of function g^{-1}. As a result, the probability $P(X_N \in A)$ is calculated as

$$P(X_N \in A) = \int dy_{1:N} dx_N \frac{\mathbf{1}_{x_N \in A}}{V_A} J(y_1, \ldots, y_N) \tilde{p}_f\left(y_N | x_N\right) \left\{ \prod_{i=1}^{N-1} \tilde{p}\left(y_i | y_{i+1}\right) \right\} p\left(g^{-1}(y_1)\right) V_A, \tag{1.10}$$

$$J(y_1, \ldots, y_N) = \left\{ \prod_{i=0}^{N-1} |\det(J_{g^{-1}}(y_{i+1}))| \right\}. \tag{1.11}$$

We can obtain the correct probability using Eq. (1.10) instead of Eq. (1.7). The Jacobian $J_{g^{-1}}$ calculation is, however, computationally expensive.

We note that the factor $J(y_1, \ldots, y_N)$ goes to

$$\exp\left(- \int_0^T \mathrm{div} f(x_t) dt \right) \tag{1.12}$$

in the limit as $\Delta t \to 0$ when we assume that $g(x) = x + f(x)\Delta t$. The proof of Eq. (1.12) is given in next section (Sect. 1.2.1). This shows that we must include factor $J(y_1, \ldots, y_N)$ for unbiased estimation even in the limit of infinitesimal Δt. We can regard the factor written in Eq. (1.12) as the change in the infinitesimal volume along each path (Fig. 1.2).

1.2.1 Derivation of Eq. (1.12)

The aim of this section is to prove Eq. (1.12). Up to the first-order Δt, the Jacobian $\det(J_g(x))$ is given by

$$\begin{aligned}\det(J_g(x)) &= \det\left(I + \nabla f(x)\Delta t\right) \\ &= 1 + \mathrm{Tr}\left(\nabla f(x)\Delta t\right) + O((\Delta t)^2) \\ &= \exp\left[\mathrm{div} f(x)\Delta t\right] + O((\Delta t)^2),\end{aligned} \tag{1.13}$$

where I is the unit matrix of order $D \times D$. D is the dimension of stochastic process X.

Using Eq. (1.13), we obtain in the limit as $\Delta t \to 0$

$$\begin{aligned}J(y_1, \ldots, y_N) &= \prod_{i=0}^{N-1} \left|\det(J_{g^{-1}}(y_{i+1}))\right| \\ &= \exp\left[\sum_{i=1}^{N} -\mathrm{div} f(x_i)\Delta t\right] + O((\Delta t)^2) \\ &\xrightarrow[\Delta t \to 0]{} \exp\left[-\int_0^T \mathrm{div} f(x_t) dt\right].\end{aligned} \tag{1.14}$$

The above equation is Eq. (1.12).

Chapter 2

Theoretical Background

In this chater, we describe the basic knowledge of the sequential Monte Carlo (SMC) method since the time reverse Monte Carlo (TRMC) method is based on the ideas of SMC. SMC is one of the Monte Carlo methods to generate samples efficiently according to a series of probability distributions evolving with time using importance sampling and resampling mechanisms. SMC have been widely used to deal with stochastic dynamic systems in engineering, bioinformatics, finance and many other fields [1, 16, 17, 18, 19, 20, 21, 22, 23, 24].

A method for estimating the posterior state of a dynamic system by SMC is specifically called a particle filter [16, 19]. Particle filter has been the most common application of SMC. See, for example, [3, 25] for reviews. SMC method is also used for Bayesian modeling [26, 27].

We start this chapter the sequential Importance Sampling (SIS) method and then present SMC as a combination of SIS and resampling. SIS corresponds to the generalization of the importance sampling technique (IS) for a sequence of distributions.

2.1 Sequential Importance Sampling

Here, we assume that a probability density $p_i(x_{1:i})$ is a complicated probability distribution not to sample from it directly. Then, we consider approximating a probability density $p_i(x_{1:i})$ for particular i. Hereafter, for any sequence $\{x_i\}_{i\geq 1}$, $x_{k:l}$

indicates $(x_k, x_{k+1}, \dots, x_l)$ for $k \leq l$. In SIS, we obtain sample from the probability density $p_i(x_{1:i})$ using a probability density $q_i(x_{1:i})$ called proposal density. Typically, we select $q_i(x_{1:i})$ to be easy to sample from (e.g., uniform or Gaussian distribution, etc.). In SIS literature, $q_i(x_{1:i})$ has the following structure

$$\begin{aligned} q_i(x_{1:i}) &= q_{i-1}(x_{1:(i-1)})q_i(x_i|x_{1:(i-1)}) \\ &= q_1(x_1)\prod_{k=2}^{i} q_k(x_k|x_{1:(k-1)}). \end{aligned} \tag{2.1}$$

We also introduce the weight function as the following

$$w_i(x_{1:i}) = \frac{p_i(x_{1:i})}{q_i(x_{1:i})}. \tag{2.2}$$

If we generate M independent samples $x_{1:i}^{(j)} \sim q_i(x_{1:i})$ for $j = 1, \dots, M$, we can approximate $p_i(x_{1:i})$ by the empirical probability density $\hat{p}_i(x_{1:i})$ using Eq. (2.1) and (2.2) as the following

$$\hat{p}_i(x_{1:i}) = \frac{1}{M}\sum_{j=1}^{M} w_i(x_{1:i}^{(j)})\delta_{x_{1:i}^{(j)}}(x_{1:i}) \tag{2.3}$$

, where $\delta_{x_0}(x)$ denotes the Dirac delta function located at x_0.

We carefully need to select $q_i(x_{1:i})$ because sampling efficiency strongly depends on the choice of an proposal density $q_i(x_{1:i})$. The same problem occurs in our study. See Sect. 3.1 for details. An appropriate selection of $q_i(x_{1:i})$ in SIS is to find $q_i(x_{1:i})$ which minimizes the variance of the weights $w_i(x_{1:i})$. Obviously, this value is minimised at $q_i(x_{1:i}) = p_i(x_{1:i})$. It is impossible to select $q_i(x_{1:i}) = p_i(x_{1:i})$ since this is the reason why we have to use $q_i(x_{1:i})$ instead of $p_i(x_{1:i})$. However, as a policy for tackling this problem, this idea results in that we should select $q_i(x_{1:i})$, which is close as possible as to the original probability density $p_i(x_{1:i})$.

To obtain samples following the distribution $q_i(x_{1:i})$, we sample $x_1 \sim q_1(x_1)$ at time 1 then $x_k \sim q_k(x_k|x_{1:(k-1)})$ at time k for $k = 2, \dots, i$. We can calculate the associated weights $w_i(x_{1:i})$ recursively using the following decomposition

$$\begin{aligned} w_i(x_{1:i}) &= \frac{p_i(x_{1:i})}{q_i(x_{1:i})} \\ &= \frac{p_{i-1}(x_{1:(i-1)})p_i(x_i|x_{1:(i-1)})}{q_{i-1}(x_{1:(i-1)})q_i(x_i|x_{1:(i-1)})} \\ &= w_{(i-1)}(x_{1:(i-1)})\frac{p_i(x_i|x_{1:(i-1)})}{q_i(x_i|x_{1:(i-1)})}. \end{aligned} \tag{2.4}$$

SIS algorithm proceeds as follows

SIS Algorithm

Step 1: At time $k = 1$

(a) Draw M samples $\left\{x_1^{(1)}, \cdots, x_1^{(M)}\right\}$ from $q_1(x)$.

(b) Calculate the weights $w_1(x_1^{(j)})$ for $j = 1, \ldots, M$.

Step 2: Apply the following steps for $j = 1, \ldots, M$, and for $k = 2, \ldots, i$

(a) Generate a sample $x_k^{(j)}$ with the probability $q_{k-1}(x|x_{1:(k-1)}^{(j)})$.

(b) Compute the weight $w_k(x_{1:k}^{(j)})$ using Eq.(2.4).

Though successes in short-run simulation, it will turn out that SIS algorithm fails in the long-run simulation. Notably, the weights will become highly degenerate after several time steps, in the sense that a small part of particles contains almost all weights. It means that most particles contribute nothing to the probability density estimation. See, for example [28].

It is a challenging problem when the probability density $p_i(x_{1:i})$ has unknown parameters. These must be estimated before running SIS algorithm. Some developments are found in [29, 30, 31] to solve this problem. In this thesis, we assume all the parameters are known.

2.2 Sequential Monte Carlo

As we describe the above, the variance of the weights $\{w_i^j\}_{j=1}^M$ tends to increase as time proceeds in most cases [3]. This phenomenon is referred to as degeneracy,

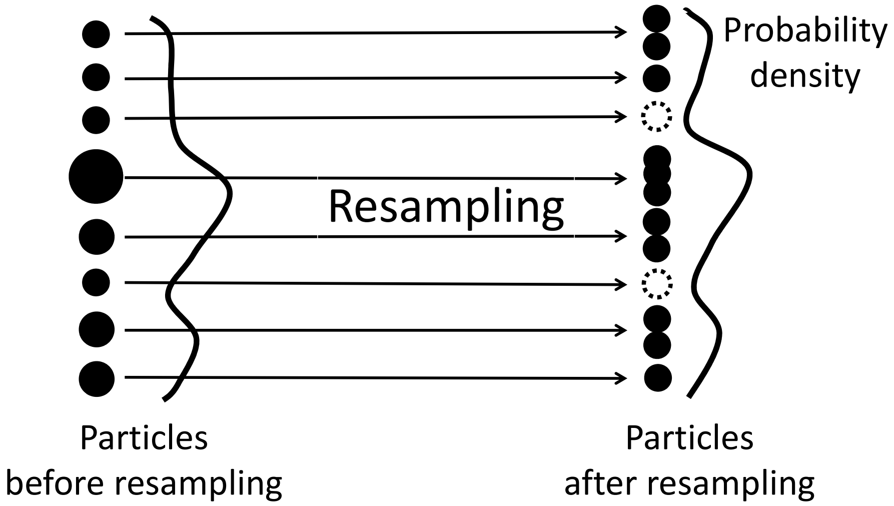


Figure 2.1: Graphical example of resampling. Particles with large weights are replaced with multiple copies of them, and particles with small weights are removed.

and it means that the weights become unbalanced, and a few weights dominate all the others. This consequently causes a decrease in computational efficiency. One of the ways to solve this problem is resampling [3, 4, 5, 32]. Resampling increases the number of high weighted particles while decreasing low weighted particles. After resampling, weights $\left\{w_i^j\right\}_{j=1}^M$ are reset to $\frac{1}{M}$. Figure 2.1 shows a graphical scheme of resampling. Resampling its own adds additional Monte Carlo error; however, resampling avoids the accumulation of error over time and make the approximation of the probability distributions $p_i(x_{1:i})$ much more stable. There have been similar ideas called splitting [33, 34, 35, 36] for rare event simulation.

We should not resample at every time step in a simulation. There is a measure of the quality of the particle approximation. One of the most standard measures is the effective sample size (ESS) [17, 20], which is given at time i

$$M_{eff} = \frac{1}{\sum_{j=1}^{M} \tilde{w}_i(x_{1:i}^{(j)})^2}, \tag{2.5}$$

where $\tilde{w}$ is the normalized weight

$$\tilde{w}_i(x_{1:i}^{(j)}) = \frac{\tilde{w}_i(x_{1:i}^{(j)})}{\sum_{j=1}^{M} w_i(x_{1:i}^{(j)})}. \tag{2.6}$$

A small value of M_{eff} corresponds to high degeneracy. M_{eff} is a number between 1 and M which indicates how many useful samples the algorithm has roughly. A resampling procedure is performed when M_{eff} is less than a certain threshold; typically $\frac{M}{2}$ [3]. The other criteria is the entropy of the weights H_i [37, 38], which reaches the maximum value when $\tilde{w}_i(x_{1:i}^{(j)}) = \frac{1}{M}$

$$H_i = -\sum_{j=1}^{M} \tilde{w}_i(x_{1:i}^{(j)}) \log 2 \left(\tilde{w}_i(x_{1:i}^{(j)}) \right). \tag{2.7}$$

We resample when the entropy is less than a certain given threshold.

Here, we use the following procedure of resampling at time i.

Resampling

1. M times with replacement from set $\left\{ x_{1:i}^{(j)} \right\}_{j=1}^{M}$ of samples, where the probability of sampling set of $x_{1:i}^{(j)}$ is proportional to $w_i(x_{1:i}^{(j)})$.
2. $\left\{ x_{1:i}^{(j)} \right\}_{j=1}^{M}$ and associated weights $\left\{ w_i(x_{1:i}^{(j)}) \right\}_{j=1}^{M}$ are replaced by the set of replicated samples with an equal weight $\frac{1}{M}$.

There are commonly used three resampling schemes: multinomial resampling [16], residual resampling [39], and stratified resampling [40]. We use the simplest approach, multinomial resampling. See [41] for some other commonly used resampling algorithms.

The sequential Monte Carlo (SMC) is a combination of SIS and resampling. Sometimes, SMC is referred to as the sequential Importance Resampling (SIR). The algorithm is as follows.

SMC Algorithm

Step 1: At time $k = 1$

(a) Draw M samples $\left\{ x_1^{(1)}, \cdots, x_1^{(M)} \right\}$ from $q_1(x)$.

(b) Calculate the weights $w_1(x_1^{(j)})$ for $j = 1, \ldots, M$.

(c) If resampling criteria is satisfied then do resampling.

Step 2: Apply the following steps for $j = 1, \ldots, M$, and for $k = 2, \ldots, i$

(a) Generate a sample $x_k^{(j)}$ with the probability $q_{k-1}(x|x_{1:(k-1)}^{(j)})$.

(b) Compute the weight $w_k(x_{1:k}^{(j)})$ using Eq.(2.4).

(c) If resampling criteria is satisfied then do resampling.

After the resampling step at any time point i, we obtain an approximations of $p_i(x_{1:i})$

$$\hat{p}_i(x_{1:i}) = \frac{1}{M} \sum_{j=1}^{M} \delta_{x_{1:i}^{(j)}}(x_{1:i}). \tag{2.8}$$

Some other commonly used Monte Carlo methods can be regarded as particular cases of SMC algorithm introduced above. For example, the annealed importance sampling [42] can be considered as SMC samplers without resampling.

Chapter 3

Time Reverse Monte Carlo Method

3.1 Time Reverse Monte Carlo Method

To overcome the difficulties we explained in Sect.1.2, we propose the TRMC method. TRMC essentially involves introducing simplified backward dynamics with a weight. This weight enables the bias of estimators to be corrected. First, we introduce a backward transition probability $q\left(x_{i+1} \to x_i\right)$ from x_{i+1} to x_i. We can choose an arbitrary probability density q, while the computation efficiency strongly depends on it. Once we introduce $q\left(x_{i+1} \to x_i\right)$, we can rewrite Eq. (1.3) as

$$P(X_N \in A) = \int dx_{0:N} \frac{\mathbf{1}_{x_N \in A}}{V_A} \left\{ \prod_{i=0}^{N-1} q\left(x_{i+1} \to x_i\right) W_i \right\} V_A p(x_0), \tag{3.1}$$

where

$$W_i = \frac{p\left(x_{i+1} | x_i\right)}{q\left(x_{i+1} \to x_i\right)} \tag{3.2}$$

is the weight required to correct the bias of estimators.

The algorithm consists of the following steps.

TRMC Algorithm

Step 1: Draw M samples $\left\{x_N^{(1)}, \cdots, x_N^{(M)}\right\}$ from the uniform distribution in V_A.

Step 2: Apply the following steps for $j = 1, \ldots, M$, and for $i = N-1, \ldots, 0$.

(a) Generate a sample from $x_{i+1}^{(j)}$ to $x_i^{(j)}$ with transition probability $q\left(x_{i+1}^{(j)} \to x_i^{(j)}\right)$.

(b) Calculate weight $W_i^{(j)}$ using Eq. (3.2).

Step 3: Evaluate the unbiased estimates of probability $P(X_N \in A)$ as

$$P(X_N \in A) \backsimeq \frac{1}{M} \sum_{j=1}^{M} W^{(j)}, \tag{3.3}$$

where the factor

$$W^{(j)} = \left\{ \prod_{i=0}^{N-1} W_i^{(j)} \right\} V_A p(x_0^{(j)}) \tag{3.4}$$

is attached to each simulation path.

The inputs of our algorithm are the number of Monte Carlo paths M, the number of time steps N, the initial distribution $p(x_0)$, the target region A, and the transition probability density q. When we actually perform the simulation on our computers, we take the logarithm of these weights to prevent numerical overflow.

This algorithm provides unbiased estimates of the desired probabilities. The idea of this scheme is a kind of SIS [1]. An advantage of our method is that we do not need to calculate g^{-1} or their Jacobian matrices at each i.

Suppose $p(x_0)$ is uniformly distributed on $B \subset \mathbb{R}^D$; $p(x_0) = \frac{1}{V_B}\mathbf{1}_{x_0 \in B}$, V_B is the volume of B. The efficiency of our algorithm does not depend on the factor V_A when V_B is considerably large while the forward simulation does not work in the limit as $V_A \to 0$. This is the advantage of using our algorithm.

The remaining problem involves determining the method for choosing the transition probability $q\left(x_{i+1} \to x_i\right)$. The basic idea is to choose the backward dynamics that generates trajectories similar to the forward dynamics defined by Eq. (1.1). The similarity of the trajectory is measured by $W^{(j)}$ in Eq. (3.4).

3.2 Implementation for Stochastic Difference Equation

To give concrete examples of the transition probability $q\left(x_{i+1} \to x_i\right)$, we assume the forward dynamics to be given in the following form:

$$X_{i+1} = X_i + f\left(X_i\right)\Delta t + \epsilon_i \sqrt{\Delta t}. \tag{3.5}$$

This corresponds to the case wherein $g(x) = x + f(x)\Delta t$ in Eq. (1.1). The noise ϵ_i is assumed to be i.i.d. Gaussian noise with mean zero and the variance-covariance matrix $\Sigma = \sigma\sigma^T$. This class of equations appears in a wide range of problems in many different fields such as physics [43], computational chemistry [44], and mathematical finance [45, 46].

In this case, as a simple choice, we can use the following backward dynamics:

$$X_i = X_{i+1} - f\left(X_{i+1}\right)\Delta t + \epsilon_i \sqrt{\Delta t}, \tag{3.6}$$

where we used the symmetry of Gaussian noise ϵ_i. This approximation corresponds to substituting $f\left(X_{i+1}\right)$ for $f\left(X_i\right)$ in Eq. (3.5). This zeroth-order approximation seems to be quite a bold assumption at first glance. However, even under this approximation, we can get the correct probabilities using the weight as we will see later in Chap. 4. In addition to that, we introduce a more efficient higher-order approximation in Sect. 5.1.

With this choice, weight W_i in Eq. (3.2) takes the form

$$\begin{aligned} W_i &= \frac{p\left(x_{i+1}|x_i\right)}{q\left(x_{i+1} \to x_i\right)} \qquad (3.7)\\ &= \frac{\exp\left[-\frac{1}{2}\left(x_{i+1} - x_i - f(x_i)\Delta t\right)^T \left(\Sigma\Delta t\right)^{-1}\left(x_{i+1} - x_i - f(x_i)\Delta t\right)\right]}{\exp\left[-\frac{1}{2}\left(x_{i+1} - x_i - f(x_{i+1})\Delta t\right)^T \left(\Sigma\Delta t\right)^{-1}\left(x_{i+1} - x_i - f(x_{i+1})\Delta t\right)\right]}\\ &= \exp\left[-\left(f(x_{i+1}) - f(x_i)\right)^T \Sigma^{-1}\left(\left(x_{i+1} - x_i\right) - \frac{\Delta t}{2}\left(f(x_{i+1}) + f(x_i)\right)\right)\right]. \qquad (3.8)\end{aligned}$$

As we show in the next section, the resultant algorithm is simple yet effective compared with the forward simulation when the target region A is smaller than the support of the initial distribution $p(x_0)$.

We note that the factor $\prod_{i=0}^{N-1} W_i$ goes to

$$\exp\left(-\int_0^T \mathrm{div} f(x_t) dt\right) \tag{3.9}$$

in the limit as $\Delta t \to 0$. The proof of Eq. (3.9) is given in next section (Sect.3.2.1). Note that Eq. (3.9) coincides with Eq. (1.12) derived from a different assumption.

3.2.1 Derivation of Eq. (3.9)

The aim of this section is to prove (3.9).

Up to the first-order Δt, the weight at time t_i is given by

$$\begin{aligned}
W_i &= \exp\left[-\left(f(x_{i+1}) - f(x_i)\right)^T \Sigma^{-1}\left((x_{i+1} - x_i) - \frac{\Delta t}{2}\left(f(x_{i+1}) + f(x_i)\right)\right)\right] \quad (3.10)\\
&= \exp\left[\mathrm{Tr}\left(-\left(f(x_{i+1}) - f(x_i)\right)^T \Sigma^{-1}\left((x_{i+1} - x_i) - \frac{\Delta t}{2}\left(f(x_{i+1}) + f(x_i)\right)\right)\right)\right]\\
&= \exp\left[-\,\mathrm{Tr}\left(\left(\nabla f(x_i)(x_{i+1} - x_i)\right)^T \Sigma^{-1}(x_{i+1} - x_i)\right) + o(\Delta t)\right]\\
&= \exp\left[-\,\mathrm{Tr}\left(\nabla f(x_i)^T \Sigma^{-1}(x_{i+1} - x_i)(x_{i+1} - x_i)^T\right) + o(\Delta t)\right].
\end{aligned}$$

In the limit as $\Delta t \to 0$, Eq. (3.5) becomes the following stochastic differential equation:

$$dX_t = f\left(X_t\right) dt + \sigma dW_t, \tag{3.11}$$

where W_t is a standard Brownian motion. Here, we used Ito's rule [45, 46], in which we substitute $\sqrt{dt}$ for dW_t and consider up to the order of dt. Using Eq. (3.11), we obtain the following relation in the limit as $\Delta t \to 0$

$$(x_{i+1} - x_i)(x_{i+1} - x_i)^T \xrightarrow[\Delta t \to 0]{} dx_t dx_t^T = \left(f\left(x_t\right) dt + \sigma dW_t\right)\left(f\left(x_t\right) dt + \sigma dW_t\right)^T \tag{3.12}$$

$$= \sigma dW_t dW_t^T \sigma^T + o(dt) = \Sigma dt, \tag{3.13}$$

where we used the relationships $dW_t dW_t^T = dt$ and $\sigma\sigma^T = \Sigma$.

As a result, we obtain using Eq. (3.10) and (3.13)

$$\prod_{i=0}^{N-1} W_i = \exp\left[-\sum_{i=0}^{N-1} \mathrm{Tr}\left(\nabla f(x_i)^T \Sigma^{-1} (x_{i+1} - x_i)(x_{i+1} - x_i)^T \right) + o(\Delta t) \right] \tag{3.14}$$

$$\xrightarrow[\Delta t \to 0]{} \exp\left[-\int_0^T \mathrm{Tr}\left(\nabla f(x_t)^T \right) dt \right] = \exp\left[-\int_0^T \mathrm{div} f(x_t) dt \right], \tag{3.15}$$

which is Eq. (3.9).

Chapter 4

Applications

We present the numerical results in this chapter. Forward simulations (FS) are used to check the consistency and computational efficiency of our result.

Using forward and backward dynamics, we simulate sample trajectories $x = \{x_1, \cdots, x_N\}$ generated by each model and compute the probability $P(X_N \in A)$ from M independent simulations.

We denote a standard error of TRMC to evaluate the computational efficiency by σ_s. We also denote the standard error of FS by σ_s^F. Using these variables, we define a relative value of variance by

$$\rho_1 = \left(\frac{\sigma_s^F}{\sigma_s} \right)^2 . \tag{4.1}$$

The factor ρ_1 indicates the computational efficiency only including the effect caused by the variance of estimators for a fixed sample size. With this definition, more complex algorithms tend to be more efficient while they require more computational time. Then, we also define another measure of the relative computational efficiency ρ_2 as

$$\rho_2 = \rho_1 \frac{\tau^F}{\tau}, \tag{4.2}$$

where τ is the computational time in seconds of the simulation and τ^F is the computational time of FS in seconds. This efficiency is defined in the sense of the actual performance considering both the computational time and the variance of the resulting estimates.

4.1 Stochastic Difference Equation

To show the working of the TRMC algorithm, we first deal with a two-dimensional stochastic difference equation defined by

$$\begin{aligned} x_{i+1} &= x_i + \left(x_i + \lambda_x y_i^2\right) \Delta t + \epsilon_x \sqrt{\Delta t}, \\ y_{i+1} &= y_i + \left(y_i + \lambda_y x_i\right) \Delta t + \epsilon_y \sqrt{\Delta t}, \end{aligned} \tag{4.3}$$

where λ_x and λ_y are constants. Noise ϵ_x and ϵ_y obey a Gaussian distribution with mean zero and variances σ_x^2 and σ_y^2, respectively.

We evaluate the convergence speed of our algorithm for the stochastic difference equations. Here we consider equation (4.3) for $\lambda_x = 1, \lambda_y = 3, \Delta t = 0.01$ and $\sigma_x = \sigma_y = 2$. Target region A is $\{(x, y); 19.875 \leq x \leq 20.125, 13.875 \leq y \leq 14.125\}$. We also assume that initial state (x_0, y_0) is uniformly distributed in $D = \{(x, y); 3 \leq x \leq 7, 8 \leq y \leq 12\}$. We set the number of Monte Carlo paths M to 10^7 and the number of time steps N to 10. The result of the simulation using these parameters is shown in Table 4.1 (Case I). It reveals that our algorithm gives unbiased probabilities as compared to those calculated by FS. Furthermore, it shows that TRMC is 2.6 times in ρ_2 (4.3 times in ρ_1) more efficient than FS. The row titled as "TRMC (no weight)" means that we ignore the factor defined by equation (3.4) when we evaluate the probability. In this case, it does not reproduce the unbiased estimates of the probability.

Fig.4.1 shows the convergence of TRMC when the number of Monte Carlo paths M increases. The horizontal line in Fig.4.1 indicates the estimated probability by FS with the number of Monte Carlo paths $M = 10^7$. The horizontal dashed line in Fig.4.1 shows the ± 1 standard error confidence intervals by FS with the number of Monte Carlo paths $M = 10^7$. It reveals that our algorithm converges correctly on increasing the number of Monte Carlo paths M.

To simulate events with smaller probabilities, we make target region A smaller as $\{(x, y); 19.9375 \leq x \leq 20.0625, 13.9375 \leq y \leq 14.0625\}$. We have shown the results as Case II in Table 4.1 and Fig.4.2. These results show that our algorithm becomes more efficient as the probability decreases.

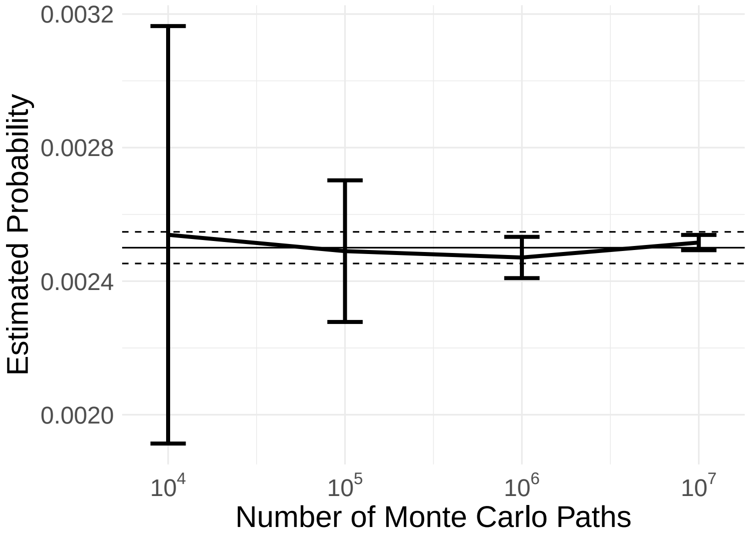


Figure 4.1: Convergence of TRMC for the stochastic difference equation (4.3). The estimated probabilities are converged to those obtained by FS, as the number of Monte Carlo paths increases. Error bars indicate approximate ±1 standard error confidence intervals by TRMC. The horizontal solid line indicates the estimated probability by FS. The horizontal dashed line represents ±1 standard error confidence intervals by FS. FS has the same number of Monte Carlo paths $M = 10^7$ as TRMC.

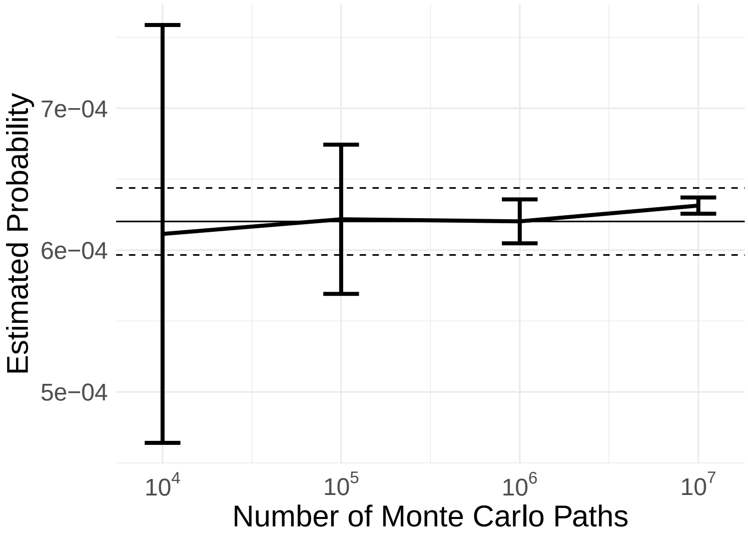


Figure 4.2: Convergence of TRMC for the stochastic difference equation (4.3). Smaller probability case. The estimated probabilities are converged to those probabilities by the forward simulation, as the number of Monte Carlo paths increases. Error bars indicate approximate ±1 standard error confidence intervals by TRMC. The horizontal solid line indicates the estimated probability by FS. The horizontal dashed line represents ±1 standard error confidence intervals by FS. FS has the same number of Monte Carlo paths $M = 10^7$ as TRMC.

Table 4.1: Comparison between TRMC, TRMC (no weight), and FS for a stochastic difference equation.

Case I

Method	$P(X_N \in A)$	σ_s	ρ_1	ρ_2
TRMC	2.516×10^{-3}	0.007×10^{-3}	4.3	2.6
TRMC (no weight)	3.546×10^{-3}	$< 10^{-6}$	–	–
FS	2.500×10^{-3}	0.015×10^{-3}	1.0	1.0

Case II

Method	$P(X_N \in A)$	σ_s	ρ_1	ρ_2
TRMC	6.314×10^{-4}	0.019×10^{-4}	17.1	10.7
TRMC (no weight)	8.893×10^{-4}	$< 10^{-7}$	–	–
FS	6.202×10^{-4}	0.079×10^{-4}	1.0	1.0

4.2 Stochastic Typhoon Model

The first example is a stochastic typhoon model [47], which gives an example of risk estimation by the proposed method. The stochastic typhoon model was designed to reproduce the statistics of typhoons in the northwestern part of the Pacific Ocean. This is a four-dimensional model given by

$$\begin{aligned} x_{i+1} &= x_i + v_i, \\ v_{i+1} &= V\left(x_{i+1}\right) + w\left(v_i - V\left(x_i\right)\right) + \epsilon_i, \\ V\left(x_i\right) &= a_0 + a_1 x_{\phi,i} + a_2 \sin x_{\lambda,i} + a_3 \sin^2 x_{\lambda,i}, \end{aligned} \tag{4.4}$$

where we use a global coordinate system defined by the geographic longitude (ϕ) and latitude (λ). We also define the two-dimensional position $x = (x_\phi, x_\lambda)$, speed $v = (v_\phi, v_\lambda)$ of a typhoon, and function $V(x) = (V_\phi(x), V_\lambda(x))$. $w, a_0, a_1, a_2,$ and a_3 are constants. The noise ϵ obeys a Gaussian distribution with mean zero and variances σ^2.

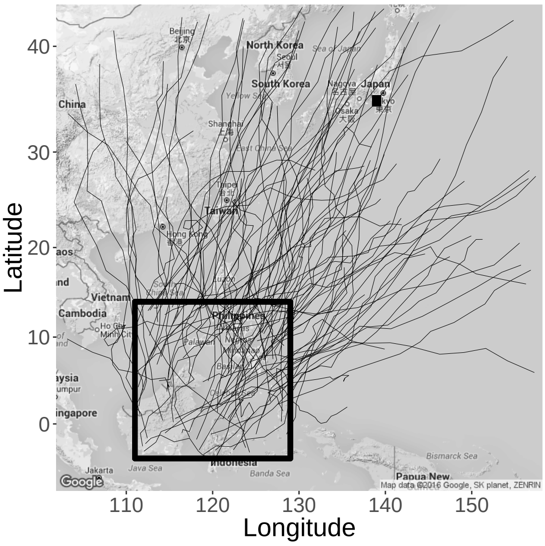
We fix $w = 0.93$, $a_0 = (0.792, 0.538)$, $a_1 = (0.122, 0.371)$, $a_2 = (-0.513, 0.583)$, $a_3 = (0.770, -0.387)$, $\sigma = 0.4$. We estimated these parameters based on the typhoon best track data for 62 years from 1951 to 2012, which are provided by the RSMC (Regional Specialized Meteorological Center) Tokyo. The best track data is available on the RSMC website [1]. Since typhoon behavior has seasonality, we have to estimate these parameters in each month. We fix these parameters based on June data so that the probability becomes $O(10^{-4})$ as we see later.

The target region A is $\{(x_\phi, x_\lambda); 138.5 \leq x_\phi \leq 139.5, 34.5 \leq x_\lambda \leq 35.5\}$. Since there is no range constraint on the distribution of the final speed v_f at the target, we adopt a uniform distribution with a suitably wide range U_f; here, U_f is defined as the region $\{(v_\phi, v_\lambda); V_\phi(x_A) - 3 \leq v_\phi \leq V_\phi(x_A) + 3, V_\lambda(x_A) - 3 \leq v_\lambda \leq V_\lambda(x_A) + 3\}$, where x_A is the center of target region A.

We also assume that the initial condition is uniformly distributed in $D = \{(x, v); 111 \leq x_\phi \leq 129, -4 \leq x_\lambda \leq 14, v \in U_0\}$, where U_0 is defined as the region $\{(v_\phi, v_\lambda); V_\phi(x_0) - 1.5 \leq v_\phi \leq V_\phi(x_0) + 1.5, V_\lambda(x_0) - 1.5 \leq v_\lambda \leq V_\lambda(x_0) + 1.5\}$ and $x_0 = (120, 5)$. This corresponds to the case wherein typhoons that occurred in the Philippines travel to

[1] http://www.jma.go.jp/jma/jma-eng/jma-center/rsmc-hp-pub-eg/trackarchives.html

the Tokyo area exactly with a small probability. We set M to 10^8 and N to 16. Examples of Monte Carlo paths for both simulations are given in Figs. 4.3 and 4.4.



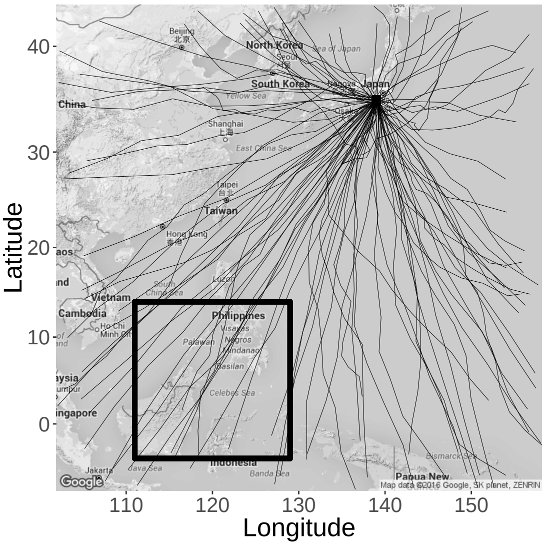


Figure 4.3: Example of Monte Carlo paths generated by the stochastic typhoon model originating from the northwestern part of the Pacific Ocean. Each line corresponds to a path generated by the forward simulation. The black rectangular region shows the possible initial position of typhoons in the northwestern part of the Pacific Ocean. The initial positions of typhoons are uniformly distributed.

Figure 4.4: Example of Monte Carlo paths generated by TRMC starting from Tokyo. Each line corresponds to a path generated by TRMC. The black rectangular region corresponds to the possible initial position of typhoons in the northwestern part of the Pacific Ocean.

This simulation was carried out on a laptop computer with 2.3 Ghz Intel core i5 and 8 GBytes memory. The computational time of TRMC in this simulation for generating 10^8 Monte Carlo paths is around 4.0×10^3 seconds.

Table 4.2 shows the result of computational experiments for the stochastic typhoon model. It shows that the probabilities of FS and TRMC agree within the error bars. If we ignore the factor defined by Eq. (3.4), it does not reproduce the un-

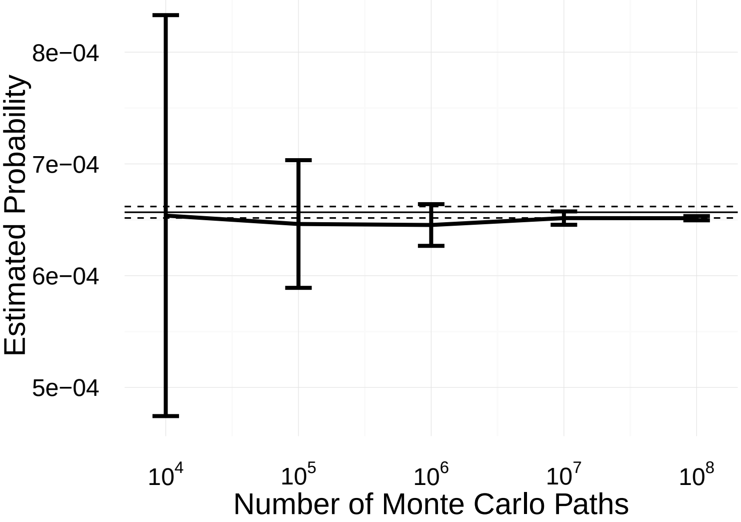


Figure 4.5: Convergence of TRMC for the stochastic typhoon model. The estimated probabilities converge to those obtained by FS as the number of Monte Carlo paths increases. Error bars indicate approximate ±1 standard error confidence intervals for TRMC. The horizontal solid line indicates the estimated probability by FS. The horizontal dashed line represents ±1 standard error confidence intervals for FS. FS has the same number of Monte Carlo paths, $M = 10^8$, as TRMC.

biased probability as in the case of the stochastic difference equation. Furthermore, it shows that TRMC is 4.2 times in terms of ρ_2 and 7.3 times in $\rho 1$) more efficient than FS. Fig. 4.5 shows the convergence of TRMC when the number of Monte Carlo paths M increases. It reveals that our algorithm converges correctly on increasing the number of Monte Carlo paths M.

To simulate events with smaller probabilities, we make the target region A smaller as $\{(x_\phi, x_\lambda); 138.75 \leq x_\phi \leq 139.25, 34.75 \leq x_\lambda \leq 35.25\}$. It shows that the smaller the probability, the more efficient our algorithm becomes as compared with the FS.

In Fig. 4.4, a few Monte Carlo paths are shown to have moved northward. To prevent this from happening and improve its efficiency, we restrict the velocity distribution of Monte Carlo paths to the tendency to move southward. We change the range U_f of the final speed v_f to $\big\{(v_\phi, v_\lambda); V_\phi(x_A) - 3 \leq v_\phi \leq V_\phi(x_A) + 3, V_\lambda(x_A) - 2 \leq$

$v_\lambda \leq V_\lambda(x_A) + 2\big\}$. We call this simulation TRMC (restricted) in Fig. 4.6. Table 4.2 shows that the probabilities of TRMC and TRMC (restricted) agree within error bars. Because the number of unnecessary Monte Carlo paths moving northward decreases, TRMC (restricted) is more efficient than TRMC. More severe constraint $v_\lambda \geq 0$, however, causes a small bias in the estimated probabilities: see TRMC ($v_\lambda \geq 0$) in Table 4.2. Fig. 4.7 also shows that our algorithm converges correctly on increasing the number of Monte Carlo paths M.

So far, we consider the case that the number of time steps from the initial position to Tokyo ($N = 16$) is precisely known. We can relax this assumption, but we should be careful with a limitation of a discrete time model. A typhoon can pass nearby Tokyo, for example, between $N = 15$ and $N = 16$, which causes an underestimation of the actual risk when we only consider hit at integral time steps. A way to reduce this effect is to develop models with smaller steps, while it is also possible to introduce some initialization or interpolation method into a backward simulation. However, we leave this as a future problem, because this study aims to check whether the concept of backward simulation is mathematically valid.

Table 4.2: Comparison among TRMC, TRMC (restricted), TRMC (no weight), and FS for stochastic typhoon model.

Case I

Method	$P(X_N \in A)$	σ_s	ρ_1	ρ_2
TRMC	6.514×10^{-4}	0.009×10^{-4}	7.3	4.2
TRMC (restricted)	6.501×10^{-4}	0.007×10^{-4}	13.5	7.9
TRMC ($v_\lambda \geq 0$)	6.424×10^{-4}	0.007×10^{-4}	–	–
TRMC (no weight)	0.805×10^{-4}	$< 10^{-7}$	–	–
FS	6.568×10^{-4}	0.026×10^{-4}	1.0	1.0

Case II

Method	$P(X_N \in A)$	σ_s	ρ_1	ρ_2
TRMC	1.631×10^{-4}	0.002×10^{-4}	29.0	16.4
TRMC (no weight)	0.202×10^{-4}	$< 10^{-7}$	–	–
FS	1.630×10^{-4}	0.012×10^{-4}	1.0	1.0

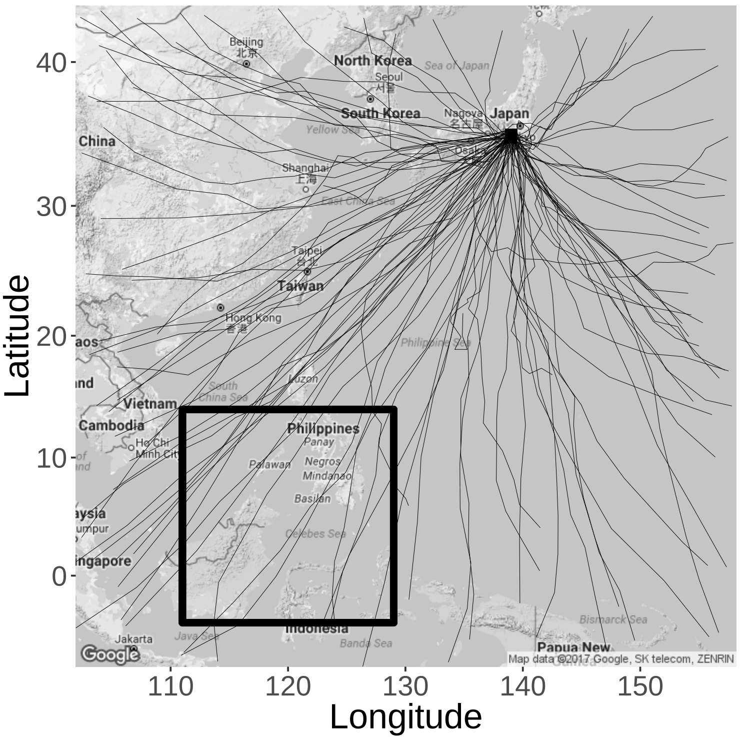


Figure 4.6: Example of Monte Carlo paths generated by TRMC starting from Tokyo. The velocity distribution is restricted to the tendency to move southward. Each line corresponds to a path generated by TRMC. The black rectangular region corresponds to the possible initial position of typhoons in the northwestern part of the Pacific Ocean.

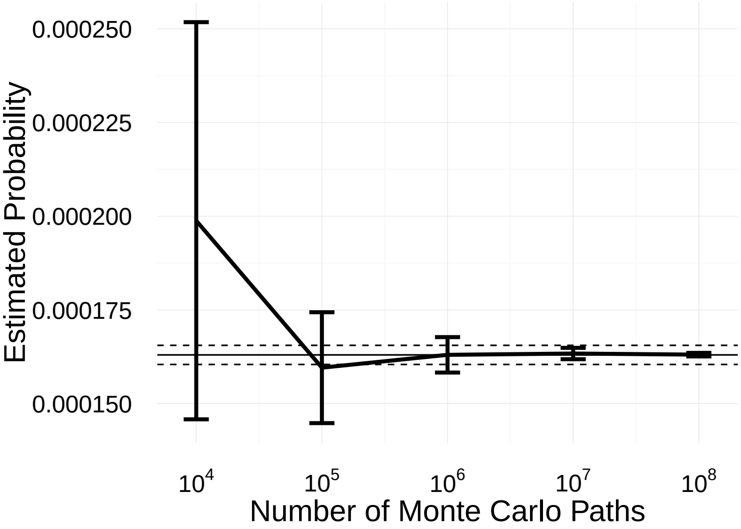


Figure 4.7: Convergence of TRMC for the stochastic typhoon model in the smaller probability case. The estimated probabilities converge to those obtained by FS as the number of Monte Carlo paths increases. Error bars indicate approximate ±1 standard error confidence intervals for TRMC. The horizontal solid line indicates the estimated probability by FS. The horizontal dashed line represents ±1 standard error confidence intervals for FS. FS has the same number of Monte Carlo paths, $M = 10^8$, as TRMC.

4.3 Lorenz 96 Model

As a higher-dimensional example, we evaluate the efficiency of our algorithm for the Lorenz 96 model [48, 49, 50]. The Lorenz 96 model is an atmospheric model and was introduced by Edward Lorenz in 1996. This model is defined as the set of coupled ordinary differential equations.

$$\begin{aligned} \frac{dx_k}{dt} &= f_k(x) + \epsilon_k, \\ f_k(x) &= -x_{k-2}x_{k-1} + x_{k-1}x_{k+1} - x_k + F, \\ k &= 1 \dots K, \\ x_k &= x_{k+K} = x_{k-K}, \forall k \end{aligned} \tag{4.5}$$

where $x = \{x_k; k = 1 \dots K\}$ is the state of the system and F is a constant. x is regarded as some atmospheric quantity in K sectors of a latitude circle (Fig. 4.8).

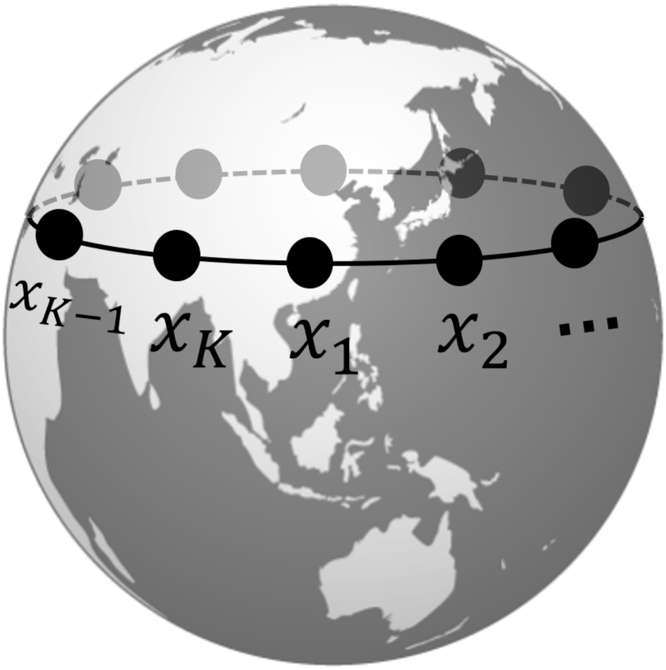


Figure 4.8: A conceptual diagram of the Lorenz 96 Model. $x = \{x_k; k = 1 \dots K\}$ is regarded as some atmospheric quantity in K sectors of a latitude circle.

The first two terms and the third term in $f_k(x)$ represent an advection and damping term, respectively. F represents an external forcing term. We set $K = 9$ and introduce Gaussian noise ϵ_k with mean zero and variance σ^2. Here, we choose $F = 8$, a value known to cause weak chaotic behavior and often used as a benchmark in data assimilation [51].

To simulate Eq. (4.5), we have to discretize it. While many discretization schemes are available, we focus on the simplest and most common scheme, the Euler scheme. The time-discretized version of Eq. (4.5) by the Euler scheme is

$$x_{k,i+1} = x_{k,i} + f(x_i)\Delta t + \epsilon_k \Delta t, k = 1 \ldots K, \tag{4.6}$$

where we set Δt to 0.001 and σ to $0.1/\sqrt{\Delta t}$.

The target region A is a K-dimensional hypercube $\{(x_1, \ldots, x_K)| - 5.0 \leq x_i \leq 7.0; i = 1 \ldots K\}$. We also assume that the initial state for x_i is uniformly distributed in $D = \{(x_1, \ldots, x_K)|1.5 \leq x_i \leq 8.5; i = 1 \ldots K\}$. We set M to 10^7 and N to 100.

We conduct this simulation on the environment described in Sect. 4.2. The computational time of TRMC in this simulation for generating 10^7 Monte Carlo paths is around 3.0×10^3 seconds.

Table 4.3 shows the result of computational experiments for the Lorenz 96 model. It shows that the probabilities of TRMC and FS agree within the error bars. The case where we ignore the factor defined by Eq. (3.4) does not reproduce the same unbiased probability as the other computational experiments. The result shows that TRMC can perform better for estimating the probabilities in the high-dimensional case. TRMC is 5.18 times more efficient than FS in terms of ρ_2, and 8.23 times in $\rho 1$ more efficient in Table 4.3. Fig. 4.9 shows the convergence of TRMC when the number of Monte Carlo paths M increases. It reveals that our algorithm converges correctly on increasing the number of Monte Carlo paths M.

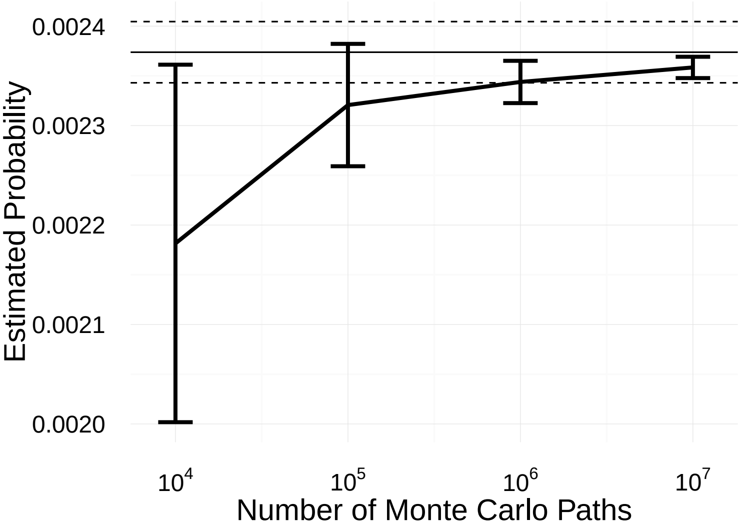


Figure 4.9: Convergence of TRMC for the Lorenz 96 model. The estimated probabilities converge to those obtained by FS as the number of Monte Carlo paths increases. Error bars indicate approximate ±1 standard error confidence intervals for TRMC. The horizontal solid line indicates the estimated probability by FS. The horizontal dashed line represents ±1 standard error confidence intervals for FS. FS has the same number of Monte Carlo paths, $M = 10^7$, as TRMC.

Table 4.3: Comparison among TRMC, TRMC (no weight), and FS for the Lorenz 96 model.

Method	$P(X_N \in A)$	σ_s	ρ_1	ρ_2
TRMC	2.358×10^{-3}	0.005×10^{-3}	8.23	5.18
TRMC (no weight)	0.957×10^{-3}	$< 10^{-6}$	–	–
FS	2.373×10^{-3}	0.015×10^{-3}	1.00	1.00

Chapter 5

Improved Schemes

5.1 Higher-order Approximation

In this section, we propose a higher-order approximation in backward dynamics; it traces paths generated by Eq. (3.6) more accurately. Hereafter, we denote this algorithm as TRMC (HO). This algorithm is effective when the amount of noise is small. Using this algorithm, the variance of the weights and estimated probabilities reduces.

To derive a higher-order approximation, we modify Eq. (3.6) as follows

$$X_i = X_{i+1} - \tilde{f}^{(2)}\left(X_{i+1}\right)\Delta t + \epsilon_i \sqrt{\Delta t} \tag{5.1}$$

where $\tilde{f}^{(2)}(x)$ is a second-order approximation of $f(X_i)$ defined as

$$\tilde{f}^{(2)}(x) = f\left(x - f(x)\Delta t\right). \tag{5.2}$$

For the n-order approximation, we recursively define the following equations

$$\tilde{f}^{(n)}(x) = \begin{cases} f(x) & (n = 1) \\ f\left(x - \tilde{f}^{(n-1)}(x)\Delta t\right) & (\text{otherwise}) \end{cases}. \tag{5.3}$$

Using this higher-order approximation, we run the backward simulation for the stochastic difference equation. The parameters are the same as case I with a larger number of time steps ($N = 15$). We set the order of the higher-order approximation

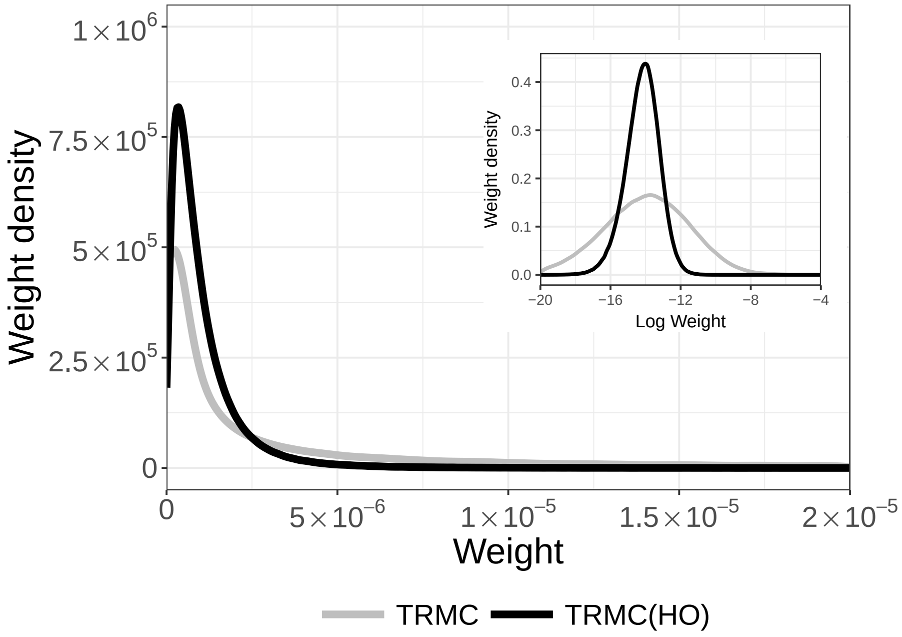


Figure 5.1: Distribution of weight $\prod_{i=0}^{N-1} W_i$ in TRMC and TRMC (HO) for the stochastic difference equation(4.3). The variance of the distribution by the TRMC (HO) is much smaller than that by TRMC.

as $n = 2$ in the following simulation. The result is shown in Table 5.1 and indicates that TRMC (HO) is much more efficient in this case.

We also show the weight distribution of TRMC and TRMC (HO) in Fig.5.1. As we expect, the variance of the distribution by TRMC (HO) is smaller than that by TRMC, which leads to the efficient estimation of probabilities.

Table 5.1: Comparison between TRMC, TRMC (HO), and the forward simulation for the stochastic difference equation (4.3)

Method	$P(X_N \in A)$	σ_s	ρ_1	ρ_2
TRMC	2.305×10^{-4}	0.063×10^{-4}	0.6	0.3
TRMC (HO)	2.324×10^{-4}	0.003×10^{-4}	181.7	85.5
FS	2.273×10^{-4}	0.048×10^{-4}	1.0	1.0

5.2 Resampling

Let us consider cases with a larger number of time steps. The proposed algorithm may not always work efficiently in this situation. For example, we consider the case where N is equal to 500 in the Lorenz 96 model (Fig. 5.2); these weights are normalized such that their sum is 1, i.e., $\sum_{j=1}^{M} W^{(j)} = 1$. The inset located at the top right of the figure shows the graph with a logarithmic scale on the x-axis. This style is also used in Fig. 5.4. The weight distribution corresponding to $N = 500$ in Fig. 5.2 has a heavy-tailed distribution. This phenomenon is referred to as degeneracy, and it means that the weights become unbalanced, and a few weights dominate all the others. This consequently causes a decrease in computational efficiency [1].

We introduce an improved scheme to solve this problem, which is realized by resampling (see Sect. 2.2). Hereafter, we denote it as TRMC (RS). This algorithm is effective when both the number of time steps and the amount of noise are large.

Note that our algorithm is based on time-reversed dynamics and uses SMC differently from the previous studies [4, 52, 49, 53, 54] on rare event sampling.

We assume that the resampling procedure modifies the weight at s time step

$$\prod_{i=0}^{s-1} W_i \tag{5.4}$$

of each Monte Carlo path to an unweighted one by eliminating Monte Carlo paths having small weights and by multiplying Monte Carlo paths having large weights.

We denote the jth Monte Carlo path as $x^{(j)} = \left\{ x_0^{(j)}, \ldots, x_s^{(j)} \right\}$. The procedure of resampling is as follows:

1. Define normalized weights

$$\tilde{W}^{(j)} = \frac{\prod_{i=0}^{s-1} W_i^{(j)}}{\sum_{j=1}^{M} \prod_{i=0}^{s-1} W_i^{(j)}}.$$

2. Resample M times with replacement from set $\left\{ x^{(j)} \right\}_{j=1}^{M}$ of Monte Carlo paths, where the probability of sampling set of $x^{(j)}$ is proportional to $\tilde{W}^{(j)}$.

After a resampling step, Monte Carlo paths $\left\{ x^{(j)} \right\}_{j=1}^{M}$ and associated weights $\left\{ W^{(j)} \right\}_{j=1}^{M}$ are replaced by the set of replicated Monte Carlo paths with an equal importance

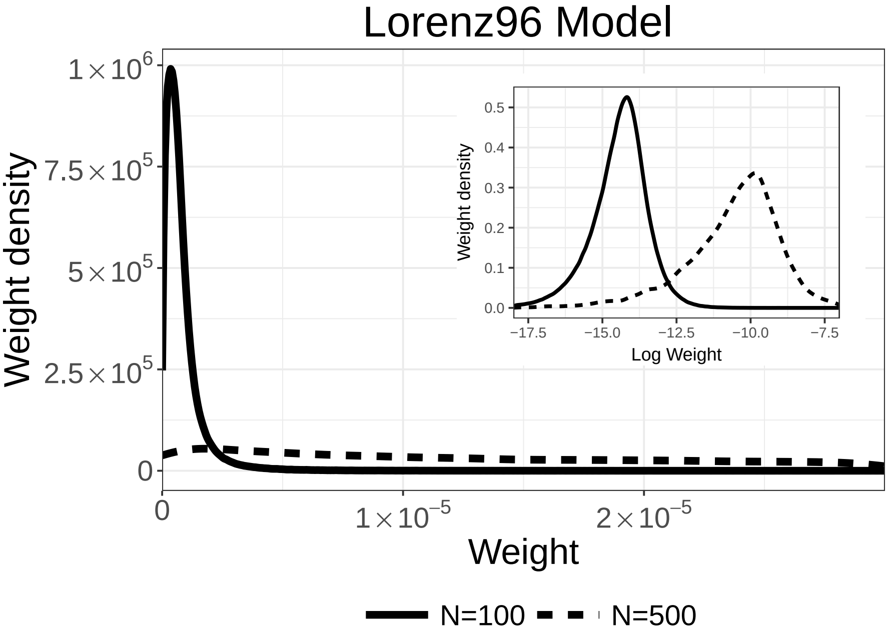


Figure 5.2: Distribution of weight $\prod_{i=0}^{N-1} W_i$ in TRMC with different numbers of time steps. The vertical and horizontal lines indicate the weight density and the value of weights, respectively. The weight distributions with a large number of time steps have a heavy-tailed distribution.

weight $W^{(j)} = \frac{1}{M}\sum_{j=1}^{M}\prod_{i=0}^{s-1} W_i^{(j)}$. Degeneracy is estimated using the effective sample size [17, 20]:

$$M_{eff} = \frac{1}{\sum_{j=1}^{M}(\tilde{W}^{(j)})^2}. \tag{5.5}$$

A small value of M_{eff} corresponds to high degeneracy. Hence, a resampling procedure is performed when this value is lower than a certain threshold $\Theta = \alpha M$, where α is a relative threshold. That is, a resampling procedure is performed when $\frac{M_{eff}}{M} < \alpha$.

Using this resampling, we simulate the Lorenz 96 model with $\sigma = 0.3/\sqrt{\Delta t}$, which is larger than that in Sect. 4.3. We set the threshold α to $0.05, 0.5$, and 0.9. The simulations with these threshold values of α are denoted by $\alpha = 5\%, 50\%$, and 90% respectively. We can use the Eq. (3.1) to evaluate probabilities even in this case.

Table 5.2 shows the result of computational experiments for the Lorenz 96 model. It shows that the probabilities of FS, TRMC, and TRMC (RS, α=50%) agree within the error bars.

Table 5.2: Comparison among TRMC, TRMC (RS, α=50%), and FS for the Lorenz 96 model.

Method	$P(X_N \in A)$	σ_s	ρ_1	ρ_2
TRMC	2.500×10^{-3}	0.025×10^{-3}	2.1	4.08
TRMC (RS, α=50%)	2.616×10^{-3}	0.020×10^{-3}	3.2	6.09
FS	2.504×10^{-3}	0.050×10^{-3}	1.0	1.0

On the other hand, Fig. 5.3 shows that TRMC (RS) is more efficient than TRMC in a wide range of threshold values. We also show the weight distributions of TRMC and TRMC (RS) in Fig. 5.4. The variance of the distribution is much smaller for TRMC (RS) than for TRMC.

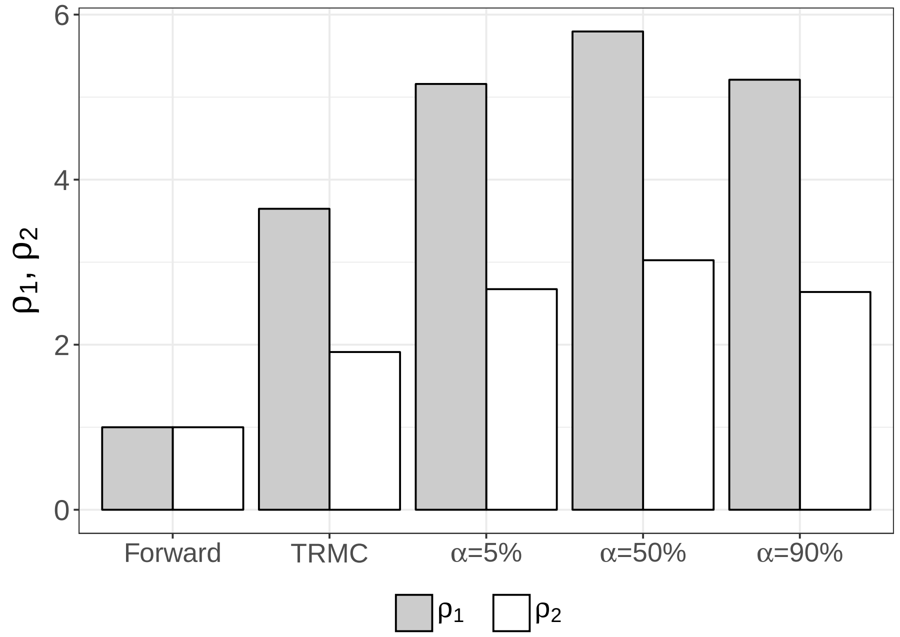


Figure 5.3: Comparison among FS (Forward), TRMC, and TRMC(RS) for the Lorenz 96 model. $\alpha = \alpha_0\%$ means TRMC(RS) with $\alpha = \frac{\alpha_0}{100}$.

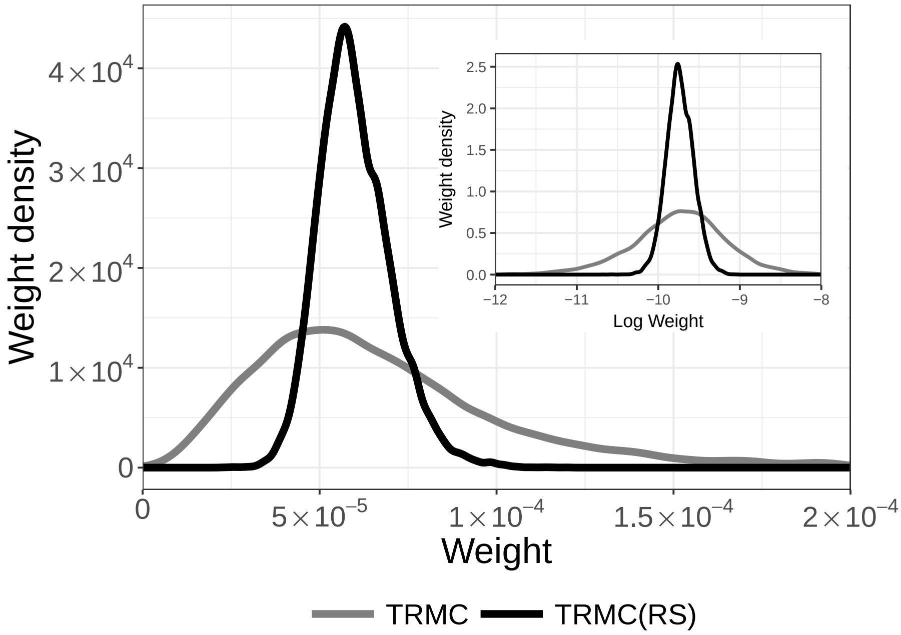


Figure 5.4: Distribution of the weight $\prod_{i=0}^{N-1} W_i$ in TRMC and TRMC (RS) for the Lorenz 96 model. We set the threshold α to 0.5 for TRMC (RS). The variance of the distribution is much smaller for TRMC (RS) than for TRMC.

5.3 External Field

As we already saw in Sect. 4.2, there are many paths that do not hit the target region A, even though the support of the initial distribution $p(x_0)$ is much bigger than the target region A. In an exact backward dynamics, as we see later in Appendix.A,

$$\sigma(T-t,X_t)\frac{\nabla(p(T-t,X_t)\sigma(T-t,X_t))^T}{p(T-t,X_t)} \tag{5.6}$$

must be considered in a drift term to simulate an exact backward dynamics for multidimensional cases. To close our backward dynamics Eq.(3.6) to an exact backward dynamics, we add an external field (EF) which induce these non hit paths in the direction of the target region A. Assuming that the $\sigma(t,x)$ is constant, this term can be written as

$$\Sigma\nabla\log(p(T-t,X_t)), \tag{5.7}$$

where $\Sigma=\sigma\sigma^T$. By adding Eq. (5.7) as the external field to the backward dynamics, we expect that this will causes a increase in computational efficiency because the dynamics can be closer to an exact backward dynamics:

$$X_i = X_{i+1} + \left(-f\left(X_{i+1}\right) + \Sigma\nabla\log(p((i+1)\Delta t, X_{i+1}))\right)\Delta t + \epsilon_i\sqrt{\Delta t}. \tag{5.8}$$

But implementation of these quantity, however, requires to know the probability distribution $p(t,x)$, which is usually not available prior to the simulations. Therefore, as one of the simplest choices, we approximate the probability $p(t,x)$ by Gaussian distributions in this thesis

$$p(t,x) \propto \exp\left[-\frac{1}{2}\left(x-((1-\frac{t}{T})x_A+\frac{t}{T}x_B)\right)^T(\Sigma\beta(t+\alpha))^{-1}\left(x-((1-\frac{t}{T})x_A+\frac{t}{T}x_B)\right)\right], \tag{5.9}$$

where x_A is the center of the target region A. x_B is the center of the support B of the initial distribution $p(x_0)$. α, β are positive parameters which control the strength of an external field.

By a simple calculation, we can prove

$$\Sigma\nabla\log p(t,x) = -\frac{1}{\beta(t+\alpha)}\left(x-\left((1-\frac{t}{T})x_A+\frac{t}{T}x_B\right)\right). \tag{5.10}$$

We introduce the factor defined by Eq. (5.10) as an external field to the backward dynamics (3.6). In this case, we can transform the backward dynamics (3.6) as the following

$$X_i = X_{i+1} + \left(-f\left(X_{i+1}\right) + \frac{1}{\beta(i\Delta t + \alpha)}\left(x - \left((1 - \frac{i}{N})x_A + \frac{i}{N}x_B\right)\right)\right)\Delta t + \epsilon_i\sqrt{\Delta t}. \tag{5.11}$$

Hereafter, we refer TRMC with this backward dynamics as TRMC (EF). TRMC and TRMC (EF) are the same in the limit where $\alpha \to \infty$ or $\beta \to \infty$. In TRMC (EF), the external field pulls the path in the direction of a target region A. As a result, TRMC (EF) forcefully increases the probability which the simulation paths hit a target region A.

Using TRMC (EF), we simulate the stochastic typhoon model explained in Sect. 4.2. We set the control parameter α to $100, 50, 10, 5, 3$ and 1. The simulations with these control values of α are denoted by $\alpha = 100, 50, 10, 5, 3$ and 1 respectively. We set the control parameter β to 0.5 and treat as fixed value here. We can estimate β from the target region A and the number of time steps N.

Table 5.3 shows the result of computational experiments for the stochastic typhoon model with TRMC (EF). It shows that the probabilities of FS, TRMC, and TRMC (EF, $\alpha = 10$) agree within the error bars. On the other hand, TRMC (EF, $\alpha = 1$) does not reproduce the unbiased estimates of the probability. This is because that the strong external field compared to the support of the initial distribution $p(x_0)$ occurs bias. Fig. 5.5 shows the estimated probabilities for FS, TRMC, and TRMC (EF) with several α.

Table 5.3: Comparison among TRMC and TRMC (EF) for the stochastic typhoon model.

Method	$P(X_N \in A)$	σ_s	ρ_1	ρ_2
TRMC	6.598×10^{-4}	0.009×10^{-4}	7.2	6.9
TRMC (EF, $\alpha = 10$)	6.593×10^{-4}	0.004×10^{-4}	49.1	46.6
TRMC (EF, $\alpha = 1$)	6.606×10^{-4}	0.110×10^{-4}	0.1	0.1
FS	6.626×10^{-4}	0.026×10^{-4}	1.0	1.0

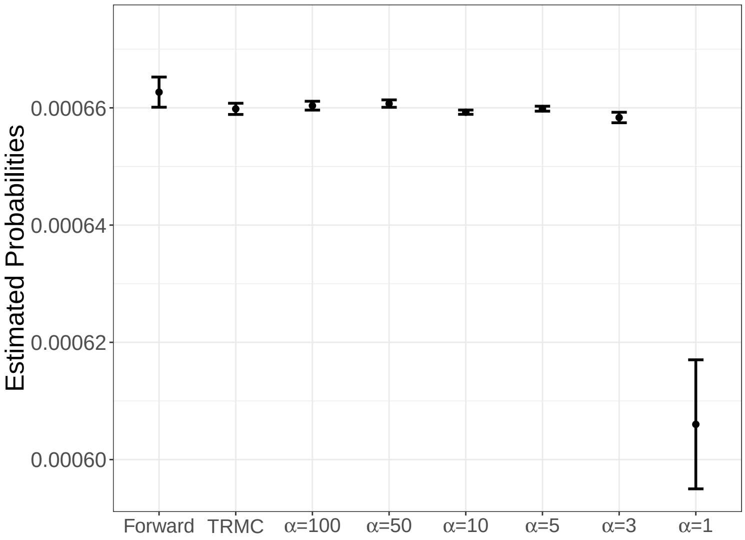


Figure 5.5: Comparison of the estimated probabilities among FS (Forward), TRMC, and TRMC (EF) for the stochastic typhoon model. $\alpha = \alpha_0$ means TRMC (EF) with $\alpha = \alpha_0$. TRMC (EF) is more efficient than TRMC in a wide range of threshold values. Error bars indicate approximate ±1 standard error confidence intervals. TRMC (EF, $\alpha = 1$) does not reproduce the unbiased estimates of the probability because of the strong external field.

Fig. 5.6 shows that TRMC (EF) is more efficient than TRMC in a wide range of threshold values. We also show the weight distributions of TRMC and TRMC (EF) in Fig. 5.7. The variance of the distribution is much smaller for TRMC (EF) than for TRMC. Obviously, TRMC (EF) does not has a heavy tailed distribution compared to TRMC. Fig. 5.8 shows the number of a hit when α changes. It reveals that the number of a hit tends to increase when α decreases as we expected.

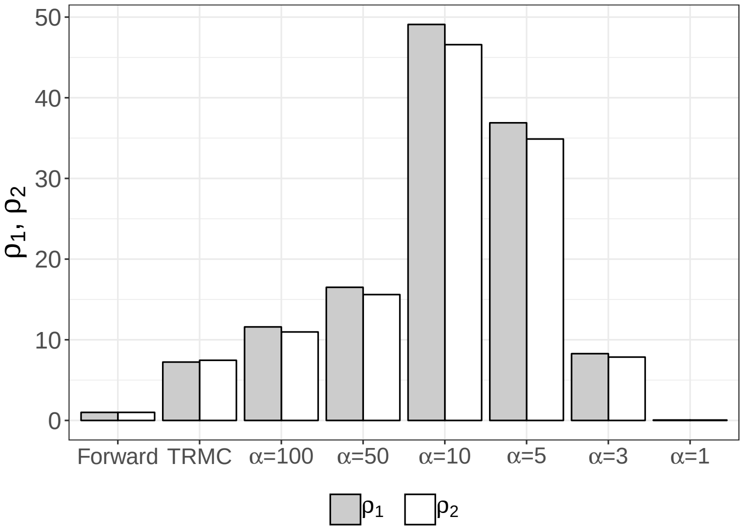


Figure 5.6: Comparison among FS (Forward), TRMC, and TRMC (EF) for the stochastic typhoon model. $\alpha = \alpha_0$ means TRMC (EF) with $\alpha = \alpha_0$. TRMC (EF) is more efficient than TRMC in a wide range of threshold values.

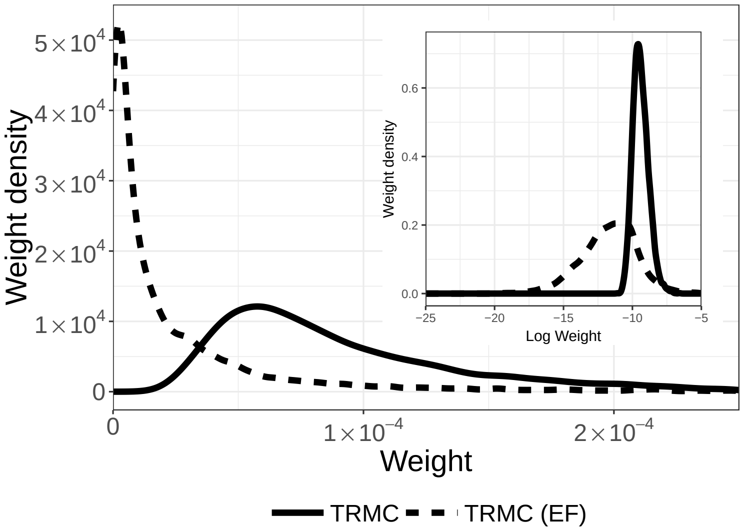


Figure 5.7: Distribution of the weight $\prod_{i=0}^{N-1} W_i$ in TRMC and TRMC (EF) for the stochastic typhoon model. We set the threshold α to 10 for TRMC (EF). The variance of the distribution is much smaller for TRMC (EF) than for TRMC.

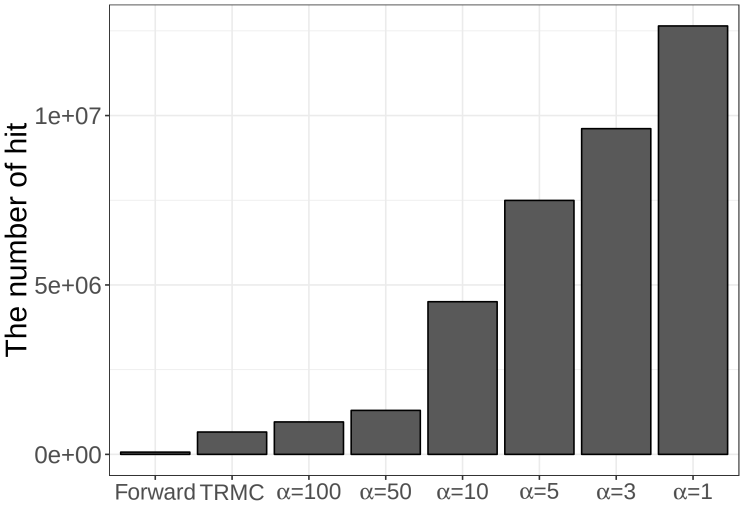


Figure 5.8: The number of a hit when α changes for the stochastic typhoon model. $\alpha = \alpha_0$ means TRMC (EF) with $\alpha = \alpha_0$. The number of a hit tends to increase when α decreases as we expected.

Examples of Monte Carlo paths for TRMC and TRMC (EF) are shown in Figs. 5.9 and 5.10. In the case of Fig. 5.10, it is confirmed that the number of paths with extreme behavior has been reduced.

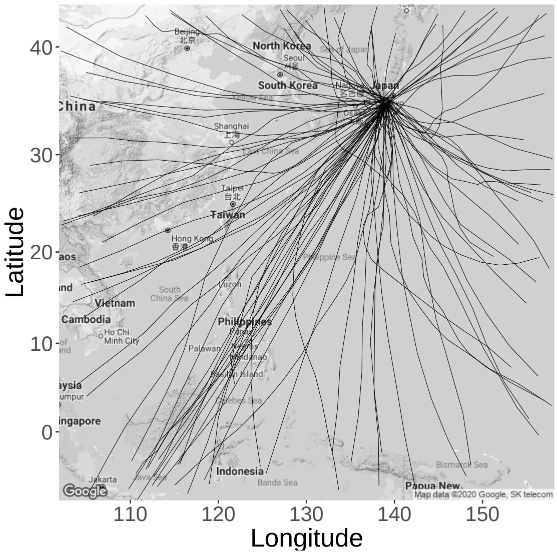


Figure 5.9: Example of Monte Carlo paths generated by TRMC starting from Tokyo. Each line corresponds to a path generated by TRMC.

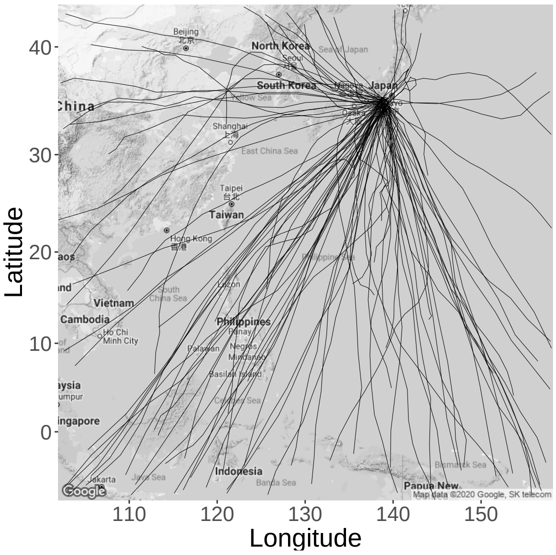
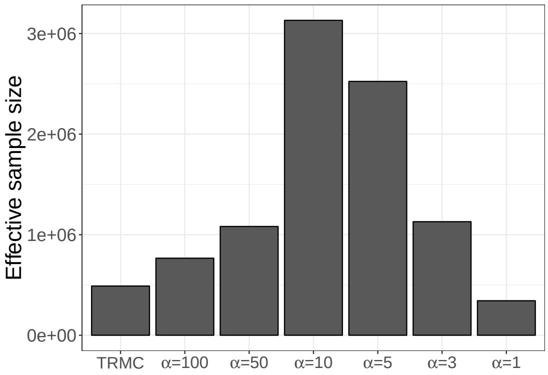


Figure 5.10: Example of Monte Carlo paths generated by TRMC (EF) starting from Tokyo. Each line corresponds to a path generated by TRMC (EF, $\alpha = 10$).

α (and β) is also a kind of hyperparameter and needs to be determined in advance of the simulation. We need an indicator of how to tune these hyperparameters. As one of the ways, we can use α (and β) where the effective sample size M_{eff} is maximal. We explicitly write the definition of the effective sample size M_{eff} used here

$$\begin{aligned} M_{eff} &= \frac{1}{\sum_{j=1}^{M}(\tilde{W}^{(j)})^2}, \\ \tilde{W}^{(j)} &= \frac{\left\{\prod_{i=0}^{N-1} W_i^{(j)}\right\} V_A p(x_0^{(j)})}{\sum_{j=1}^{M}\left\{\prod_{i=0}^{N-1} W_i^{(j)}\right\} V_A p(x_0^{(j)})}. \end{aligned} \tag{5.12}$$

Fig. 5.11 shows the α dependency of M_{eff}. The effective sample size becomes the largest when $\alpha = 10$, which is consistent with the tendency of computational efficiency. The other way is to use the effective sample size ratio defined by effective sample size M_{eff} divided by the number of a hit. Fig. 5.12 shows the α dependency of the effective sample size ratio. In this method, we can use α before the sharp decrease of the effective sample size ratio. This result is also consistent with the tendency of computational efficiency.



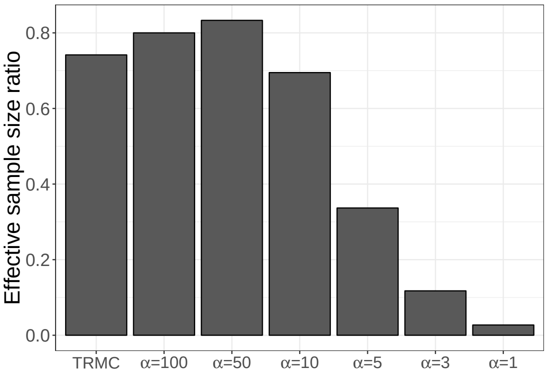


Figure 5.11: The α dependency of M_{eff} for the stochastic typhoon model. $\alpha = \alpha_0$ means TRMC (EF) with $\alpha = \alpha_0$. The effective sample size becomes the largest when $\alpha = 10$, which is consistent with the tendency of computational efficiency.

Figure 5.12: The α dependency of the effective sample size for the stochastic typhoon model. $\alpha = \alpha_0$ means TRMC (EF) with $\alpha = \alpha_0$.

Chapter 6

Discussion and Concluding Remarks

6.1 Discussion

The examples provided in the preceding sections show that backward simulations using TRMC provide unbiased estimates of the probabilities and can be more efficient than forward simulations. In these examples, the computational efficiencies of TRMC are 3–16 times higher than those obtained by forward simulation, when the calculated probability of hitting the target is 2×10^{-3}–10^{-5}. Note that TRMC can be used to calculate the probability for an arbitrarily small target region; this would be impossible by using forward simulation.

There are, however, cases in which TRMC is inefficient. First, TRMC is not advantageous if the time-reversed paths rarely encounter a region in which the initial density $p(x_0)$ is high; this can occur when the initial density is not broad. Another case in which TRMC can be inefficient is when the weight in Eq. (3.8) (or, in the continuous time version, Eq. (3.9)) is highly time dependent. If paths with smaller weights in the initial stage of backward simulation acquire larger weights in the latter stage, resampling of the path (particle splitting) in SMC may not be effective. In this case, if TRMC with SIS is ineffective, TRMC with SMC also shows poor performance.

To discuss the possible improvement of the algorithm, it is useful to introduce optimal backward dynamics. Although it is not easy to obtain these dynamics *a priori*, the formal definition is derived as follows. First, the marginal probability at step n obtained from the forward simulation is defined as

$$p(x_n) = \int dx_{0:n-1} \left\{ \prod_{i=0}^{n-1} p\left(x_{i+1}|x_i\right) \right\} p(x_0), \tag{6.1}$$

which satisfies the relation

$$p(x_{i+1}) = \int p(x_{i+1}|x_i) p(x_i) dx_i. \tag{6.2}$$

Using Eqs. (6.1) and (6.2), the transition probability q^* of the optimal backward dynamics is defined as

$$q^*(x_{i+1} \to x_i) = \frac{p(x_{i+1}|x_i)p(x_i)}{\int p(x_{i+1}|x_i)p(x_i)dx_i} = \frac{p(x_{i+1}|x_i)p(x_i)}{p(x_{i+1})}.. \tag{6.3}$$

Note that Eq. (6.3) appears similar to the formulas used in Bayesian inference when the probability $p(x_i)$ obtained by forward simulations is regarded as an analog of the prior distribution of x_i. In terms of the selection of Eq. (6.3) for backward dynamics, the following relation holds

$$\left\{ \prod_{i=0}^{n-1} p\left(x_{i+1}|x_i\right) \right\} p(x_0) = p(x_N) \prod_{i=N-1}^{0} q^*(x_{i+1} \to x_i). \tag{6.4}$$

Eq. (6.4) means that the combined probability of time-reversed paths defined by forward simulation is recovered by the backward dynamics Eq. (6.3). Specifically, the time-reversed paths initialized by $p(x_N)$ automatically converge to their initial density $p(x_0)$ using the backward dynamics Eq. (6.3). In this sense, $q^*(x_{i+1} \to x_i)$ in Eq. (6.3) is considered as the optimal backward dynamics. This idea is known in Monte Carlo studies [2]. Implementation of these dynamics, however, requires the probabilities $p(x_i), i = 1, \dots N$, which are usually not available prior to the simulations.

Note that the backward dynamics defined by the Langevin equation in previous studies [8, 9, 10] can be derived from Eq. (6.3) as a continuous-time limit.

Equations (6.3) and (6.4) were previously discussed [11, 12, 13] in the field of time-series data analysis, where approximations of the marginal probabilities

$p(x_i), i = 1, \dots N$ are used to define the backward dynamics $q(x_{i+1} \to x_i)$. In these studies, the observed data were available at many of the time steps $i = 0, \dots, N$, whereas the target is given only at $i = N$ in our problem. Then, approximations of probabilities $p(x_i), i = 1, \dots N$ are derived using forward simulations constrained with the observed data ("filtering stage").

It is, however, difficult to apply these methods to our problem. If we were to apply a similar method to our problem, we would have to run a number of forward simulations to estimate $p(x_i), i = 1, \dots N$ before executing backward simulations. This would be computationally expensive and seems unrealistic without a highly efficient method for the probability estimation. Methods such as those discussed previously [55, 56] may be applied to optimize the backward dynamics in our problem, but this is left for future study.

On the other hand, when some observed data are available outside the equations that describe the stochastic process, we may use these data to approximate $p(x_i), i = 1, \dots N$ and hence use them to approximate the optimal backward dynamics. In this case, we avoid the use of a large amount of forward computation to construct the optimized backward dynamics. This seems possible for the stochastic typhoon model, where data from actual observations of real typhoons are available. Note that this idea is different from data assimilation (i.e., inference with simulations combined with observed data), because here we use observed data only to improve the computational efficiency; they do not cause the bias of calculated probabilities.

As a practical approach, it is possible to approximate $q^*(x_{i+1} \to x_i)$ as follows

$$\begin{aligned} q(x_{i+1} \to x_i) &= \frac{p(x_{i+1}|x_i)p(x_i)}{p(x_{i+1})} \\ &\approx \exp\left[-\frac{1}{2}\left(x_{i+1} - x_i - f(x_i)\Delta t\right)^T \left(\Sigma\Delta t\right)^{-1}\left(x_{i+1} - x_i - f(x_i)\Delta t\right) - \log\frac{p(x_i)}{p(x_{i+1})}\right] \\ &= \exp\left[-\frac{1}{2}\left(\Delta x_i - f(x_i)\Delta t\right)^T \left(\Sigma\Delta t\right)^{-1}\left(\Delta x_i - f(x_i)\Delta t\right) - \Delta\log p(x_i)\right] \\ &= \exp\left[-\frac{1}{2}\left(\Delta x_i - \left(f(x_{i+1}) - \Sigma\nabla\log p(x_{i+1})\right)\Delta t\right)^T \left(\Sigma\Delta t\right)^{-1}\left(\Delta x_i - \left(f(x_{i+1}) - \Sigma\nabla\log p(x_{i+1})\right)\Delta t\right) + O(\Delta t)\right], \end{aligned} \tag{6.5}$$

where $\Delta x_i = x_{i+1} - x_i$, $\Delta \log p(x_i) = \log p(x_{i+1}) - \log p(x_i)$. The dynamics derived from Eq. (6.5) is the following stochastic difference equation

$$X_i = X_{i+1} + \left(-f\left(X_{i+1}\right) + \Sigma\nabla\log(p(X_{i+1}))\right)\Delta t + \epsilon_i\sqrt{\Delta t}. \tag{6.6}$$

Because Eq. (6.6) coincides with Eq. (5.8) after rearranging, the exact backward dynamics coincide with the optimal backward dynamics in the range of $O(\Delta t)$ under this approximation. Therefore, we can interpret TRMC (EF) as one of the approximation methods of the optimal backward dynamics.

In this thesis, we assumed that the number of time steps is fixed. This is not, however, always clear in advance in realistic problems. In the case of a realistic typhoon model, we must consider the case of passing through Tokyo between two discrete time steps. This problem is called the high-dimensional boundary crossing problem. To our best knowledge, a few research is known for the high-dimensional boundary crossing problem (see, e.g., [57, 58, 59]). Therefore, the interpolation method for our case must be developed in future work.

6.2 Concluding Remarks

We discussed methods for the backward simulation of the stochastic process. These methods trace a time-reversed path from the target region to the initial configuration. A naïve approach to this problem was shown not to function as expected. To resolve the difficulties, the time reverse Monte Carlo method (TRMC) was introduced. The TRMC method is based on SIS and SMC, and is designed to provide the probabilities of events correctly. TRMC with SIS was tested for the stochastic difference equation and the stochastic typhoon model and the Lorenz 96 model; it converges more efficiently than forward simulations in some of these examples. Three types of improved versions of TRMC are also introduced. The first one is a higher-order approximation in backward dynamics. The second one is TRMC with resampling for simulations with a larger number of steps. The third one is TRMC with an external field. In these improvements, TRMC provides unbiased estimates of the probabilities without expensive computation. These three types of improved schemes are shown to be advantageous. We also discussed the limitation and possible improvement of TRMC and its relation to the Bayes formula.

Appendix

A Time Reversal of Diffusions

In this section, we review the theory of time reversal for diffusion processes developed by Haussmann and Pardoux [9] and Föllmer [60]. For simplicity, we treat a one-dimensional system.

Let the dynamics of a stochastic process $X = \{X_u; 0 \leq u < T\}$ on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is defined by

$$X_t = X_0 + \int_0^t b(u, X_u)du + \int_0^t \sigma(u, X_u)dW_u, 0 \leq t \leq T \tag{A.1}$$

where the integral is defined as the Ito integral and $W = \{W_u; 0 \leq u < T\}$ is Brownian motion. This type of process is called "diffusion process" or "Ito process". A diffusion process is a Markov process with continuous sample paths. The functions b and σ are called, respectively, the "drift coefficient" and "diffusion coefficient". Standard Brownian motion corresponds to a diffusion process with $b(t, x) = 0$ and $\sigma(t, x) = 1$.

Definition A.1 *The Ito integral. We assume that X_t, Y_t be an adapted stochastic process. Let $\Pi = \{t_0, t_1, ..., t_n\}$ be a partition of $[0, t]$ with $0 = t_0 < t_1 < \cdots < t_n = t$, then define the sum*

$$I(t) = \sum_{j=0}^{n-1} Y_{t_{j+1}}(X_{t_{j+1}} - X_{t_j}). \tag{A.2}$$

The stochastic process I(t) in Eq.(A.2) *is the Ito integral. $I(t)$ converges in probability as the mesh $||\Pi||$ of the partition tends to zero if the limit exists. In this case, the Ito integral is written as*

$$I(t) = \int_0^t Y_u dX_u. \tag{A.3}$$

Given a stochastic process X, we define the time-reversed process

$$\tilde{X}_t = X_{T-t}. \tag{A.4}$$

Using Eq. (A.1), we can write

$$X_{T-t} = X_T + \int_T^{T-t} b(u, X_u)du + \int_T^{T-t} \sigma(u, X_u)dW_u. \tag{A.5}$$

Eq.(A.5) can be written with time-reversed process $\tilde{X}$ as

$$\tilde{X}_t = \tilde{X}_0 - \int_0^t b(T-s, \tilde{X}_s)ds - \int_0^t \sigma(T-s, \tilde{X}_s)dW_{T-s}, \tag{A.6}$$

where we make the substitution $s = T - u$. The stochastic integral $\int_0^t \sigma(T - s, \tilde{X}_s)dW_{T-s}$ in Eq.(A.6) is defined as

$$\sum_{i=0}^{n-1} \sigma(T - t_{i+1}, \tilde{X}_{t_{i+1}})(W_{T-t_i} - W_{T-t_{i+1}}), \tag{A.7}$$

where we choose partition points

$$0 = t_0 < t_1 < \cdots < t_n = t. \tag{A.8}$$

We also define the time-reversed Brownian motion as the following:

$$\bar{W}_t = W_{T-t} - W_T. \tag{A.9}$$

Lemma A.1 *The Ito's Lemma. let $f(t,x)$ be a function for which the partial derivatives $f_t(t,x)$, $f_x(t,x)$, and $f_{xx}(t,x)$ are defined and continuous. Let $X = \{X_t; 0 \leq u \leq T\}$ be an Ito process defined by Eq.*(A.1). *Then, for every $T \geq 0$,*

$$\begin{aligned} f(T, X_T) = f(0, X_0) & \\ &+ \int_0^T \left(f_t(t, X_t) + f_x(t, X_t)b(t, X_t) + \frac{1}{2} f_{xx}(t, X_t)\sigma^2(t, X_t) \right) dt \\ &+ \int_0^T f_x(t, X_t)\sigma(t, X_t)dW_t. \end{aligned} \tag{A.10}$$

In shorthand differential form,

$$\begin{aligned} df(t, X_t) = &\left(f_t(t, X_t) + f_x(t, X_t)b(t, X_t) + \frac{1}{2} f_{xx}(t, X_t)\sigma^2(t, X_t) \right) dt \\ &+ f_x(t, X_t)\sigma(t, X_t)dW_t. \end{aligned} \tag{A.11}$$

Applying Ito's lemma to $\sigma(t, X_t)$, Eq. (A.7) is written as

$$\begin{aligned}
&\sum_{i=0}^{n-1} \sigma(T-t_{i+1}, \tilde{X}_{t_{i+1}})(W_{T-t_i} - W_{T-t_{i+1}}) \\
&= -\sum_{i=0}^{n-1} \sigma(T-t_{i+1}, \tilde{X}_{t_{i+1}})(\bar{W}_{t_{i+1}} - \bar{W}_{t_i}) \\
&= -\sum_{i=0}^{n-1} \sigma(T-t_i, \tilde{X}_{t_i})(\bar{W}_{t_{i+1}} - \bar{W}_{t_i}) + \sigma(T-t_i, \tilde{X}_{t_i})\partial_x\sigma(T-t_i, \tilde{X}_{t_i})\Delta t_i, \qquad \text{(A.12)}
\end{aligned}$$

where $\Delta t_i = t_{i+1} - t_i$. Taking the limit as $n \to \infty$, we get

$$\int_0^t \sigma(T-s, \tilde{X}_s) dW_{T-s} = \int_0^t \sigma(T-s, \tilde{X}_s) d\bar{W}_s + \int_0^t \sigma(T-s, \tilde{X}_s)\partial_x\sigma(T-s, \tilde{X}_s) ds. \qquad \text{(A.13)}$$

Using Eq. (A.13) and (A.6), we get

$$\begin{aligned}
\tilde{X}_t = \tilde{X}_0 &+ \int_0^t \left\{ -b(T-s, \tilde{X}_s) + \sigma(T-s, \tilde{X}_s)\partial_x\sigma(T-s, \tilde{X}_s) \right\} ds \\
&+ \int_0^t \sigma(T-s, \tilde{X}_s) d\bar{W}_s \qquad \text{(A.14)}
\end{aligned}$$

It is important here to note that $\tilde{W}$ is not Brownian motion because

$$\mathbb{E}\left[\tilde{X}_t(\bar{W}_u - \bar{W}_t)\right] = \mathbb{E}\left[X_{T-t}(W_{T-u} - W_{T-t})\ \right] \neq 0, \qquad \text{(A.15)}$$

where $0 \leq t \leq u \leq T$. To find the correct Brownian motion, we prove the following relation

$$\mathbb{E}\left[G(X_t)\left(W_t - W_u + \int_u^t \frac{\partial_x(p(s, X_s)\sigma(s, X_s))}{p(s, X_s)} ds\right)\right] = 0, \qquad \text{(A.16)}$$

where $G : \mathbb{R} \to \mathbb{R}$. See [61] for the detail. Then, we define $\tilde{W}_t$ as the following

$$\tilde{W}_t = \bar{W}_t - \int_{T-t}^{T} \frac{\partial_x(p(s, X_s)\sigma(s, X_s))}{p(s, X_s)} ds, \qquad \text{(A.17)}$$

$\tilde{W}_t$ is the Brownian motion.

As Eq.(A.1) can be written in differential form

$$dX_t = b(t, X_t)dt + \sigma(t, X_t)dW_t \qquad \text{(A.18)}$$

, we can also write the exact backward dynamics in differential form as

$$d\tilde{X}_t = \tilde{b}(T-t,\tilde{X}_t)dt + \sigma(T-t,\tilde{X}_t)d\tilde{W}_t \tag{A.19}$$

$$\tilde{b}(t,x) = -b(t,x) + \sigma(t,x)\partial_x\sigma(t,x) + \sigma(t,x)\frac{\partial_x(p(t,x)\sigma(t,x))}{p(t,x)}. \tag{A.20}$$

This equation is an equation that strictly reverses time. We call this the exact backward dynamics here. We can simulate the exact backward dynamics using a numerical method such as the Euler method.

B Application to Options Portfolio Valuation

In this appendix, we introduce options portfolio valuation where TRMC can be effective as an application for finance. An option is a kind of derivative securities. Derivative securities are securities whose value is based on underlying security price (e.g., stock price, bond price, etc.). For example, the simplest option, a European call option on an underlying security, gives the holder a right (not an obligation; then, the name "option") to buy the underlying security at a specified date for a specified price (this is called the strike price, we denote by K). There are different types of options. Here, we only deal with the call/put option, which gives the holder the right to buy/sell security. See, for example, [62] for further information about derivative securities and financial engineering.

Estimating the correct value for these options is an essential and challenging problem in the financial industry. The breakthrough was occurred by Black and Scholes (1973) [63], which introduced the no-arbitrage principle and pricing-formula for generic options. Generally speaking, we can estimate the right value of the option V_T

$$V_T = \int_0^\infty dU_T \ \mathrm{payoff}(U_T) p(U_T|U_0) p(U_0), \tag{B.1}$$

where T is the maturity time of the options, $U = \{U_t; 0 \leq t \leq T\}$ is the stochastic process of an underlying security price, $p(U_T|U_0)$ is the transition probability density from U_0 to U_T, and payoff$(\cdot)$ is a payoff function depending on U_T, K, and the option types. the payoff of a call/put option are the following

$$\mathrm{payoff}(u) = \begin{cases} \max(u - K, 0) & \text{(call option)} \\ \max(K - u, 0) & \text{(put option)} \end{cases}. \tag{B.2}$$

In this appendix, we assume that there is one underlying security and the discretized dynamics of the price process $U = \{U_0, U_1, \ldots, U_N\}$ follows a log-normal model

$$U_{i+1} = rU_i\Delta t + \epsilon_i U_i \sqrt{\Delta t}, \tag{B.3}$$

where r is the annualized risk-free interest rate, $N\Delta t = T$, and the noise ϵ_i is assumed to be i.i.d. Gaussian noise with mean zero and the variance σ^2.

Monte Carlo methods are popular computational methods to estimates Eq. (B.1), especially when the portfolio includes a lot of options (options portfolio), or the underlying space has a large dimensionality [64].

An important point in the practical financial risk management of an options portfolio is that the value of an option portfolio changes non-linearly. By starting a simulation from multiple initial conditions (e.g., multiple stock prices) and calculating the value of an options portfolio for each initial condition, we can grasp the non-linear risk. Then, it is possible to estimate how much loss in the value of a portfolio will occur when the financial market falls sharply. TRMC can be efficient in evaluating derivative portfolios with payoffs under "limited conditions" as we explain later.

Here, we regard this "limited condition" as the target region A and estimate an options portfolio value by applying TRMC. In the financial industry, it is common to calculate using a forward simulation, so we use FS as a baseline in this appendix.

We fix $r = 0.01, \Delta t = 0.01, T = 1, K = 18000, \sigma = 0.2$. We also assume that initial conditions (underlying security price) U_0 is uniformly distributed in $\{u_0; 15000 \leq u_0 \leq 25000\}$. The target region A is $\{u_T; 17500 \leq u_T \leq 18500\}$ We set the number of Monte Carlo paths M to 10^6 and the number of time steps N to 100.

We can use the Black-Scholes formula to price call/put options respectively

$$\begin{aligned} V_{call,T} &= N(d_1)U_0 - N(d_2)Ke^{-rT} \\ V_{put,T} &= -N(-d_1)U_0 - N(-d_2)Ke^{-rT}, \end{aligned} \tag{B.4}$$

where $d_1 = \frac{log(\frac{U_T}{K})}{\sigma\sqrt{T}}$, $d_2 = d_1 - \sigma\sqrt{T}$ and $N(\cdot)$ is the standard normal cumulative distribution function [62].

We also designed the payoff function shown in Fig. B.1. The non-zero region of this payoff function corresponds to the target region A in TRMC. Since the payoff function introduced here can be described as a combination of call/put options, we can use Black-Scholes formula to calculate the exact solution of this simulation. As Black-Scholes formula calculates the value at a specific underlying security price U_0, we divided the initial distribution of underlying security price into bins (we set this to 40), and evaluated the options using the security price at the center of each

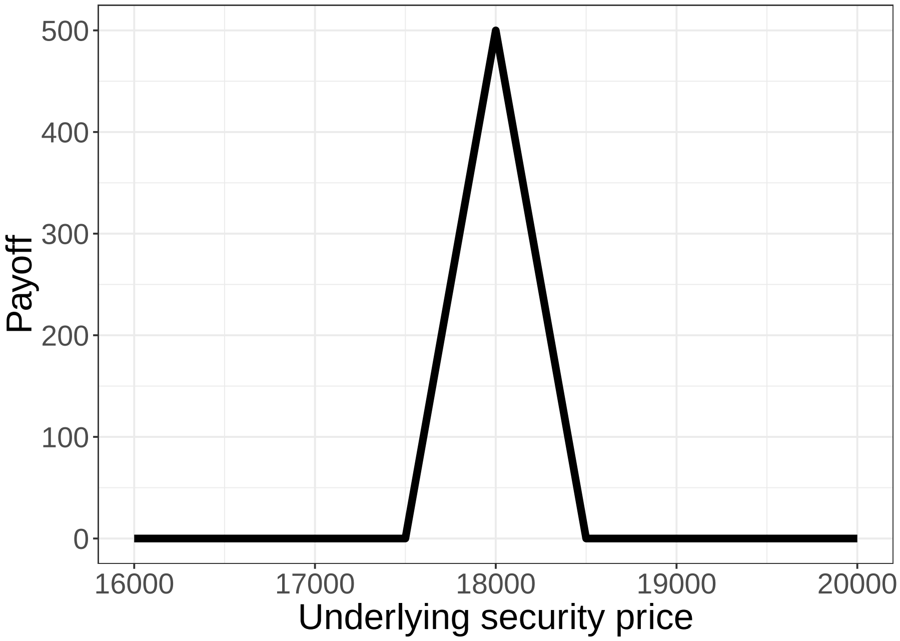


Figure B.1: The shape of the payoff function, which we use here. The non-zero region of this payoff function corresponds to the target region A in TRMC.

bin. Fig. B.2 shows the result of computational experiments for this simulation. It shows that the value calculated using TRMC and the exact solution by Black-Scholes formula almost agree within the error bars. Fig. B.3 shows that TRMC is more efficient than FS in a wide range of underlying security prices. Here, we denote a standard error of TRMC to evaluate the computational efficiency by σ_s. We also denote the standard error of FS by σ_s^F. Using these variables, as in the main text, we define a measure of the relative computational effciency ρ_2 as

$$\rho_2 = \left(\frac{\sigma_s^F}{\sigma_s}\right)^2 \frac{\tau^F}{\tau},$$

where τ is the computational time in seconds of the simulation and τ^F is the computational time of FS in seconds. This efficiency is defined in the sense of the actual performance considering both the computational time and the variance of the resulting estimates.

There are two advantageous points in TRMC compared to using Black-Scholes formula. The first one is that we can use TRMC even though when we can not

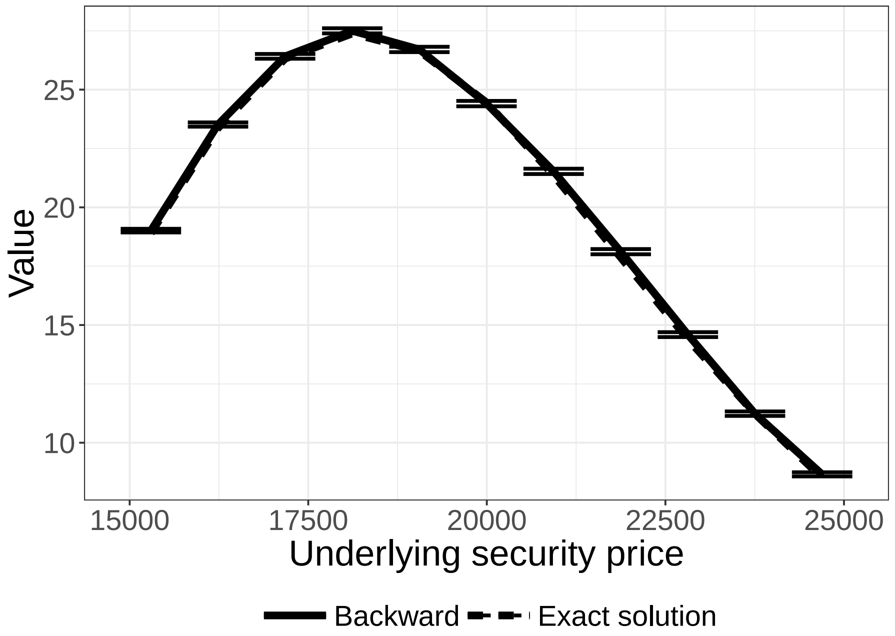


Figure B.2: Comparison among TRMC and the exact solution by BS formula for options portfolio valuation. Error bars indicate approximate 1 standard error confidence intervals for TRMC.

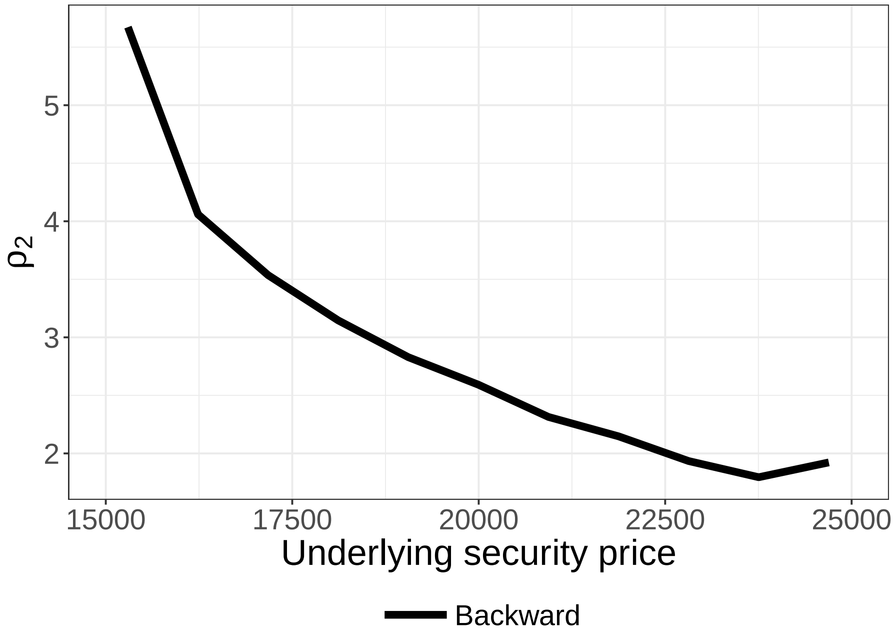


Figure B.3: TRMC is more efficient than FS in a wide range of underlying security prices. ρ_2 values in a wide range of underlying security prices.

obtain an analytical solution by Black-Scholes formula. The second point is that we can easily apply TRMC to high-dimensional systems compared to getting an analytical solution by Black-Scholes formula.

Bibliography

[1] Arnaud Doucet, Nando De Freitas, and Neil Gordon. *Sequential Monte Carlo methods in practice*. Springer-Verlag New York, New York, 2001.

[2] Jun S Liu. *Monte Carlo Strategies in Scientific Computing.* Springer-Verlag New York, New York, 2004.

[3] Arnaud Doucet and Adam M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. In *The Oxford Handbook of Nonlinear Filtering*, pages 656–704. Oxford University Press, 2011.

[4] Julien Tailleur and Jorge Kurchan. Probing rare physical trajectories with lyapunov weighted dynamics. *Nature Physics*, 3(3):203–207, 2007.

[5] Daniel M Zuckerman. *Statistical Physics of Biomolecules: An Introduction.* CRC Press, Florida, 2010.

[6] Lorenzo Donati, Carsten Hartmann, and Bettina G Keller. Girsanov reweighting for path ensembles and markov state models. *The Journal of chemical physics*, 146(24):244112, 2017.

[7] John D. Chodera, William C. Swope, Frank Noé, Jan-Hendrik Prinz, Michael R. Shirts, and Vijay S. Pande. Dynamical reweighting: Improved estimates of dynamical properties from simulations at multiple temperatures. *The Journal of chemical physics*, 134(24):244107, 2011.

[8] JE Anderson, SJ Hoffman, and CR Peters. Factors influencing reverse osmosis rejection of organic solutes from aqueous solution. *The Journal of Physical Chemistry*, 76(26):4006–4011, 1972.

[9] Ulrich G Haussmann and Etienne Pardoux. Time reversal of diffusions. *The Annals of Probability*, 14:1188–1205, 1986.

[10] Annie Millet, David Nualart, and Marta Sanz. Integration by parts and time reversal for diffusion processes. *The Annals of Probability*, 17:208–238, 1989.

[11] Mark Briers, Arnaud Doucet, and Simon Maskell. Smoothing algorithms for state–space models. *Annals of the Institute of Statistical Mathematics*, 62(1):61–89, 2010.

[12] Fredrik Lindsten and Thomas B Schön. Backward simulation methods for monte carlo statistical inference. *Foundations and Trends in Machine Learning*, 6(1):1–143, 2013.

[13] Michael Isard and Andrew Blake. A smoothing filter for condensation. In *European conference on computer vision*, pages 767–781. Springer, 1998.

[14] RC Griffiths and Simon Tavare. Simulating probability distributions in the coalescent. *Theoretical Population Biology*, 46(2):131–159, 1994.

[15] Kai Zhu and Lei Ying. Information source detection in the sir model: A sample-path-based approach. *IEEE/ACM Transactions on Networking (TON)*, 24(1):408–421, 2016.

[16] Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pages 107–113. IET, 1993.

[17] Augustine Kong, Jun S Liu, and Wing Hung Wong. Sequential imputations and bayesian missing data problems. *Journal of the American statistical association*, 89(425):278–288, 1994.

[18] D Avitzour. Stochastic simulation bayesian approach to multitarget tracking. *IEE Proceedings-Radar, Sonar and Navigation*, 142(2):41–44, 1995.

[19] Genshiro Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25, 1996.

[20] Jun S Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044, 1998.

[21] Michael K Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599, 1999.

[22] Rong Chen, Xiaodong Wang, and Jun S Liu. Adaptive joint detection and decoding in flat-fading channels via mixture kalman filtering. *IEEE transactions on Information Theory*, 46(6):2079–2094, 2000.

[23] William Fong, Simon J Godsill, Arnaud Doucet, and Mike West. Monte carlo smoothing with application to audio signal enhancement. *IEEE transactions on signal processing*, 50(2):438–449, 2002.

[24] Simon J Godsill, Arnaud Doucet, and Mike West. Monte carlo smoothing for nonlinear time series. *Journal of the american statistical association*, 99(465):156–168, 2004.

[25] Olivier Cappé, Simon J Godsill, and Eric Moulines. An overview of existing methods and recent advances in sequential monte carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

[26] JM Bernardo, MJ Bayarri, JO Berger, AP Dawid, D Heckerman, AFM Smith, and M West. Sequential monte carlo for bayesian computation. In *Bayesian statistics 8: proceedings of the eighth Valencia International Meeting, June 2-6, 2006*, volume 8, page 115. Oxford University Press, USA, 2007.

[27] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[28] Pierre Del Moral and Jean Jacod. Interacting particle filtering with discrete observations. In *Sequential Monte Carlo methods in practice*, pages 43–75. Springer, 2001.

[29] Walter R Gilks and Carlo Berzuini. Following a moving target—monte carlo inference for dynamic bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146, 2001.

[30] Carlos M Carvalho, Michael S Johannes, Hedibert F Lopes, Nicholas G Polson, et al. Particle learning and smoothing. *Statistical Science*, 25(1):88–106, 2010.

[31] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

[32] Sorin Tănase-Nicola and Jorge Kurchan. Topological methods for searching barriers and reaction paths. *Physical review letters*, 91(18):188302, 2003.

[33] Frédéric Cérou, Pierre Del Moral, Teddy Furon, and Arnaud Guyader. Rare event simulation for a static distribution. 2009.

[34] Frédéric Cérou and Arnaud Guyader. Adaptive particle techniques and rare event estimation. In *ESAIM: Proceedings*, volume 19, pages 65–72. EDP Sciences, 2007.

[35] Paul Glasserman, Philip Heidelberger, Perwez Shahabuddin, and Tim Zajic. Splitting for rare event simulation: analysis of simple cases. In *Proceedings Winter Simulation Conference*, pages 302–308. IEEE, 1996.

[36] Jerome Morio, Rudy Pastel, and François Le Gland. An overview of importance splitting for rare event simulation. *European Journal of Physics*, 31(5):1295, 2010.

[37] Thomas M Cover and Joy A Thomas. *Elements of information theory.* John Wiley & Sons, 2012.

[38] Pierre Del Moral, Arnaud Doucet, Ajay Jasra, et al. On adaptive resampling strategies for sequential monte carlo methods. *Bernoulli*, 18(1):252–278, 2012.

[39] Jun S Liu and Rong Chen. Blind deconvolution via sequential imputations. *Journal of the american statistical association*, 90(430):567–576, 1995.

[40] James Carpenter, Peter Clifford, and Paul Fearnhead. Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1):2–7, 1999.

[41] Randal Douc and Olivier Cappé. Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 64–69. IEEE, 2005.

[42] Radford M Neal. Annealed importance sampling. *Statistics and computing*, 11(2):125–139, 2001.

[43] Dieter W Heermann. Computer-simulation methods. In *Computer Simulation Methods in Theoretical Physics*, pages 8–12. Springer, 1990.

[44] Eric Vanden-Eijnden. Transition-path theory and path-finding algorithms for the study of rare events. *Annual review of physical chemistry*, 61:391–420, 2010.

[45] Mark Suresh Joshi. *The Concepts and Practice of Mathematical Finance.* Cambridge University Press, Cambridge, 2008.

[46] Bernt Oksendal. *Stochastic differential equations: an introduction with applications.* Springer-Verlag Berlin Heidelberg, Berlin, 2013.

[47] S Nakano, K Suzuki, and G Ueno. A data-driven monte carlo simulation model of typhoon tracks. presented at JSST 2013 International Conference on Simulation Technology, 2013.

[48] Edward N Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1, 1996.

[49] Jeroen Wouters and Freddy Bouchet. Rare event computation in deterministic chaotic systems using genealogical particle analysis. *arXiv:1511.02703*, 2015.

[50] Edward N Lorenz and Kerry A Emanuel. Optimal sites for supplementary weather observations: Simulation with a small model. *Journal of the Atmospheric Sciences*, 55(3):399–414, 1998.

[51] Edward Ott, Brian R Hunt, Istvan Szunyogh, Aleksey V Zimin, Eric J Kostelich, Matteo Corazza, Eugenia Kalnay, DJ Patil, and James A Yorke. A local ensemble kalman filter for atmospheric data assimilation. *Tellus A: Dynamic Meteorology and Oceanography*, 56(5):415–428, 2004.

[52] Tanguy Laffargue, Khanh-Dang Nguyen Thu Lam, Jorge Kurchan, and Julien Tailleur. Large deviations of lyapunov exponents. *Journal of Physics A: Mathematical and Theoretical*, 46(25):254002, 2013.

[53] Rosalind J Allen, Chantal Valeriani, and Pieter Rein ten Wolde. Forward flux sampling for rare event simulations. *Journal of physics: Condensed matter*, 21(46):463102, 2009.

[54] Daniel M Zuckerman. Equilibrium sampling in biomolecular simulation. *Annual review of biophysics*, 40:41, 2011.

[55] Hilbert Johan Kappen and Hans Christian Ruiz. Adaptive importance sampling for control and inference. *Journal of Statistical Physics*, 162(5):1244–1266, 2016.

[56] Jeremy Heng, Adrian N Bishop, George Deligiannidis, and Arnaud Doucet. Controlled sequential monte carlo. *arXiv:1708.08396*, 2017.

[57] JG Wendel. Hitting spheres with brownian motion. *The annals of probability*, pages 164–169, 1980.

[58] Frank Spitzer. Some theorems concerning 2-dimensional brownian motion. In *Random Walks, Brownian Motion, and Interacting Particle Systems*, pages 21–31. Springer, 1991.

[59] James C Fu and Tung-Lung Wu. Boundary crossing probabilities for high-dimensional brownian motion. *Journal of Applied Probability*, 53(2):543–553, 2016.

[60] Hans Föllmer. Time reversal on wiener space. In *Stochastic Processes—Mathematics and Physics*, pages 119–129. Springer, 1986.

[61] Ioannis Karatzas, Walter Schachermayer, and Bertram Tschiderer. Applying itô calculus to otto calculus. *arXiv preprint arXiv:1811.08686*, 2018.

[62] John C. Hull. *Options, futures, and other derivatives.* Pearson Prentice Hall, Upper Saddle River, NJ [u.a.], 6. ed., pearson internat. ed edition, 2006.

[63] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81(3):637–654, 1973.

[64] Paul Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer Science & Business Media, 2013.