NORTHWESTERN UNIVERSITY

Green Simulation
Reusing the Output of Repeated Experiments

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Industrial Engineering and Management Sciences

By

Mingbin Feng

EVANSTON, ILLINOIS

December 2016

ABSTRACT

Green Simulation

Reusing the Output of Repeated Experiments

Mingbin Feng

We introduce and advocate a new paradigm in simulation experiment design and analysis, called "green simulation," for the setting in which experiments are performed repeatedly with the same simulation model but different input parameters. Green simulation means reusing outputs from previous experiments to answer the question currently being asked of the simulation model.

In this dissertation three classes of green simulation estimators are proposed: the likelihood-ratio-based estimators, the metamodeling-based estimators, and the green Database Monte Carlo estimators. These estimators reuse old simulation outputs in different ways and thus have different requirements, features, and merits. We identify conditions under which these methods are most effective, establish convergence properties for some of the methods, and conduct numerical experiments on practical applications such as catastrophe bond pricing and credit risk evaluation. We find that green simulation can greatly improve computational efficiency in different applications.

Acknowledgements

Though only my name appears on the cover of this dissertation, I owe my gratitude to all the people who have made this dissertation possible. PhD is a life-changing experience: it is not easy, but I had a great time!

I would like to first express my sincere gratitude to my advisor Professor Jeremy Staum for his invaluable guidance for my PhD, my career, and my life in general. Thank you for your inspiration, inquisitiveness, enthusiasm, and eagerness for perfection. Sometimes you are strict and demanding, so impressing you is never easy. Only by exceeding your expectations I can tell that I am ready to move on to the next stage in my life. Thank you for your patience, encouragement (sometimes discouragement too), and the visionary planning for my development. My PhD advanced fast initially when you always gave abundant useful advise. I was fairly upset when you later said "I am not going to help you this time, you should be able to do it on your own now". It was not until the end of my PhD that I realized how much you had planned ahead for me in order to make sure I am on the right track and at the right pace. Most importantly, I want to thank you and Kathryn for your caring and understanding friendship, which gives me a sense of belonging in Evanston. Twenty years from now, when the pain of PhD is forgotten, I surely will still remember the themed dinners prepared by Kathryn, the dinner that we cooked together, and the dinners that you invited me to just to talk me through hardships in life.

I want to thank my committee members Professors Andreas Waechter, Barry Nelson, and Ken Seng Tan for their valuable feedback and insights. Thank you Andreas for having me as your first PhD student (well, kind of!). I enjoyed doing research with you, and it is a pity that it did not converge into a thesis. Barry, you are such a deep thinker. I always have mixed feelings about receiving questions from you: they are so challenging that I could easily embarrass myself, at the same time the questions are so nicely crafted that I gained new insights when I revisited them. In addition, I want to thank you especially for your voluntary support for my job hunt, application, and negotiation process. Thank you Ken Seng for my academic training, research advice, and career planning. This grand 5-year journey starting at Waterloo has been a great motivation for me during the dawn of PhD life. In addition, I want to give special thanks to my dear academic brother Imry Rosenbaum. You offered great support to my PhD, both in research and in life.

I want to thank Jo Ann, Vicky, Agnes and other IEMS staff for their administrative support. Jo Ann, thank you for being so kind to me and to everyone! You, like Kathryn and Jeremy, make me feel at home in the department. Vicky, thank you for chasing every penny for me, even when I myself sometimes had forgotten. Agnes, thank you for all the candies and smiles, both of which are so sweet! While research is no doubt the most important task for a PhD student, the IEMS staff have added so much value to my PhD ***life***.

I spent most of my PhD life in the optimization lab, L375. The lab was intellectually stimulating, the weekly group meeting is an endless source of knowledge, discussion, and laughter! Years after my PhD, I may not remember lab mates research interests, but I will remember how much fun we had asking Alexa nonsense questions, the never-ending

gossip, and the great summer barbecues. While gathered by common research interests, my officemates form a very diverse group. Nitish is the childish Indian: If you had spent as much time in research as in playing pranks, you could have graduated before me. Albert is the passionate Greek: It is very nice of you trying so hard to everyone's problem, but you need to know how to let go when it comes to the daily quizzes. Francisco is the humorous Chilean: In hindsight, I am convinced that INFORMS Philadelphia trip was a conspiracy for you to enjoy the Hard Rock Cafe. Nice assist Isabel! Vijaya is the worrisome (south) Indian: Just relax and chill, research ideas will sparkle when you mind is free from worrying.

Lastly and most importantly, this PhD journey would not have been possible without the support of my family. Mom and Dad, thank you for always believing in me and encouraging me in all of my pursuits. Indeed, *you raise me up to more than I can be!* Jessica, in my master thesis you were acknowledged as my *fiancée*, here in this thesis you are addressed as my *beloved wife*! I am so grateful that our long distant relationship finally comes to an end. In fact, by the time this thesis completes, we will have moved to the next stage in our lives: *parenthood*! To our dear son Albert Feng, thank you for joining us in this life journey!

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

Consider a setting in which simulation experiments are performed repeatedly, using the same simulation model with different values of its inputs. As we discuss in detail below, such settings occur when a simulation model is used routinely to support a business process, and over the lifecycle of a simulation model as it goes from development to application in repeated simulation studies. In these settings, the standard practice is that each new simulation experiment is designed to answer a particular question without using the output of previous simulation experiments. We advocate a paradigm of *green simulation* for repeated experiments, meaning that one should reuse output from previous experiments to answer new questions. The benefit of green simulation is greater computational efficiency. In this article, we show that when old simulation output is reused well, it provides greater accuracy when combined with a new simulation experiment as opposed to what would be achieved by the same new simulation experiment alone.

Green simulation entails a new perspective on management of simulation experiments. The standard practice is to discard or ignore the output of a simulation experiment after it has delivered the desired answer. When a new question arises, a new simulation experiment is designed to answer it without using the output of previous simulation experiments. From this perspective, running the simulation model is a computational cost or expense. From the green simulation perspective, running the simulation model is a computational

investment that provides future benefits, because simulation output is a valuable resource to be used in answering questions that will be asked of the simulation model in the future.

The main contribution of this dissertation is to introduce and advocate the paradigm of green simulation for repeated experiments and to demonstrate theoretically and experimentally that it yields significant benefits in computational efficiency. Specifically, we investigate the case in which a simulation experiment is run routinely (e.g., to support a business process) with updated values of inputs to the simulation model. For this case, we propose, analyze, and test three classes of green simulation estimators based on the likelihood ratio method, the metamodeling method, and the database Monte Carlo method. While these estimators follow the same green simulation paradigm, i.e., reuse simulation output efficiently, the simulation output is reused differently so the resulting estimators have different applicability, features, and merits.

This dissertation is organized as follows: The remainder of this chapter provides a brief literature review that differentiates our work from relevant research and presents a setting of repeated experiments that we study in this dissertation. The next three chapters propose, analyze, and test three classes of green simulation estimators based on the likelihood ratio method (Chapter 2), the metamodeling method (Chapter 3), and the database Monte Carlo method (Chapter 4). The features and merits of different green simulation estimators are studied in the respective chapters. Moreover, we compare and contrast the theoretical properties and practical performance of the green simulation estimators in different applications.

1.1. Literature Review

The core idea of green simulation, reusing simulation output, has been applied to isolated experiments that contain multiple runs. Many types of simulation experiments use multiple runs to learn about the model's *response surface*, the function that maps the model's inputs to a performance measure. In stochastic simulation, this performance measure is often the expected output of the simulation model. There are metamodeling and sensitivity analysis experiments that run the model at different input values to learn about how the response surface varies, globally or locally. In nested simulation experiments, an outer-level simulation generates random values of the inputs at which it is desired to learn the value of the response surface of an inner-level simulation. For example, in assessing the impact of uncertainty about a simulation model's input on the conclusions of a simulation study, this model is the inner-level simulation model, and the outer-level simulation samples values of the inputs from an appropriate distribution. In optimization via simulation experiments, the model is run at different input values in a search for optimal input values. If simulation output from runs at some values of the inputs can be reused in estimating the value of the response surface at another value of the inputs, then experiment designs that involve multiple runs can be modified to be cheaper. It is unnecessary to run many replications at every value of the inputs for which it is desired to estimate the value of the response surface if estimates of these values can reuse output from runs at other values of the inputs. For example, the nested simulation methods of Barton et al. (2014) and Xie et al. (2014) use stochastic kriging (Ankenman et al., 2010) to reuse output from a moderate number of runs to estimate the value of the response surface for many values of the inputs, thus reducing the required number of runs in the

simulation experiment. The likelihood ratio method, also known as the score function method (Rubinstein and Shapiro, 1993; Kleijnen and Rubinstein, 1996), has also been applied to reuse output within isolated experiments for metamodeling, sensitivity analysis, and optimization. Database Monte Carlo (Borogovac and Vakili, 2008) has been applied to reuse simulation outputs from idle computations (Rosenbaum and Staum, 2015) to deliver fast and accurate answers for isolated experiments. We adopt these methods for reusing simulation outputs across experiments.

We acknowledge that green simulation is not the first paradigm to reuse data from old experiments to yield a better estimator in the current experiment. Another such paradigm is the empirical Bayes framework. For example, shrinkage estimators in an empirical Bayes framework (Efron and Morris, 1973, 1975) can improve upon the sample average of data gathered in the current experiment as an estimator of a mean Θ by "shrinking" it towards the mean of an empirical prior distribution for Θ. This mean of the empirical prior distribution could be, for example, the sample average of all data gathered in all experiments so far. This empirical Bayes approach is inefficient in our setting of repeated simulation experiments because it does not take into account that we know the inputs to the simulation model in every experiment. Our green simulation estimators use this knowledge and achieve much lower mean squared error than a shrinkage estimator in a simulation experiment we performed. The results of the experiment are reported in Appendix B, and show that the shrinkage estimator is inappropriate in our setting: it reuses simulation output across experiments but has little improvement on computational efficiency.

1.2. Repeated Experiments

One setting of repeated simulation experiments occurs when simulation supports a business process. For example, in finance and insurance, simulation models support pricing and risk management decisions that are made periodically. At each period, current information, such as prices and forecasts, is used to update the inputs to the model, and a simulation experiment is performed to answer a question about price or risk. Similarly, in manufacturing, service, and logistics systems, simulation models can be used routinely to provide information about expected completion times and to support decisions about such matters as dispatching and staffing. A simulation experiment is run whenever information is required, using inputs that describe the current state of the system.

Another setting of repeated simulation experiments occurs in the lifecycle of a simulation model as it goes from development to application in simulation studies. First, experiments are performed for purposes of verification and validation of the model. They may also be performed for model calibration: to choose realistic values of unknown inputs. For these purposes, the simulation model is run many times with different values of its inputs, to see how its outputs change with its input, and when this behavior is reasonable and realistic. Once model development is complete, experiments are performed for purposes such as making predictions for particular values of the inputs, metamodeling, sensitivity analysis, and optimization. Moreover, some simulation models are used in many simulation studies. Consequently, each model is used in several or many experiments.

Figure 1.1 illustrates a sequence of two repeated experiments, each with a single run. To clarify our terminology, by "a run," we mean one or more replications of simulation output generated with the inputs to the model held fixed. By a simulation "experiment,"

we mean a collection of one or more runs of a simulation model, designed for the purpose of answering a specific question. In Figure 1.1, the first experiment has a single run with input x_1, and the second experiment has a single run with input x_2. Each run has r replications. The purpose of the nth experiment is to estimate $\mu(x_n)$, the mean output of the simulation model when the input is x_n. The standard practice is to estimate $\mu(x_2)$ in the second experiment using only the single run with input x_2. The green question mark in Figure 1.1 indicates the question answered in this article: "How can we reuse the simulation output from the first experiment to improve our answer in the second experiment?"

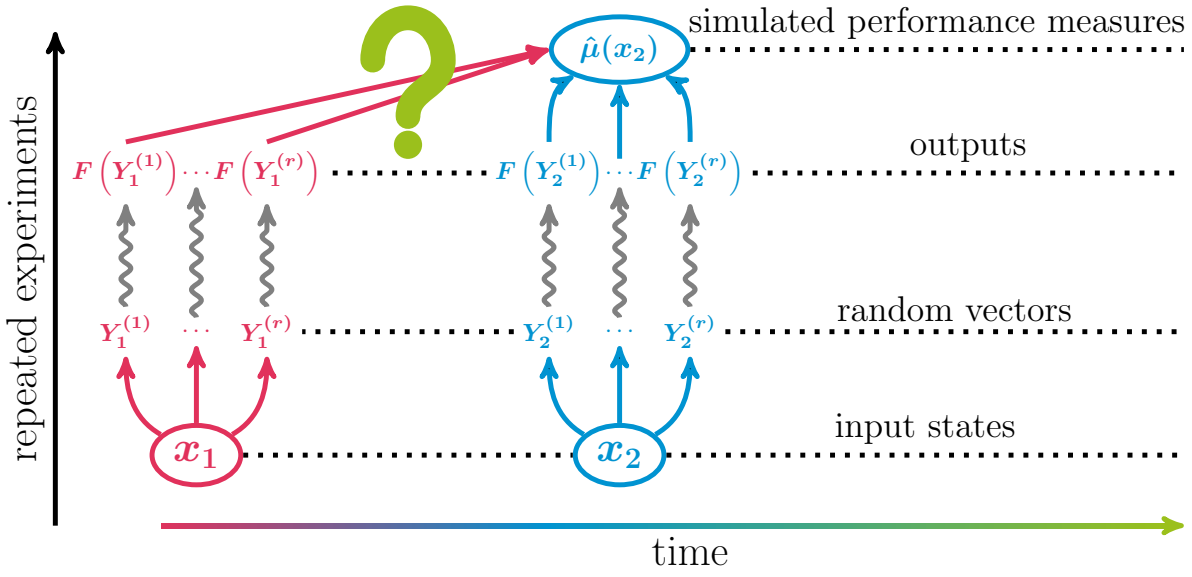


Figure 1.1. Setting of repeated simulation experiments.

CHAPTER 2

Green Simulation via the Likelihood Ratio Method

2.1. Introduction

In finance, insurance, manufacturing, service, logistic systems, and many other practical applications, simulation experiments are repeated with changes in parameters of distributions of random variables generated in the simulation model. In this case, the likelihood ratio (LR) method (Rubinstein and Shapiro, 1993; Kleijnen and Rubinstein, 1996) can be applied to use previous outputs in the current experiment. In particular, one can weigh the previous outputs based on likelihood ratios to obtain unbiased estimators in the current experiment, whose input parameters may be different from those from the previous experiments.

In this chapter we consider three green simulation estimators based on the LR method, or the LR estimators for short. In particular, we prove a novel theorem about how these LR estimators converge as the number of repeated experiments increases, while the number of simulation replications per experiment remains constant. In this setting, the estimator based on standard Monte Carlo does not converge at all, but our LR estimators converge at the canonical rate for Monte Carlo: variance inversely proportional to total computational budget. They converge at this rate even though the total computational budget includes outputs from previous experiments, which are not obviously relevant to answering the question currently being asked of the simulation model. In addition, we

import methods from multiple importance sampling (where importance sampling is used for variance reduction) into the LR method (which reuses simulation output), and provide evidence that there can be great practical value to using one of the LR estimators that we propose.

In applications and surveys of the likelihood ratio method that we have seen, such as Beckman and McKay (1987), Rubinstein and Shapiro (1993), Kleijnen and Rubinstein (1996), and Glasserman and Xu (2014), there is an isolated experiment within which each estimator reuses the output from a single run. The exception is Maggiar et al. (2015): they construct an estimator by reusing output from multiple runs in an isolated experiment. We reuse output from multiple runs that come from multiple experiments. Even in the simple setting of a single run per experiment, which we analyze in this chapter, reusing outputs from multiple experiments entails reusing outputs from multiple runs. Estimation using multiple simulation runs, weighted based on likelihood ratios, was given the name *multiple importance sampling* by Veach (1997). On multiple importance sampling, see, for example, Hesterberg (1988, 1995), Owen and Zhou (2000), Veach and Guibas (1995), and Veach (1997). Drawing on this literature on multiple importance sampling, Maggiar et al. (2015) use two of the LR estimators we consider in this chapter, the individual likelihood ratio (ILR) estimator (Section 2.2.1) and the mixture likelihood ratio (MLR) estimator (Section 2.2.3); we also treat a new estimator, the weighted likelihood ratio (WLR) estimator (Section 2.2.2).

The work of Maggiar et al. (2015) and the current chapter differ in setting and findings. The key differences are that their work focuses on an isolated experiment with a deterministic simulation model, whereas ours focuses on repeated experiments with

a stochastic simulation model. Their goal is optimization of the response surface after smoothing by convolution with a Gaussian kernel, to reduce the influence of numerical noise in the deterministic simulation. They apply the likelihood ratio method to the corresponding Gaussian random variable. Their convergence theorem describes convergence to an optimal solution within an isolated optimization experiment with an increasing number of iterations. Our green simulation paradigm applies broadly to stochastic simulation, and we emphasize the setting of repeated experiments. Our convergence theorems describe convergence of estimators of values of the response surface at some or all points as the number of repeated experiments increases. Another difference between the work of Maggiar et al. (2015) and our work is in the findings about the ILR and MLR estimators. They report that the choice between ILR and MLR estimators makes "only a small difference on the performance" of their optimization procedure when applied to a testbed of optimization problems. In Section 2.5, we find that the difference in practical performance between ILR and MLR estimators could be large, depending on the simulation model of the repeated experiments in consideration. We recommend the MLR estimator, which is theoretically superior, because we find that it can work well in practice even in cases where the ILR estimator yields poor results.

2.2. Green Simulation via Likelihood Ratio Method

We develop green simulation estimators via the LR method in the setting of repeated experiments with the same simulation model and some changing parameters that affect the likelihood of simulated random variables. Let X_n represent the parameters in the nth experiment, for example, prices observed in the market on day n or forecasted arrival rates

for period n. We treat $\{X_n : n = 1, 2, \ldots\}$ as a discrete-time stochastic process taking values in a Polish space $\mathcal{X}$. We use "state" to refer to an element of $\mathcal{X}$ or a random variable taking values in $\mathcal{X}$, and "current" to refer to quantities associated with the nth experiment. Thus, X_n is the current state. We suppose that the current state is observable at the current time step n, when the current experiment needs to be run, but was not observable earlier.

Given the current state X_n, the current simulation experiment samples a random vector Y_n according to the conditional likelihood $h(\cdot; X_n)$. For example, $h(y; x)$ could be the conditional probability density for a stock's price to be y in one year given that the stock's current price is x. As another example, $h(y; x)$ could be the conditional probability density for the vector y of interarrival times and service times given the arrival rate x, regarding service rate as fixed and not included in x. We will write expectations in the form of integrals, which implicitly assumes that Y_n has a conditional probability density $h(\cdot; X_n)$, but this is not essential; the setting allows for continuous, discrete, and mixed conditional distributions.

The simulation output or simulated performance of the stochastic system is $F(Y_n)$, where the function $F : \mathcal{Y} \mapsto \mathbb{R}$ represents the logic of the simulation model. For example, $F(y)$ could be the discounted payoff of a stock option if the stock's price in one year is y, or $F(y)$ could be the average customer waiting time in a queue given the vector y of interarrival times and service times.

In the current experiment, we wish to estimate the conditional expected performance $\mu(X_n)$ of the stochastic system given the current state X_n; $\mu(X_n)$ can also be described as

the current expected performance. The expected performance for state x is

$$\mu(x) = \mathbb{E}\left[F\left(Y_n\right)|X_n = x\right] = \int_{\mathcal{Y}} F\left(y\right) h\left(y;x\right) dy, \tag{2.1}$$

which is the same for all n. The purpose of the current experiment is to estimate the current expected performance $\mu(X_n)$. This is a random variable because the current state X_n was not observable at time step 0. Figures of merit for an estimator, such as bias and variance, should be evaluated conditional on the current state X_n.

Standard practice in the setting of repeated experiments is to estimate $\mu(X_n)$ by the *Standard Monte Carlo (SMC)* estimator

$$\widehat{\mu}_r^{SMC}(X_n) = \frac{1}{r}\sum_{j=1}^{r} F\left(Y_n^{(j)}\right), \tag{2.2}$$

based on running r replications of the simulation model with the parameters set according to the current state X_n. For simplicity in notation, we have assumed that the number r of replications is fixed, but this is not essential. We refer to $\left\{F\left(Y_n^{(j)}\right) : j = 1,\ldots,r\right\}$ as the output of the current experiment. Clearly, $\widehat{\mu}_r^{SMC}(X_n)$ is conditionally unbiased for $\mu(X_n)$, given X_n. The conditional variance for state x is

$$\sigma^2(x) = \mathbb{V}\mathrm{ar}\left[F\left(Y_n\right)|X_n = x\right] = \int_{\mathcal{Y}} \left(F\left(y\right) - \mu(x)\right)^2 h\left(y;x\right) dy, \tag{2.3}$$

which is the same for all n. The conditional variance of the SMC estimator, given the current state X_n, is $\sigma^2(X_n)/r$. Thus, to reduce the conditional variance of the SMC estimator, one must increase the number r of replications in the current experiment.

We propose three LR estimators that reuse the output of the $n-1$ previous experiments and combine it with the current (nth) experiment's output. In the current experiment, the *target distribution* appears in the conditional expectation $\mu(X_n) = \mathbb{E}\left[F\left(Y_n\right)|X_n\right]$ that we are estimating. The target distribution has the likelihood $h\left(\cdot; X_n\right)$. In a previous experiment at time step $k < n$, the *sampling distribution* had likelihood $h\left(\cdot; X_k\right)$. To use the output of a previous experiment in estimating $\mu(X_n)$ in a way that is conditionally unbiased given the state history $X_1, \ldots, X_n$, we can adjust the old output by using the likelihood ratio between the target distribution and the sampling distribution:

$$\mathbb{E}\left[\frac{h\left(Y_k; X_n\right)}{h\left(Y_k; X_k\right)} F\left(Y_k\right) \middle| X_1, \ldots, X_n\right] = \int_{\mathcal{Y}} \frac{h\left(y; X_n\right)}{h\left(y; X_k\right)} h\left(y; X_k\right) F\left(y\right) dy = \mu(X_n). \tag{2.4}$$

In Section 2.4, we show that, under some conditions, the conditional variance of the LR estimators goes to zero as the number n of experiments goes to infinity, even if the number r of replications per experiment is fixed.

We make the following assumptions to support the LR estimators. Although not all of the assumptions in the theorems are transparent, we show in Appendix A that they can be verified in a realistic example.

(A.1) The nth simulation experiment is affected by n and the stochastic process $\{X_n : n = 1, 2, \ldots\}$ only through the conditional likelihood In other words, the input state X_n does not affect the simulation logic F.

(A.2) For any n, $n' \neq n$, j, and j', given the state X_n, $Y_n^{(j)}$ is conditionally independent of the state $X_{n'}$ and the simulated random vector $Y_{n'}^{(j')}$.

(A.3) For any n, j, and $j' \neq j$, given the state X_n, the random vector $Y_n^{(j)}$ simulated in the jth replication of the nth experiment is conditionally independent of $Y_n^{(j')}$.

(A.4) For all $x \in \mathcal{X}$, the likelihoods $h(\cdot; x)$ have the same support $\mathcal{Y}$.

(A.5) For any state $x \in \mathcal{X}$ and any $y \in \mathcal{Y}$, the likelihood $h(y; x)$ can be evaluated.

(A.6) For any states $x, x' \in \mathcal{X}$, the target-x-sample-x' variance defined as

$$\sigma_x^2(x') = \int_{\mathcal{Y}} \left(F(y) \frac{h(y; x)}{h(y; x')} - \mu(x) \right)^2 h(y; x')\, dy \tag{2.5}$$

and the expected performance $\mu(x)$ defined in Equation (2.1) are finite.

In Assumption (A.1), it is essential for the LR method that the parameters that change affect only likelihoods, not the simulation logic F. Assumptions (A.1) and (A.2) imply that the current simulation experiment is not affected directly by its time index n or by any past state $X_{n'}$ for $n' < n$ or by any of the output of a past experiment. We use these assumptions on the sequence of repeated experiments to support our analysis of the LR estimators. Assumption (A.3), which asserts that each simulation experiment has independent replications, is made for the sake of simplicity in the analysis. It is not essential to the LR method or its analysis: for example, it would be acceptable to use stratified sampling within each experiment, and this would introduce the usual complications in analyzing and estimating variance. Assumptions (A.4) and (A.5) ensure that the likelihood ratio $h(y; x)/h(y; x')$ is finite and can be computed; it is not enough merely to be able to sample according to the likelihood $h(\cdot; x)$. Assumption (A.6) is needed to give the LR estimators finite conditional variance.

The need to satisfy these assumptions limits the applicability of the LR method. Furthermore, the LR estimators could work poorly if the variances associated with Equation (2.5) are too large. This can happen, for example, if the input x affects the likelihood of many independent terms. These are known limitations of the LR method

in general, not specific to green simulation. Despite them, interesting examples may fit well into the LR framework. One such example, in Section 2.5.1, involves simulating a random number of independent random variables and summing them. It might appear to pose a difficulty for the LR method if Y_n is a random vector including the number of terms in the sum and every term in the sum. We avoid this difficulty by defining Y_n to be the sum and working with its likelihood, not with the joint likelihood of the number of terms in the sum and every term in the sum. By using such techniques if necessary, the LR method can be made to apply even to non-trivial examples. It is not always possible to make the LR method work well; in such situations, one can use other green simulation methods, such as those proposed in Chapter 3 and Chapter 4.

2.2.1. Individual Likelihood Ratio Estimator

Based on the fundamental idea of the LR method as expressed in Equation (2.4), for any state $x \in \mathcal{X}$ we get the following unbiased estimator of $\mu(x)$ using outputs from the kth experiment:

$$\widehat{\mu}_{k,r}^{LR}(x) = \frac{1}{r}\sum_{j=1}^{r} \frac{h\left(Y_k^j; x\right)}{h\left(Y_k^j; X_k\right)} F\left(Y_k^{(j)}\right). \tag{2.6}$$

Averaging these estimators from all previous experiments and the current (nth) experiment yields the *Individual Likelihood Ratio (ILR)* estimator of $\mu(x)$:

$$\widehat{\mu}_{n,r}^{ILR}(x) = \frac{1}{n}\sum_{k=1}^{n}\widehat{\mu}_{k,r}^{LR}(x) = \frac{1}{n}\sum_{k=1}^{n}\left(\frac{1}{r}\sum_{j=1}^{r}\frac{h\left(Y_k^{(j)};x\right)}{h\left(Y_k^{(j)};X_k\right)}F\left(Y_k^{(j)}\right)\right) \tag{2.7a}$$

$$= \sum_{j=1}^{r}\sum_{k=1}^{n}\frac{1}{nr}\frac{h\left(Y_k^{(j)};x\right)}{h\left(Y_k^{(j)};X_k\right)}F\left(Y_k^{(j)}\right). \tag{2.7b}$$

The ILR estimator is so named because it contains likelihood ratios that each involve one individual sampling distribution; this distinguishes it from the mixture likelihood ratio estimator proposed in Section 2.2.3. In particular, $\widehat{\mu}_{n,r}^{ILR}(X_n)$ is our estimator of the current expected performance $\mu(X_n)$. However, the LR method enables us to estimate expected performance given a state x that we never used in a sampling distribution.

The ILR estimator in (2.7a) can be seen as the average of n individual likelihood ratio estimators $\widehat{\mu}_{k,r}^{LR}(x)$ for $k = 1, \ldots, n$. If the target state is the current state, i.e., $x = X_n$, then $\widehat{\mu}_{n,r}^{LR}(x)$ is the SMC estimator. It is worth emphasizing the difference between the present application of likelihood ratios and the typical application of importance sampling. In importance sampling, the designer of the simulation experiment chooses the sampling distribution with the aim of reducing variance. In our LR estimators, we do not address the choice of sampling distribution. We assume that the sampling distribution in the current experiment is determined by the current state X_n, as is standard practice in the setting of repeated experiments. We use likelihood ratios not to reduce variance, but to enable reuse of simulation output based on different sampling distributions.

Of course, the effect of likelihood ratios on variance needs to be considered. The target-x-sample-x' variance defined in Equation (2.5) could be more or less than the target-x-sample-x variance $\sigma_x^2(x) = \sigma^2(x)$ associated with standard Monte Carlo; in general it could even be infinite. The target-x-sample-X_k variance $\sigma_x^2(X_k)$ can be estimated by

$$\widehat{\sigma}_x^2(X_k) = \frac{1}{r}\sum_{j=1}^{r}\left(\frac{h\left(Y_k^{(j)};x\right)}{h\left(Y_k^{(j)};X_k\right)}F\left(Y_k^{(j)}\right) - \widehat{\mu}_{r,X_k}^{LR}(x)\right)^2.$$

Based on (2.7a)

$$\mathbb{V}\mathrm{ar}\left[\widehat{\mu}_{n,r}^{ILR}(x)\middle|\, X_1,\ldots,X_n\right] = \frac{1}{n^2 r}\sum_{k=1}^{n}\sigma_x^2(X_k) \tag{2.8}$$

which can be estimated by $\sum_{k=1}^{n}\widehat{\sigma}_x^2(X_k)/(n^2 r)$. If none of $\sigma_{X_n}^2(X_1),\ldots,\sigma_{X_n}^2(X_n)$ is too large, then the ILR estimator of current expected performance has lower conditional variance $\sum_{k=1}^{n}\sigma_{X_n}^2(X_k)/(n^2 r)$ than $\sigma^2(X_n)/r$, which is the conditional variance of the SMC estimator. However, the ILR estimator could have higher or infinite conditional variance.

Considering that $\sigma_{X_n}^2(X_1),\ldots,\sigma_{X_n}^2(X_n)$ may be unequal, one might try to construct a lower-variance estimator by using unequal weights instead of the equal weights $1/n$ in (2.7a). This leads to the weighted likelihood ratio estimator proposed in Section 2.2.2. Even better results are possible if we replace the equal weights $1/n$ in (2.7b) with unequal weights that depend on the random vector $Y_k^{(j)}$. This topic is addressed in the research literature on multiple importance sampling, which leads to the mixture likelihood ratio estimator proposed in Section 2.2.3.

2.2.2. Weighted Likelihood Ratio Estimator

Instead of the ILR estimator (2.7a) with its equal weights $1/n$, consider a weighted average $\sum_{k=1}^{n} w_k \widehat{\mu}_{k,r}^{LR}(x)$ where the weights $w_1, \ldots, w_n$ sum to one but may be unequal. Its conditional variance given the state history $X_1, \ldots, X_n$ is $\sum_{k=1}^{n} w_k^2 \sigma_{X_n}^2 (X_k) / r$. Choosing the weights

$$w_k^{WLR} = \frac{\sigma_x^{-2}(X_k)}{\sum_{k=1}^{n} \sigma_x^{-2}(X_k)}, \quad \forall k = 1, \ldots, n \tag{2.9}$$

that minimize the conditional variance yields the *Weighted Likelihood Ratio (WLR)* estimator of $\mu(x)$:

$$\widehat{\mu}_{n,r}^{WLR}(x) = \sum_{k=1}^{n} w_k^{WLR} \widehat{\mu}_{k,r}^{LR}(x) = \sum_{k=1}^{n} w_k^{WLR} \left[\frac{1}{r} \sum_{j=1}^{r} F\left(Y_k^{(j)}\right) \frac{h\left(Y_k^{(j)}; x\right)}{h\left(Y_k^{(j)}; X_k\right)} \right] \tag{2.10}$$

Proposition 2.2.1. *Suppose Assumptions (A.1)–(A.6) hold, for any $n, r \in \mathbb{N}_+$ and $x, X_1, \ldots, X_n \in \mathcal{X}$, the weights given in Equation (2.9) minimize the conditional variance $\mathbb{V}ar\left[\sum_{k=1}^{n} w_k \widehat{\mu}_{k,r}^{LR}(x) \middle| X_1, \ldots, X_n\right]$ subject to the constraint $\sum_{k=1}^{n} w_k = 1$. The resulting minimum conditional variance is*

$$\mathbb{V}ar\left[\widehat{\mu}_{n,r}^{WLR}(x) \middle| X_1, \ldots, X_n\right] = \frac{1}{r \sum_{k=1}^{n} \sigma_x^{-2}(X_k)}. \tag{2.11}$$

Proof. The Lagrangian function for this minimization problem is

$$L(w_1, \ldots, w_n, \lambda) = \sum_{k=1}^{n} w_k^2 \sigma_{X_n}^2 (X_k) + \lambda \left(1 - \sum_{k=1}^{n} w_k\right).$$

From the first-order optimality condition

$$\frac{\partial L(w_1, \ldots, w_n, \lambda)}{\partial w_k^{WLR}} = 2w_k^{WLR}\sigma^2_{X_n}(X_k) - \lambda = 0$$

we get $w_k^{WLR} = \lambda/2\sigma^2_{X_n}(X_k)$ for all $k = 1, \ldots, n$. Therefore w_k^{WLR} is proportional to $\sigma_x^{-2}(X_k)$, and the denominator in Equation 2.9 is what is required to satisfy the constraint $\sum_{k=1}^n w_k^{WLR} = 1$. Substituting w_k^{WLR} into the conditional variance formula $\sum_{k=1}^n w_k^2\sigma^2_{X_n}(X_k)/r$ yields $1/(r\sum_{k=1}^n \sigma_x^{-2}(X_k))$ □

Proposition 2.2.2 and its proof show how much better the WLR estimator is than (2.12) the ILR estimator and (2.14) the SMC estimator in terms of conditional variance. It also shows (2.13) that the WLR estimator has a defensive property that the ILR estimator lacks: the conditional variance of the WLR estimator is bounded above by that of the best LR estimator generated from any single sampling distribution. The WLR estimator can only be improved by including samples from another sampling distribution, even if the associated variance is large; if so, the WLR estimator compensates by giving this sampling distribution a small weight. Proposition 2.2.2 relies on the following lemma, which holds due to convexity of the function that maps a to $1/a$, or equivalently, due to the well-known inequality between arithmetic and harmonic mean for non-negative numbers.

Lemma 2.2.1. *For any n and $a_1, \ldots, a_n > 0$, $\sum_{k=1}^n a_k/n \geq n/\sum_{k=1}^n a_k^{-1}$, with strict inequality if and only if there exist k, k' such that $a_k \neq a_{k'}$.*

Proposition 2.2.2. *If Assumptions (A.1)–(A.6) hold, then for any* $n, r \in \mathbb{N}_+$ *and* $x, X_1, \ldots, X_n \in \mathcal{X}$,

$$\frac{\mathbb{V}ar\left[\widehat{\mu}_{n,r}^{ILR}(x)\middle| X_1, \ldots, X_n\right]}{\mathbb{V}ar\left[\widehat{\mu}_{n,r}^{WLR}(x)\middle| X_1, \ldots, X_n\right]} = \frac{1}{n^2}\left(\sum_{k=1}^{n}\sigma_x^2\left(X_k\right)\right)\left(\sum_{k=1}^{n}\sigma_x^{-2}(X_k)\right) \geq 1 \tag{2.12}$$

with strict inequality if $\sigma_x^2\left(X_k\right) \neq \sigma_x^2\left(X_{k'}\right)$ *for some* $k, k' = 1, \ldots, n$*; also*

$$\frac{1}{nr}\min_{k\in\{1,\ldots,n\}}\sigma_x^2\left(X_k\right) \leq \mathbb{V}ar\left[\widehat{\mu}_{n,r}^{WLR}(x)\middle| X_1, \ldots, X_n\right] \leq \frac{1}{r}\min_{k\in\{1,\ldots,n\}}\sigma_x^2\left(X_k\right), \tag{2.13}$$

and for $x = X_n$,

$$\frac{\mathbb{V}ar\left[\widehat{\mu}_{r}^{SMC}(X_n)\middle| X_1, \ldots, X_n\right]}{\mathbb{V}ar\left[\widehat{\mu}_{n,r}^{WLR}(X_n)\middle| X_1, \ldots, X_n\right]} = \sigma_{X_n}^2(X_n)\sum_{k=1}^{n}\sigma_{X_n}^{-2}(X_k) \geq 1 \tag{2.14}$$

with strict inequality if $\sigma_{X_n}^2(X_k) < \infty$ *for some* $k < n$.

Proof. By Lemma 2.2.1, $\sum_{k=1}^{n}\sigma_x^{-2}(X_k)/n \geq 1/(\sum_{k=1}^{n}\sigma_x^2(X_k)/n)$. Using this with Equations (2.8) and (2.11), the inequality in Equation (2.12) follows.

Equation (2.13) follows from Equation (2.11) and taking the reciprocal of

$$\max_{k\in\{1,\ldots,n\}}\sigma_x^{-2}(X_k) \leq \sum_{k=1}^{n}\sigma_x^{-2}(X_k) \leq n\max_{k\in\{1,\ldots,n\}}\sigma_x^{-2}(X_k).$$

Equation (2.14) follows from Equation (2.11) and $\mathbb{V}\mathrm{ar}\left[\widehat{\mu}_{r}^{SMC}(X_n)\middle| X_1, \ldots, X_n\right] = \sigma_{X_n}^2(X_n)/r$. □

Proposition 2.2.2 showcases the theoretical advantages of the WLR estimator using the weights in Equation (2.9), which involve target-x-sample-X_k variances $\sigma_x^2\left(X_k\right)$. Unfortunately, these variances are usually unknown. Therefore, in a practical implementation,

one would replace these variances by their estimates $\widehat{\sigma}_x^2(X_k)$. The resulting weights would be suboptimal, and Proposition 2.2.2 would not apply to this *empirical* WLR (EWLR) estimator. As our numerical experiments show, the EWLR estimator can be worse than the ILR and SMC estimators. In the next section we propose an estimator that enjoys good theoretical properties, like the WLR estimator, and can be implemented without losing its theoretical properties by estimation of weights.

2.2.3. Mixture Likelihood Ratio Estimator

In our setting of repeated experiments, we have r replications sampled from each of n distributions. The collection of nr observations can be viewed as a stratified sample from an equally-weighted mixture of these n distributions. The likelihood of the mixture is denoted by $\bar{h}(\cdot; X_1, \ldots, X_n)$, where

$$\bar{h}(y; x_1, \ldots, x_n) = \frac{1}{n}\sum_{k=1}^{n} h(y; x_k). \tag{2.15}$$

Hesterberg (1988) and Veach and Guibas (1995) advocated replacing the equal weights $1/nr$ in (2.7b) with "balance heuristic" weights which, in our setting, are

$$w^{MLR}(Y_k^{(j)}) = \frac{h\left(Y_k^{(j)}; X_k\right)}{nr\bar{h}\left(Y_k^{(j)}; X_1, \ldots, X_n\right)}.$$

This leads to the following *mixture likelihood ratio (MLR)* estimator for $\mu(x)$:

$$\begin{aligned}\widehat{\mu}_{n,r}^{MLR}(x) &= \sum_{j=1}^{r}\sum_{k=1}^{n} w^{MLR}(Y_k^{(j)})\frac{h\left(Y_k^{(j)};x\right)}{h\left(Y_k^{(j)};X_k\right)}F\left(Y_k^{(j)}\right) \\ &= \sum_{k=1}^{n}\sum_{j=1}^{r}\frac{1}{nr}\frac{h\left(Y_k^{(j)};x\right)}{\bar{h}\left(Y_k^{(j)};X_1,\ldots,X_n\right)}F\left(Y_k^{(j)}\right).\end{aligned} \tag{2.16}$$

The MLR estimator is the LR estimator (2.6) that arises when we consider the pooled outputs of all simulation experiments performed so far, $\left\{Y_k^{(j)} : k = 1,\ldots,n, j = 1,\ldots,r\right\}$ as stratified sampling from the mixture distribution (2.15) that has likelihood $\bar{h}$ with r independent samples allocated to each of n strata (Hesterberg, 1995). It follows from this interpretation, or immediately from the results of Veach and Guibas (1995, Section 3.2), that the MLR estimator is conditionally unbiased for $\mu(x)$ given $X_1,\ldots,X_n$.

Proposition 2.2.3. *Suppose Assumptions (A.1)–(A.6) hold, for any* $n, r \in \mathbb{N}_+$ *and* $x, X_1,\ldots,X_n \in \mathcal{X}$, $\mathbb{E}\left[\widehat{\mu}_{n,r}^{MLR}(x)\middle| X_1,\ldots,X_n\right] = \mu(x)$.

The conditional variance of the MLR estimator is better than that of the ILR estimator (Proposition 2.2.4). The MLR estimator has a defensive property similar to that of the WLR estimator: an upper bound on its conditional variance related to the conditional variance of the best LR estimator from any single sampling distribution (Proposition 2.2.5). The conditional variance of the MLR estimator can be estimated by

$$\widehat{\mathbb{V}\mathrm{ar}}\left[\widehat{\mu}_{n,r}^{MLR}(x)\middle| X_1,\ldots,X_n\right] = \frac{1}{r}\sum_{j=1}^{r}\left(\sum_{k=1}^{n}\frac{h\left(Y_k^{(j)};x\right)F\left(Y_k^{(j)}\right)}{n\bar{h}\left(Y_k^{(j)};X_1,\ldots,X_n\right)} - \widehat{\mu}_{n,r}^{MLR}(x)\right)^2.$$

The following proposition and proof are taken from Theorem A.2 of Martino et al. (2014), expanded to provide sufficient conditions for a strict inequality. They involve the conditional expectation of the kth term in Equation (2.16),

$$\begin{aligned} m_k(x; X_1, \ldots, X_n) &= \mathbb{E}\left[\frac{F(Y_k)\, h(Y_k; x)}{\bar{h}(Y_k; X_1, \ldots, X_n)} \middle| X_1, \ldots, X_n \right] \\ &= \int_{\mathcal{Y}} \frac{F(y)\, h(y; x)}{\bar{h}(y; X_1, \ldots, X_n)} dy \end{aligned} \tag{2.17}$$

Proposition 2.2.4. *Suppose Assumptions (A.1)–(A.6) hold, for any $n, r \in \mathbb{N}_+$ and $x, X_1, \ldots, X_n \in \mathcal{X}$,*

$$\mathbb{V}ar\left[\widehat{\mu}_{n,r}^{MLR}(x) \middle| X_1, \ldots, X_n \right] \leq \mathbb{V}ar\left[\widehat{\mu}_{n,r}^{ILR}(x) \middle| X_1, \ldots, X_n \right]. \tag{2.18}$$

This inequality is strict if there exists k, k' such that $m_k(x; X_1, \ldots, X_n) \neq m_k(x; X_1, \ldots, X_n)$.

Proof. Let $\left\{ \widetilde{Y}_k^{(j)} : k = 1, \ldots, n, j = 1, \ldots, r \right\}$ be an i.i.d. sample according to the mixture likelihood $\bar{h}(\cdot; X_1, \ldots, X_n)$ and define

$$\widehat{\mu}_{n,r}^{MIX}(x) = \sum_{k=1}^{n} \sum_{j=1}^{r} \frac{1}{nr} \frac{h\left(\widetilde{Y}_k^{(j)}; x \right)}{\bar{h}\left(\widetilde{Y}_k^{(j)}; X_1, \ldots, X_n \right)} F\left(\widetilde{Y}_k^{(j)} \right).$$

The estimators $\widehat{\mu}_{n,r}^{MIX}(x)$ and $\widehat{\mu}_{n,r}^{MLR}(x)$ are similar: the latter is a stratified-sampling version of the former, with equal number of samples allocated to the n equally weighted strata; the sampling likelihood of the k stratum is $h(\cdot; X_k)$. Therefore

$$\mathbb{E}\left[\widehat{\mu}_{n,r}^{MIX}(x) \middle| X_1, \ldots, X_n \right] = \mathbb{E}\left[\widehat{\mu}_{n,r}^{MLR}(x) \middle| X_1, \ldots, X_n \right] = \mu(x)$$

where the last equality holds by Proposition 2.2.3. We have

$$\begin{aligned}&\mathbb{V}\mathrm{ar}\left[\widehat{\mu}_{n,r}^{ILR}(x)\middle|X_1,\ldots,X_n\right]-\mathbb{V}\mathrm{ar}\left[\widehat{\mu}_{n,r}^{MIX}(x)\middle|X_1,\ldots,X_n\right]\\&=\frac{1}{nr}\int_{\mathcal{Y}}\left(F\left(y\right)h\left(y;x\right)\right)^2\left(\frac{1}{n}\sum_{k=1}^{n}\frac{1}{h\left(y;X_k\right)}-\frac{1}{\bar{h}\left(y;X_1,\ldots,X_n\right)}\right)dy\geq 0\end{aligned}$$

because by Lemma 2.2.1, for all $y\in\mathcal{Y}$

$$\frac{1}{n}\sum_{k=1}^{n}\frac{1}{h\left(y;X_k\right)}\geq\frac{1}{\frac{1}{n}\sum_{k=1}^{n}h\left(y;X_k\right)}=\frac{1}{\bar{h}\left(y;X_1,\ldots,X_n\right)}.$$

Next we observe that $\widehat{\mu}_{n,r}^{MLR}(x)$ is a stratified-sampling version of $\widehat{\mu}_{n,r}^{MIX}(x)$, with equal number of samples allocated to the n equally weighted strata: the sampling likelihood of the kth stratum is $h\left(\cdot;X_k\right)$. Therefore

$$\mathbb{V}\mathrm{ar}\left[\widehat{\mu}_{n,r}^{MLR}(x)\middle|X_1,\ldots,X_n\right]\leq\mathbb{V}\mathrm{ar}\left[\widehat{\mu}_{n,r}^{MIX}(x)\middle|X_1,\ldots,X_n\right]$$

and inequality is strict if there exist strata k,k' with different means $m_k(x;X_1,\ldots,X_n)\neq m_{k'}(x;X_1,\ldots,X_n)$. □

The next proposition follows from Theorem 9.2 of Veach (1997) and Equation (8) of Owen and Zhou (2000).

Proposition 2.2.5. *Suppose Assumptions (A.1)–(A.6) hold, for any $n,r\in\mathbb{N}_+$ and $x,X_1,\ldots,X_n\in\mathcal{X}$, if F is a non-negative function or there exists $k\in\{1,\ldots,n\}$ such that $x=X_k$, then*

$$\mathbb{V}ar\left[\widehat{\mu}_{n,r}^{MLR}(x)\middle|X_1,\ldots,X_n\right]\leq\frac{1}{r}\min_{k\in\{1,\ldots,n\}}\sigma_x^2\left(X_k\right)+\left(\frac{1}{r}-\frac{1}{nr}\right)\left(\mu(x)\right)^2. \tag{2.19}$$

2.3. Green Algorithms for Likelihood-Ratio-Based Estimators

This section proposes and analyzes algorithms for the three LR estimators in the setting of repeated experiments. The algorithms are also green, in the sense that they store and reuse likelihood evaluations as well as simulation output. Suppose that one simulation replication has computational cost C_F, one evaluation of a likelihood has computational cost C_h, and the computational cost of basic arithmetic operations such as addition, multiplication, and division is negligible in comparison to these. We envision a situation in which C_F is large, C_h is smaller but need not be negligible, storage space is abundant, and memory access is fast. We consider a sequence of experiments, indexed $n = 1, 2, \ldots$, of r replications each. For each experiment in the sequence, the SMC, ILR, EWLR or MLR estimator of the current expected performance $\mu(X_n)$ is computed, and the green simulation procedures store some information to be reused in the next experiment. We analyze the storage requirement and computation cost of the nth experiment.

For benchmarking purposes, we first consider the SMC estimator. It has zero storage requirement in the sense that no information is stored from one experiment to the next. Its computation cost is rC_F.

Algorithm 1 applies to the ILR and EWLR estimators. Consider the ILR estimator $\widehat{\mu}_{n,r}^{ILR}(X_n)$ in (2.7). The likelihood $h\left(Y_k^{(j)}; X_k\right)$ in the denominator does not change as n increases. Therefore we store and reuse likelihoods from one experiment to the next in Algorithm 1. The storage requirements and non-negligible computation costs for the nth experiment are shown on the right in Algorithm 1. The algorithm has storage requirement $2nr$ and computation cost $rC_F + nrC_h$ for the nth experiment. The linear growth rate in n is reassuring; it suggests that it is affordable to reuse the outputs of many experiments

in the ILR and EWLR estimators. Compared to the ILR estimator, the EWLR estimator requires more basic arithmetic operations to estimate the EWLR weights.

Algorithm 1 Green implementation of ILR or EWLR estimator in the nth experiment

Observe X_n and initialize $\widehat{\mu}_{n,r}^{ILR}(X_n) \leftarrow 0$ or $\widehat{\mu}_{n,r}^{EWLR}(X_n) \leftarrow 0$
for $j = 1, \ldots, r$ **do**
    Sample $Y_n^{(j)}$ and evaluate $F\left(Y_n^{(j)}\right)$ ▷ rC_F computation
    Append to output storage $F\left(Y_n^{(j)}\right)$ ▷ nr storage
    Calculate likelihood $h\left(Y_n^{(j)}; X_n\right)$ ▷ rC_h computation
    Append to likelihood storage $h\left(Y_k^{(n)}; X_n\right)$ ▷ nr storage
end for
Set $\widehat{\mu}_{r,X_k}^{LR}(X_n) \leftarrow \frac{1}{r}\sum_{j=1}^{r} F\left(Y_n^{(j)}\right)$
for $k = 1, \ldots, n-1$ **do**
    for $j = 1, \ldots, r$ **do**
        Retrieve $F\left(Y_k^{(j)}\right)$ and $h\left(Y_k^{(j)}; X_k\right)$ from storage
        Calculate likelihood $h\left(Y_k^{(j)}; X_n\right)$ ▷ $(n-1)rC_h$ computation
    end for
    Set $\widehat{\mu}_{r,X_k}^{LR}(X_n) \leftarrow \frac{1}{r}\sum_{j=1}^{r} \frac{h\left(Y_k^{(j)};X_n\right)}{h\left(Y_k^{(j)};X_k\right)} F\left(Y_k^{(j)}\right)$
end for
if ILR estimator **then**
    Set $\widehat{\mu}_{n,r}^{ILR}(X_n) \leftarrow \frac{1}{n}\sum_{k=1}^{n} \widehat{\mu}_{r,X_k}^{LR}(X_n)$ and output.
else EWLR estimator
    for $k = 1, \ldots, n$ **do**
        Set $\widehat{\sigma}_{X_n}^2(X_k) \leftarrow \frac{1}{r}\sum_{j=1}^{r}\left(\frac{h\left(Y_k^{(j)};X_n\right)}{h\left(Y_k^{(j)};X_k\right)} F\left(Y_k^{(j)}\right) - \widehat{\mu}_{r,X_k}^{LR}(X_n)\right)^2$, for $k = 1, \ldots, n$
        Set $\widehat{w}_k^{EWLR} \leftarrow \widehat{\sigma}_{X_n}^{-2}(X_k) / \sum_{i=1}^{n} \widehat{\sigma}_{X_n}^{-2}(X_i)$, for $k = 1, \ldots, n$
    end for
    Set $\widehat{\mu}_{n,r}^{EWLR}(X_n) \leftarrow \frac{1}{n}\sum_{k=1}^{n} \widehat{w}_k^{EWLR} \cdot \widehat{\mu}_{r,X_k}^{LR}(X_n)$ and output
end if

For the MLR estimator $\widehat{\mu}_{n,r}^{MLR}(X_n)$, a green algorithm is especially valuable. Inspection of (2.15) and (2.16) suggests that the MLR estimator requires $n^2 r$ likelihood evaluations: $h\left(Y_k^{(j)}; x_\ell\right)$, for all $j = 1, \ldots, r$ and $k, \ell = 1, \ldots, n$. A quadratic growth rate of computation cost in n could be an obstacle for using the MLR estimator when reusing the output of many experiments. By storing and reusing likelihoods from one experiment to the next in Algorithm 2, we avoid this quadratic growth and achieve linear growth of the computation cost in n, as was the case for the ILR estimator. Algorithm 2 has storage requirement $2nr$ and computation cost $rC_F + (2n-1)rC_h$ for the nth experiment. This result for MLR is similar to the result for ILR, but MLR requires almost twice as many likelihood evaluations.

Algorithm 2 Green implementation of MLR estimator in the nth experiment

Observe X_n and initialize $\widehat{\mu}_{n,r}^{MLR}(X_n) \leftarrow 0$
for $j = 1, \ldots, r$ **do**
  for $k = 1, \ldots, n$ **do**
    if $k < n$ **then**
      Retrieve $F\left(Y_k^{(j)}\right)$ and $\bar{h}(Y_k^{(j)})$ from storage
    else
      Sample $Y_n^{(j)}$ and evaluate $F\left(Y_n^{(j)}\right)$ ▷ rC_F computation
      Append to output storage $F\left(Y_n^{(j)}\right)$ ▷ nr storage
      Set $\bar{h}(Y_n^{(j)}) \leftarrow 0$
      for $\ell = 1, \ldots, n-1$ **do**
        Calculate likelihood $h\left(Y_n^{(j)}; X_\ell\right)$ ▷ $(n-1)rC_h$ computation
        Set $\bar{h}(Y_n^{(j)}) \leftarrow \bar{h}(Y_n^{(j)}) + \frac{1}{n-1} h\left(Y_n^{(j)}; X_\ell\right)$
      end for
    end if
    Calculate likelihood $h\left(Y_k^{(j)}; X_n\right)$ ▷ nrC_h computation
    Set $\bar{h}(Y_k^{(j)}) \leftarrow \frac{n-1}{n} \bar{h}(Y_k^{(j)}) + \frac{1}{n} h\left(Y_k^{(j)}; X_n\right)$
    Update/Append to likelihood storage $\bar{h}(Y_n^{(j)})$ ▷ nr storage
  end for
end for
Set $\widehat{\mu}_{n,r}^{MLR}(X_n) \leftarrow \frac{1}{nr} \sum_{k=1}^{n} \sum_{j=1}^{r} h\left(Y_k^{(j)}; X_n\right) F\left(Y_k^{(j)}\right) / \bar{h}(Y_k^{(j)})$ and output

2.4. Convergence of Likelihood-Ratio-Based Estimators

In this section, we analyze the convergence of the three LR estimators as the number n of experiments grows while the number r of replications per experiment is fixed. To

this end, we make an assumption on the stochastic process $\{X_n : n = 1, 2, \ldots\}$, which determines the sampling distributions.

(C1) The stochastic process $\{X_n : n = 1, 2, \ldots\}$ is ergodic.

We adopt the following definition of ergodicity, which is consistent with the well-known Birkhoff ergodic theorem. For the class of Markov chains, this definition of ergodicity is standard; for a Markov chain to be ergodic, it is sufficient for it to be positive Harris recurrent and aperiodic (Nummelin, 2004; Meyn and Tweedie, 2009).

Definition 2.4.1. *A stochastic process $\{X_n : n = 1, 2, \ldots\}$ taking values in a Polish state space $\mathcal{X}$ is ergodic if*

(i) it has a stationary probability measure π on the Borel σ-algebra of $\mathcal{X}$, and

(ii) for any random variable $f(X)$ that has a finite expectation under π, i.e., $f \in L_1(\mathcal{X}, \mathcal{B}(\mathcal{X}), \pi)$,

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} f(X_k) = \int_{\mathcal{X}} f(x) d\pi(x), \quad \textit{a.s. and in } L_1. \tag{2.20}$$

The reason to make Assumption (C1) is as follows. For any target state $x \in \mathcal{X}$, we envision that it has a neighborhood such that, for every x' in this neighborhood, the sampling distribution associated with x' is a good sampling distribution for the target distribution associated with x. A good sampling distribution would be one for which the target-x-sample-x' variance $\sigma_x^2(x')$ defined in (2.5) is sufficiently small. An ergodic process returns to this neighborhood infinitely often. The consequence is that, as $n \to \infty$, the number of good samples to be used in estimating $\mu(x)$ also grows without bound.

Theorem 2.4.1 shows that, under our assumptions, the conditional variance of the ILR estimator evaluated at a fixed target state x, given the state history $X_1, \ldots, X_n$, goes to zero at the rate $\mathcal{O}(n^{-1})$. Theorem 2.4.1 also provides a result about unconditional variance, which may be easier to interpret.

Theorem 2.4.1. *Suppose that Assumptions (A.1)–(A.6) and (C1) hold, and π is the stationary probability measure of $\{X_n : n = 1, 2, \ldots\}$. For any target state $x \in \mathcal{X}$, if the function σ_x^2 defined in Equation (2.5) is in $L_1(\mathcal{X}, \mathcal{B}(\mathcal{X}), \pi)$, then*

$$\lim_{n\to\infty} nr\mathbb{V}ar\left[\widehat{\mu}_{n,r}^{ILR}(x)\middle|\, X_1, \ldots, X_n\right] = \int_{\mathcal{X}} \sigma_x^2(x')d\pi(x'). \tag{2.21}$$

If, furthermore, the target-x-sample-X_n variance process $\{\sigma_x^2(X_n), n = 1, 2\ldots\}$ is uniformly integrable, then

$$\lim_{n\to\infty} nr\mathbb{V}ar\left[\widehat{\mu}_{n,r}^{ILR}(x)\right] = \int_{\mathcal{X}} \sigma_x^2(x')d\pi(x'). \tag{2.22}$$

Proof. Since the random vectors $\left\{Y_k^{(j)} : j = 1, \ldots, r, k = 1, \ldots, n\right\}$ are conditionally independent given $X_1, \ldots, X_n$, we have

$$\begin{aligned} \mathbb{V}\text{ar}\left[\widehat{\mu}_{n,r}^{ILR}(x)\middle|\, X_1, \ldots, X_n\right] &= \tfrac{1}{n^2r^2}\textstyle\sum_{k=1}^{n}\sum_{j=1}^{r}\mathbb{V}\text{ar}\left[F\left(Y_k^{(j)}\right)\frac{h\left(Y_k^{(j)};x\right)}{h\left(Y_k^{(j)};X_k\right)}\right] \\ &= \tfrac{1}{n^2r}\textstyle\sum_{k=1}^{n}\sigma_x^2\left(X_k\right). \end{aligned} \tag{2.23}$$

Therefore, by ergodicity of $\{X_n : n = 1, 2, \ldots\}$, we obtain (2.21):

$$\lim_{n\to\infty} nr\mathbb{V}\text{ar}\left[\widehat{\mu}_{n,r}^{ILR}(x)\middle|\, X_1, \ldots, X_n\right] = \lim_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n}\sigma_x^2\left(X_k\right) = \int_{\mathcal{X}}\sigma_x^2(x')d\pi(x'), \quad \text{a.s. and in } L_1.$$

To establish (2.22), consider that

$$\begin{aligned}\mathbb{V}\text{ar}\left[\widehat{\mu}_{n,r}^{ILR}(x)\right] &= \mathbb{E}\left[\mathbb{V}\text{ar}\left[\widehat{\mu}_{n,r}^{ILR}(x)\middle| X_1,\ldots,X_n\right]\right] + \mathbb{V}\text{ar}\left[\mathbb{E}\left[\widehat{\mu}_{n,r}^{ILR}(x)\middle| X_1,\ldots,X_n\right]\right] \\ &= \mathbb{E}\left[\mathbb{V}\text{ar}\left[\widehat{\mu}_{n,r}^{ILR}(x)\middle| X_1,\ldots,X_n\right]\right] + \mathbb{V}\text{ar}\left[\mu(x)\right] \\ &= \mathbb{E}\left[\frac{1}{n^2 r}\sum_{k=1}^{n}\sigma_x^2\left(X_k\right)\right],\end{aligned}$$

using (2.23). Therefore

$$\begin{aligned}\lim_{nr\to\infty} nr\mathbb{V}\text{ar}\left[\widehat{\mu}_{n,r}^{ILR}(x)\right] &= \lim_{n\to\infty}\mathbb{E}\left[\tfrac{1}{n}\textstyle\sum_{k=1}^{n}\sigma_x^2\left(X_k\right)\right] \\ &= \mathbb{E}\left[\lim_{n\to\infty}\tfrac{1}{n}\textstyle\sum_{k=1}^{n}\sigma_x^2\left(X_k\right)\right] \\ &= \textstyle\int_{\mathcal{X}}\sigma_x^2(x')d\pi(x'),\end{aligned}$$

where the exchange of limit and expectation holds by uniform integrability, and ergodicity of $\{X_n : n = 1, 2, \ldots\}$ justifies the last step. □

Corollary 2.4.1. *If the corresponding conditions stated in Theorem 2.4.1 hold, then*

$$\lim_{n\to\infty} nr\mathbb{V}ar\left[\widehat{\mu}_{n,r}^{*}(x)\middle| X_1,\ldots,X_n\right] \leq \int_{\mathcal{X}}\sigma_x^2(x')d\pi(x'),\ \textit{and} \tag{2.24}$$

$$\lim_{n\to\infty} nr\mathbb{V}ar\left[\widehat{\mu}_{n,r}^{*}(x)\right] \leq \int_{\mathcal{X}}\sigma_x^2(x')d\pi(x'), \tag{2.25}$$

where the asterisk may represent WLR or MLR.

Proof. Due to Propositions 2.2.2 and 2.2.4, under the appropriate conditions stated in Theorem 2.4.1, the right sides of Equations (2.21) and (2.22) serve as upper bounds for $nr\mathbb{V}\text{ar}\left[\widehat{\mu}_{n,r}^{*}(x)\middle| X_1,\ldots,X_n\right]$ and $nr\mathbb{V}\text{ar}\left[\widehat{\mu}_{n,r}^{*}(x)\right]$, respectively. □

By Theorem 2.4.1 and Corollary 2.4.1, the conditional variance of any of the three LR estimators, evaluated at a fixed target state x, given the state history $X_1, \ldots, X_n$, goes to zero at the rate $\mathcal{O}(n^{-1})$. Because all three LR estimators are unbiased, it follows that they are consistent as $n \to \infty$.

Theorem 2.4.1 and Corollary 2.4.1 show the asymptotic superiority of the LR estimators to standard Monte Carlo (SMC), as the number n of repeated experiments increases. Recall, from the discussion of Equation (2.3), that the conditional variance of the SMC estimator for the current expected performance $\mu(X_n)$, given the state history, is $\sigma^2(X_n)/r$. This does not converge to zero as $n \to \infty$, assuming that sampling variances are positive. Yet the LR estimators converge to the true value $\mu(x)$ as $n \to \infty$. This means that we can obtain arbitarily high accuracy without increasing the budget r per experiment, merely by reusing output from experiments that are repeated at each time step with budget r. Under the assumptions of Theorem 2.4.1, reusing old simulation output is highly effective in the sense that the variance of the LR estimators converges as $\mathcal{O}((nr)^{-1})$, which is the standard rate of convergence for Monte Carlo in terms of the computational budget nr expended on all experiments that were ever run.

2.5. Numerical Examples

In this section, we use two numerical examples to illustrate green simulation via the LR method, demonstrate its value, and compare the three LR estimators with each other and with standard Monte Carlo. First is a reinsurance example of pricing catastrophe bonds. For this example, we verify the conditions of Theorem 2.4.1 in Appendix A, which shows that these conditions are applicable to a realistic example. The experiment results

conform to the theoretical predictions in that the MLR estimator was superior to the ILR estimator, which was superior to the SMC estimator. The EWLR estimator performed similarly to MLR when the number of experiments was small, but it was not even as good as the ILR estimator when many experiments were run.

The second example involves measuring the credit risk of a loan portfolio. In this example, the conditions of Theorem 2.4.1 do not hold, but the experiment results show that green simulation via the LR method can still deliver valuable results in such a situation. Although the ILR and EWLR estimators were not successful in this example, the MLR estimator was superior to the SMC estimator.

2.5.1. Catastrophe Bond Pricing with Compound Losses

A catastrophe bond ("CAT bond") is an important reinsurance contract that helps insurance companies to hedge against losses from catastrophic events (Munich Re Geo Risks Research, 2015). This example is relevant beyond insurance; for example, senior tranches of structured financial instruments are essentially economic catastrophe bonds, because they suffer credit losses only in the event of an economic catastrophe (Coval et al., 2009). Simulation of CAT bonds can be computationally intensive, because it involves fairly rare events in a complex geophysical model. Specifically, the example illustrates a simple simulation for pricing hurricane CAT bonds.

In practice, reinsurance contracts are subject to periodic renewals. This is the source of the repeated experiments: the same CAT bond is priced every period, using the same hurricane simulation model, with parameters X_n updated to reflect the current climatological forecast. In this example, the period is semi-annual. The state process

$\{X_n : n = 1, 2, \ldots\}$ is modeled by an ergodic Markov chain. Given the state X_n, we can simulate hurricanes that take place during the lifetime of the CAT bond and the resulting total insured loss Y_n underlying the CAT bond. Finally, we compute the payoff of the CAT bond per dollar invested,

$$F(Y_n) = \mathbf{1}_{\{Y_n \leq K\}} + p\mathbf{1}_{\{Y_n > K\}},$$

where $\mathbf{1}_{\{.\}}$ is the indicator function, K is the trigger level, and $p \in [0, 1)$ is the fraction of face value that is received if insured losses exceed the trigger level. The fair price of this CAT bond can be obtained in terms of the expected payoff $\mu(X_n) = \mathbb{E}[F(Y_n)|X_n]$. We consider a hurricane CAT bond with lifetime 10 years, trigger level $K = 25$ million dollars, and recovery fraction $p = 0.5$.

In this example, we use a simplified version of the model of Dassios and Jang (2003). The insured loss is modeled as a compound random variable: $Y_n = \sum_{i=1}^{M_n} Z_n^i$, where M_n denotes the number of claims and Z_n^i denotes the ith claim size. In this model, Z_n^i, $i = 1, 2 \ldots$ are i.i.d. and independent of M_n. This is a popular loss model due to its flexibility and suitability for many practical applications (Klugman et al., 2012), yet it can provide mathematical tractability. Let the probability mass function of M_n be $p(m; \lambda_n)$ and the probability density of Z_n^i be $f(z; \theta_n)$, where λ_n and θ_n are parameters determined by the state X_n. Specifically, we take M_n to be Poisson with mean λ_n and Z_n^i to be exponential with mean θ_n. In this model, the expected payoff is

$$\mu(X_n) = \mathbb{E}[F(Y_n)|X_n] = \mathbb{E}\left[p + (1-p)\mathbf{1}_{\{Y_n \leq K\}}\right] = p + \sum_{m=1}^{\infty} p(m; \lambda_n) F(K; \theta_n, m) \tag{2.26}$$

where $p(m;\lambda)$ is the Poisson probability mass function with mean λ and $F(K;\theta,m)$ is the Gamma cumulative distribution function with scale parameter θ and shape parameter m. From 1981-2010, the average number of major hurricanes was 2.7 per decade and the average cost per hurricane was about \$5,000 million (Blake and Gibney, 2011). Therefore, we set up a stochastic model of the states $\{X_n : n = 1, 2, \ldots\}$ and a transformation $(\lambda, \theta) = \psi(x)$ so that λ_n is usually around 2.7 and θ_n is usually around 5 (measured in thousands of millions of dollars).

In this example, the ergodic Markov chain driving the parameters of the loss model is a stationary AR(1) process with state space $\mathcal{X} = \mathbb{R}^2$, given by

$$X_n = \mu_\infty + \varphi X_{n-1} + \varepsilon_n,$$

where $\{\varepsilon_n : n = 1, 2, \ldots\}$ is an i.i.d. sequence of bivariate normal random vectors with mean zero and variance $\mathrm{diag}(\sigma_\varepsilon^2)$, and the parameters are

$$\mu_\infty = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \varphi = \begin{bmatrix} 0.6 \\ 0.5 \end{bmatrix}, \quad \text{and} \quad \sigma_\varepsilon^2 = \begin{bmatrix} 0.8^2 \\ 0.5^2 \end{bmatrix}.$$

The state space $\mathcal{X} = \mathbb{R}^2$ is inappropriate for parameters that must be non-negative because they represent an expected number of hurricanes and an expected loss per hurricane. We introduce a transformation $\psi : \mathbb{R}^2 \mapsto (\underline{\lambda}, \bar{\lambda}) \times (\underline{\theta}, \bar{\theta})$ so that the parameters $(\lambda, \theta) = \psi(x)$ lie between plausible lower and upper bounds. The transformation is sigmoidal and maps $x = [x^1, x^2]$ to

$$(\lambda, \theta) = \psi(x) = \left[\underline{\lambda} + \tfrac{\bar{\lambda} - \underline{\lambda}}{1 + e^{-x^1}}, \quad \underline{\theta} + \tfrac{\bar{\theta} - \underline{\theta}}{1 + e^{-x^2}} \right].$$

In particular, we took $(\underline{\lambda}, \bar{\lambda}) = (2, 4)$ as the range for expected number of hurricanes per decade and $(\underline{\theta}, \bar{\theta}) = (4, 6)$ as the range for expected loss per hurricane. To give a picture of the variability of the parameters for repeated experiments in this example, Figure 2.1 shows histograms of the parameters λ and θ resulting from sampling from the stationary distribution of the AR(1) process.

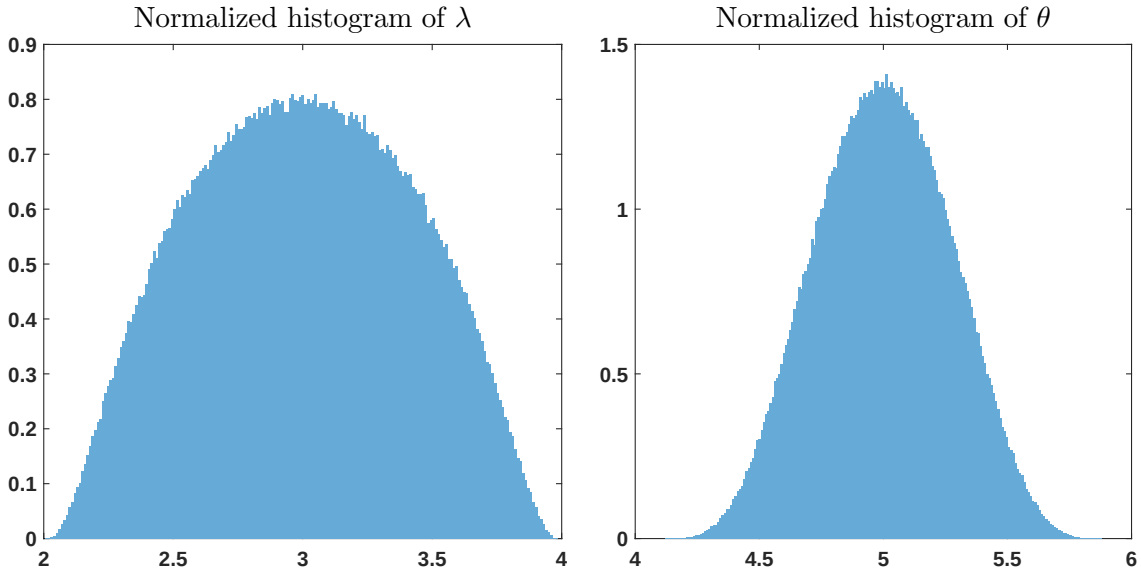


Figure 2.1. Histograms of parameters sampled based on the stationary distribution of the AR(1) process.

To clarify the model, Algorithm 3 shows how the standard Monte Carlo simulation works.

Algorithm 3 Standard Monte Carlo Simulation for Catastrophe Bond Pricing

Sample state X_0 from the stationary distribution of the AR(1) process
for $n = 1, 2, \ldots,$ **do**
    Sample state X_n from AR(1) process, conditional on X_{n-1}
    Set parameters $(\lambda_n, \theta_n) \leftarrow \psi(x_n)$
    Set $\widehat{\mu}_{n,r}^{SMC}(X_n) \leftarrow 0$
    for $j = 1, \ldots, r$ **do**
        Sample number of hurricanes $M_n^{(j)} \sim p(\cdot; \lambda_n)$
        for $i = 1, \ldots, M_n^{(j)}$ **do**
            Sample loss for ith hurricane $Z_n^{(i,j)} \sim f(\cdot; \theta_n)$
        end for
        Set $Y_n^{(j)} \leftarrow \sum_{i=1}^{M_n} Z_n^{i,j}$
        if $Y_n^{(j)} \leq K$ **then**
            Set $F\left(Y_n^{(j)}\right) \leftarrow 1$
        else
            Set $F\left(Y_n^{(j)}\right) \leftarrow p$
        end if
        Set $\widehat{\mu}_{x_n}^{SMC}(X_n) \leftarrow \widehat{\mu}_{x_n}^{SMC}(X_n) + F\left(Y_n^{(j)}\right)$
    end for
    Set $\widehat{\mu}_{n,r}^{SMC}(X_n) \leftarrow \widehat{\mu}_{n,r}^{SMC}(X_n)/r$ and output
end for

To investigate the effectiveness of green simulation, we performed a sequence of 100 repeated simulation experiments (i.e., $n = 1, 2, \ldots, 100$) with $r = 100$ replications each. Using the same sample path $\{X_n : n = 1, 2, \ldots, 100\}$ and the same simulation output $\left\{Y_n^{(j)} : n = 1, 2, \ldots, 100, j = 1, 2, \ldots, 100\right\}$, we evaluated the SMC, ILR, EWLR and MLR estimators at each period $n = 1, 2, \ldots, 100$, in each of three states: the current state

X_n, the central state $x_{mi} = (0, 0)$ corresponding to the moderate parameters $\lambda = 3$ and $\theta = 5$, and an extreme state $x_{hi} = (\infty, \infty)$ corresponding to extreme parameters $\lambda = 4$ and $\theta = 6$[1].

For the purpose of accurately estimating the unconditional variances of the all these estimators, we performed such a sequence of experiments 10,000 times. These 10,000 macro-replications of the sequence of experiments have independent sample paths and simulation output. The estimated variance of a fixed-state estimator $\widehat{\mu}(x)$ of $\mu(x)$ was $\sum_{k=1}^{10,000} (\widehat{\mu}^{(k)}(x) - \mu(x))^2/10,000$, where $\widehat{\mu}^{(k)}(x)$ is the value of the estimator on the kth macro-replication. Likewise, the estimated variance of a current-state estimator $\widehat{\mu}(X_n)$ of $\mu(X_n)$ was $\sum_{k=1}^{10,000} (\widehat{\mu}^{(k)}(X_n^{(k)}) - \mu(X_n^{(k)}))^2/10,000$. Due to using 10,000 macro-replications, the standard errors of these estimated variances are less than 1% of the corresponding estimated variance.

Figure 2.2 is a log-log plot of the variances of the SMC, ILR, EWLR, and MLR estimators for two fixed states, x_{mi} and x_{hi}, for each experiment $n = 1, 2, \ldots, 100$. The fixed-state SMC estimators $\widehat{\mu}_{n,r}^{SMC}(x_{mi})$ and $\widehat{\mu}_{n,r}^{SMC}(x_{hi})$ were generated by sampling according to $h(\cdot; x_{mi})$ and $h(\cdot; x_{hi})$, respectively, in experiments distinct from the experiments described in the preceding paragraph. The SMC variance forms a horizontal line because, for each experiment n, $\widehat{\mu}_{n,r}^{SMC}(x_{mi})$ and $\widehat{\mu}_{n,r}^{SMC}(x_{hi})$ use a fixed number r of replications drawn from the sampling distribution associated with the fixed state x_{mi} or x_{hi}. In addition, a black solid line with slope -1 and intercept equal to the SMC variance is plotted for reference. This line shows the variance of an SMC estimator with nr replications, which is the cumulative number of replications simulated up through the nth experiment. We compare

[1]Strictly speaking, the extreme state x_{hi} does not belong to the state space $\mathcal{X} = \mathbb{R}^2$, but the parameter vector $(\lambda, \theta) = (4, 6)$ is a limit point of the range of the transformation ψ that maps states to parameters.

the lines for green simulation estimators against the solid lines. When they go below the horizontal line, they have lower variance than a SMC simulation with r replications, the budget for a single experiment. When they go near the line with slope -1, they have variance nearly as low as a SMC simulation with nr replications, the cumulative budget for all experiments so far. That is a major success for green simulation, because it shows that reusing old simulation output is nearly as effective as generating new simulation output in the current experiment.

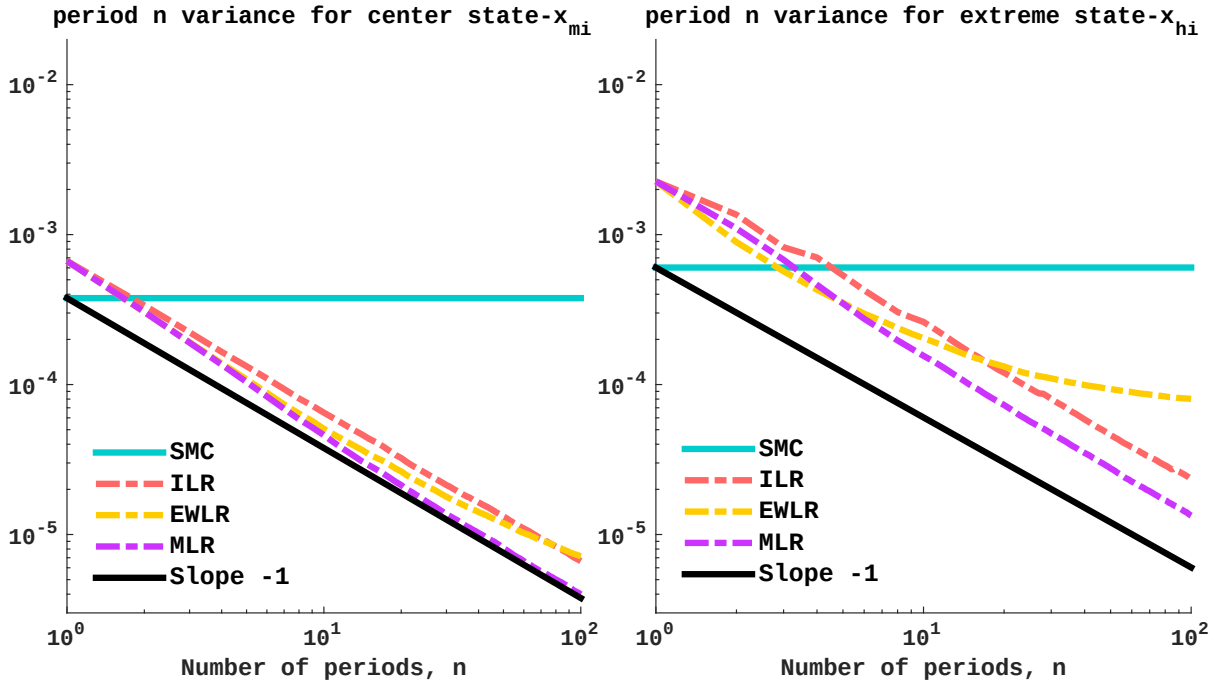


Figure 2.2. Log-log plots of estimated variances of fixed-state SMC and LR estimators for CAT bond pricing example.

We see from Figure 2.2 that the MLR estimator has lower variance than the ILR estimator, which is consistent with (2.18). In this example, the gap between them grows

to be substantial as the number n of repeated experiments increases: for $n = 100$, the ratio between ILR and MLR variances is about 1.7 for both x_{mi} and x_{hi}. Initially, for very small n, the LR estimators have higher variances than an SMC estimator based on a simulation in the fixed state x_{mi} or x_{hi}. The cause is the probability that none of the states visited so far, $X_1, X_2, \ldots, X_n$, were near x_{mi} or x_{hi}. This event is more likely for the extreme state x_{hi} than for the center state x_{mi}, which is why the higher variances persist longer for the extreme state (for the first 5 experiments) than for the center state (only for the first experiment). The variances of the green simulation soon become lower than those of SMC and continue to decrease as the outputs of more experiments are reused. After 100 experiments, the MLR estimator's variance is over 45 times smaller than the SMC estimator's variance for x_{hi} and over 95 times smaller for x_{mi}. In other words, by using 10,000 total replications simulated in 100 simulation experiments, with simulation based on parameters corresponding to $X_1, X_2, \ldots, X_{100}$ and not x_{mi}, the MLR estimator achieves higher accuracy in estimating $\mu(x_{mi})$ than standard Monte Carlo with $9,500$ replications simulated based on parameters corresponding to x_{mi}. Comparing to the black solid reference line, we see that the green simulation estimators' variances eventually decrease approximately at a rate of n^{-1}, as discussed in connection with Theorem 2.4.1. For small n, the EWLR estimator has variance comparable to that of the MLR estimator, but it is not competitive for large n. In the extreme state, the EWLR variance grows much larger than the ILR variance for large n, and it does not appear to converge as $\mathcal{O}(n^{-1})$.

Next we consider the variance of the SMC and the LR estimators for the current-state expected performance $\mu(X_n)$. Figure 2.3 shows their variances. For the first experiment ($n = 1$), there is no stored simulation output from a previous experiment, so all LR

estimators are the same as the SMC estimator. In this example, the green simulation estimators' variances decrease from the beginning and are always less than the SMC variance. Otherwise, the performance of the green simulation estimators is similar to what was seen when the state was fixed. After 100 experiments, the MLR estimator's variance is over 61 times smaller than the SMC estimator's variance.

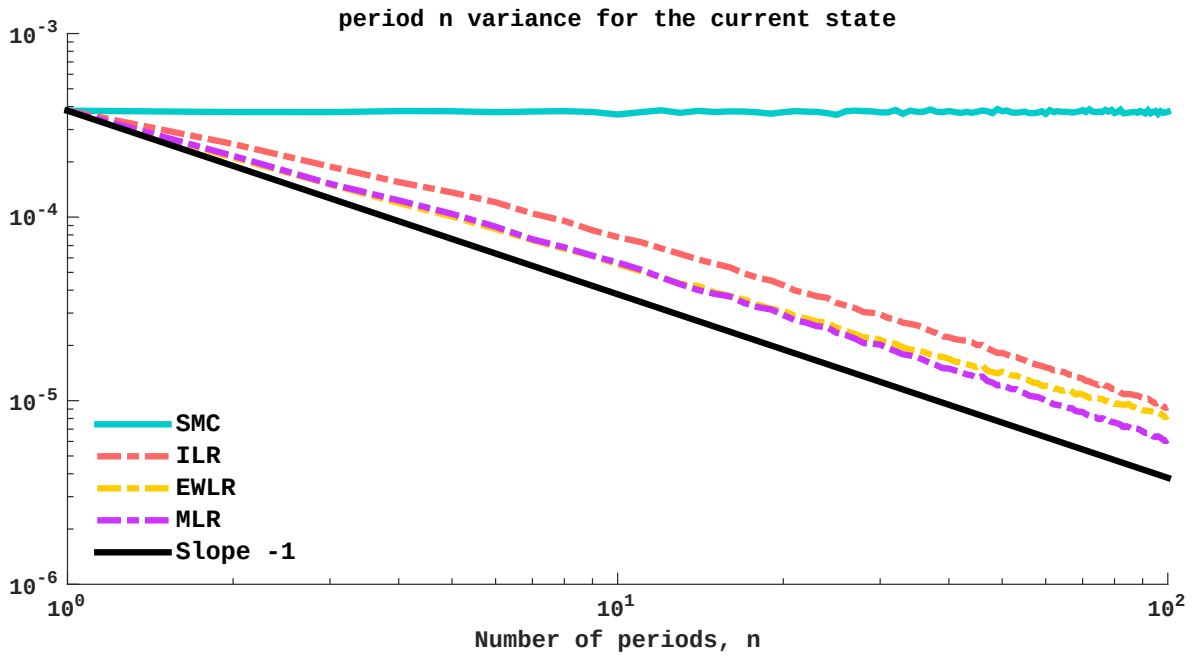


Figure 2.3. Log-log plot of estimated variance for current-state SMC and LR estimators for CAT bond pricing example.

2.5.2. Credit Risk Management

In simulation for financial risk management, experiments using the same simulation model are performed periodically, as often as daily. Information observed in the markets is used to update parameters that affect risk, and the simulation model is run again with new

parameters. In this example, the risk management simulation measures the credit risk exposure of a portfolio containing loans to companies with listed equity. The asset values of these debtor companies are observable and serve as parameters in the risk model: the lower the asset value of a debtor company, the more likely it is to default in the future. Thus, in our setting of repeated experiments, the current state X_n contains the current asset values of all debtor companies.

In this example, we work with a structural model of default based on the influential work of Merton (1974); for an exposition, see McNeil et al. (2005), for example. At any period n, the asset value of a company equals the sum of its equity and debt values. The equity value can be observed in the stock market and the debt value can be observed from public records, so the asset value can be computed. For simplicity of exposition, we assume that debt remains constant. In this model, the asset value follows geometric Brownian motion, and a company defaults when its asset value falls below its debt. Because geometric Brownian motion is not an ergodic process, Theorem 2.4.1 and Corollary 2.4.1 do not apply. As n increases, the current state X_n tends to drift further away from an earlier state, such as X_1. Therefore, intuition suggests that the benefit of reusing old simulation output would diminish over time. We consider this example to show that green simulation via the LR method nonetheless delivers valuable results.

We consider a loan portfolio whose composition remains constant over time. Many loan portfolios are dynamic: as outstanding loans are being repaid, new loans are being initiated. Despite the dynamic nature of such portfolios, there are lending businesses in which portfolios retain a similar composition over time. For example, in the business of accounts receivable, customers may place regular, periodic orders of the same size, each

resulting in payment due in 90 days. An investment fund may target a loan portfolio with fixed characteristics such as maturity and portfolio weights on different types of loans.

In our simplified example, we consider risk management of the value at time horizon $t = 0.5$ years of a portfolio in which there are two loans, both having maturity $T = 5$ years. Simulation experiments are repeated weekly, i.e., with a period of $\Delta t = 1/52$ years. In the nth experiment, the quantity $\mu(X_n)$ being estimated is the conditional probability, given the current asset values X_n, that the cost of defaults and anticipated defaults after t years will exceed a threshold $\kappa = 6$. The random vector Y_n that we simulate in the nth experiment is the asset values $S_t = [S_{t,1}, S_{t,2}]$ at time t, given the current asset values $S_0 = [S_{0,1}, S_{0,2}] = X_n$. For each company $i = 1, 2$, the marginal distribution of the asset return $S_{t,i}/S_{0,i}$ is lognormal, determined by the drift η_i and volatility ς_i of the geometric Brownian motion for asset value. Specifically, $\eta_1 = 15\%$, $\eta_2 = 10\%$, $\varsigma_1 = 30\%$, and $\varsigma_2 = 20\%$. The joint distribution of the asset returns $S_{t,1}/S_{0,1}$ and $S_{t,2}/S_{0,2}$ is specified by a Student t copula with 3 degrees of freedom and correlation 0.5. The initial asset values of the two companies are $X_{0,1} = 100$ and $X_{0,2} = 90$, and their debt is $D_1 = D_2 = 85$. The loss if company i defaults is denoted a_i, and $a_1 = 5$ and $a_2 = 4$. The discount rate is $r = 5\%$. The cost of a default or anticipated default by company i, as of time t, is

$$L_i = \ell_i(S_{t,i}) = \begin{cases} a_i & \text{if } S_{t,i} < D_i \\ a_i e^{-r(T-t)} \Phi\left(\frac{\ln(S_{t,i}/D_i) + (r - \varsigma_i^2/2)(T-t)}{\varsigma_i \sqrt{T-t}} \right) & \text{if } S_{t,i} \geq D_i \end{cases} .$$

The first line of the formula represents the loss if company i defaults at time t. The second line of the formula represents a risk-neutral conditional expectation, as of time t, of the discounted loss at time T if company i defaults then. The portfolio's loss is $L_1 + L_2$, the

sum of losses over the debtor companies. The simulation output $F(Y_n)$ is 1 if $L_1 + L_2 > \kappa$, and 0 otherwise. To clarify the model, Algorithm 4 shows how the standard Monte Carlo simulation works.

Algorithm 4 Standard Monte Carlo Simulation for Credit Risk Management

for $n = 1, 2, \ldots,$ **do**
  Sample state X_n from bivariate lognormal distribution with Student-t copula, based on time increment Δt, conditional on X_{n-1}
  Set $\widehat{\mu}_{n,r}^{SMC}(X_n) \leftarrow 0$
  for $j = 1, \ldots, r$ **do**
    Sample state $Y_n^{(j)}$ from bivariate lognormal distribution with Student t copula, based on time increment t, conditional on X_n
    Set $L_n^{(j)} \leftarrow \ell_1(Y_{n,1}^{(j)}) + \ell_2(Y_{n,2}^{(j)})$
    if $L_n^{(j)} \leq \kappa$ **then**
      Set $F\left(Y_n^{(j)}\right) \leftarrow 1$
    else
      Set $F\left(Y_n^{(j)}\right) \leftarrow 0$
    end if
    Set $\widehat{\mu}_{x_n}^{SMC}(X_n) \leftarrow \widehat{\mu}_{x_n}^{SMC}(X_n) + F\left(Y_n^{(j)}\right)$
  end for
  Set $\widehat{\mu}_{n,r}^{SMC}(X_n) \leftarrow \widehat{\mu}_{n,r}^{SMC}(X_n)/r$ and output
end for

To investigate the effectiveness of green simulation, we performed a sequence of 52 simulation experiments repeated weekly (i.e., $n = 1, 2, \ldots, 52$) with $r = 1,000$ replications each. Using the same sample path $\{X_n : n = 1, 2, \ldots, 52\}$ and the same simulation output $\left\{Y_n^{(j)} : n = 1, 2, \ldots, 52, j = 1, 2, \ldots, 100\right\}$, we evaluated the SMC and the LR estimators at each period $n = 1, 2, \ldots, 52$, in the current state X_n. For the purpose of accurately

estimating the unconditional variances of the all these estimators, we performed such a sequence of experiments 10,000 times. These 10,000 macro-replications of the sequence of experiments have independent sample paths and simulation output. The estimated variance of a current-state estimator $\widehat{\mu}(X_n)$ of $\mu(X_n)$ was $\sum_{k=1}^{10,000} (\widehat{\mu}^{(k)}(X_n^{(k)}) - \mu(X_n^{(k)}))^2/10,000$. These estimated variances appear in a log-log plot in Figure 2.4, along with vertical error bars representing their 95% approximate-normal confidence intervals.

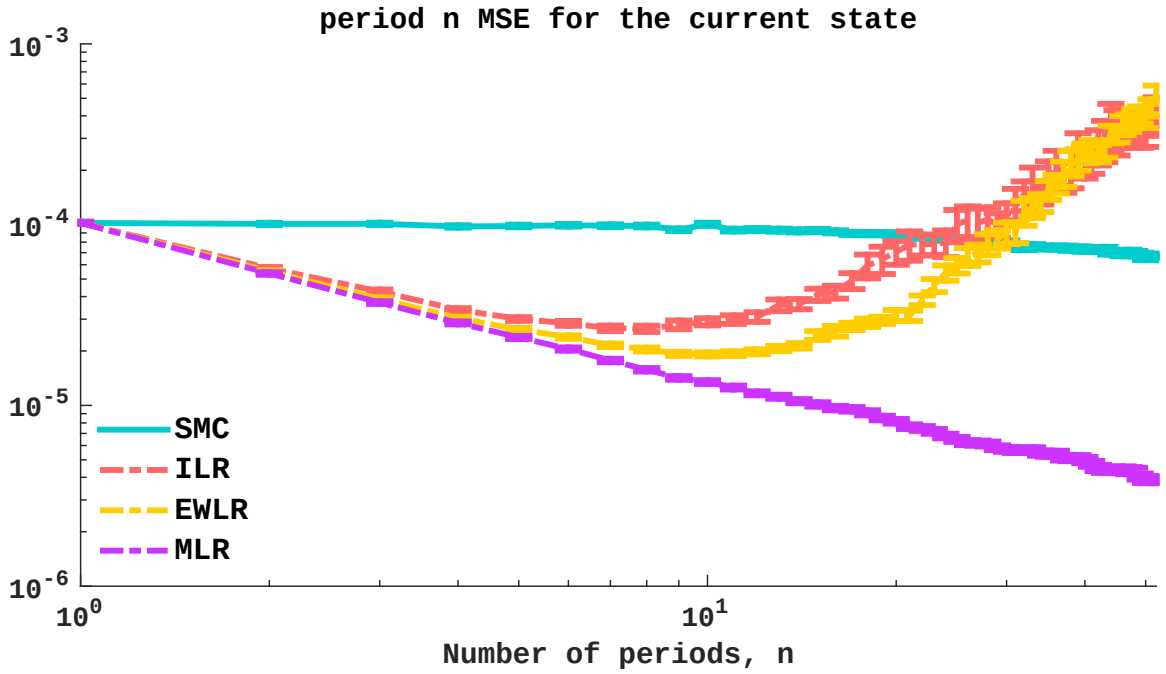


Figure 2.4. Log-log plot with error bars for estimated variances of the SMC and LR estimators for current state estimators for credit risk example.

Figure 2.4 shows behavior for the MLR estimator similar to what was seen for the previous example in Figure 2.3. The MLR estimator's variance is less than the SMC estimator's variance, and it decreases as the number n of repeated experiments increases.

By $n = 52$, it is over 17 times smaller than the SMC estimator's variance. However, in Figure 2.4, we see effects of the non-ergodic nature of the state process $\{X_n : n = 1, 2, \ldots\}$. Due to the positive drifts of the asset prices, debtor companies tend to become less likely to default, so the SMC variance decreases slightly as the number of periods n increases, instead of forming a straight line as in Figure 2.3. The dramatic effect is on the behavior of the ILR and EWLR estimators. Their variances decrease over the first 6 experiments, but after about 20 experiments, its variance increases again. Eventually, their variances exceed the SMC estimator's variance. Apparently, the difference between sampling distributions for state X_1 and X_{52} is likely to become so large that the use of likelihood ratios in the ILR estimator (2.7) inflates variance. In such a situation, Proposition 2.2.2 states that the WLR estimator is no worse than the SMC estimator. However, Proposition 2.2.2 does not apply to the EWLR estimator, whose weights involve variance estimates. When a large variance is underestimated, the associated weight is too large, leading to inflated variance. The failure of the ILR and EWLR estimators and the success of the MLR estimator in this example demonstrate the practical importance of the MLR estimator. The evident disutility of some of the old simulation output in this example also raises a future research question in green simulation: how to determine which old simulation output is worthwhile to reuse in estimating the expected performance in the current state.

2.6. Conclusions and Future Research

In Section 2.4, we established theorems about the convergence of green simulation estimators as the number of repeated experiments increases. We tested their practical performance for small and moderate numbers of repeated experiments in two examples

in Section 2.5. In the example of Section 2.5.1, the conditions of the theorems held. All our LR estimators were successful in significantly reducing variance compared to standard Monte Carlo, but MLR was the best. In the example of Section 2.5.2, the conditions of the theorems did not hold. The MLR estimator was successful in significantly reducing variance, whereas the ILR and EWLR estimators had problems. Consequently, among the three LR estimators, we recommend the MLR estimator for doing green simulation in the setting where simulation experiments are repeated with changes to parameters of distributions of random variables generated in the simulation. The variance reduction achieved by the LR estimators depends on several aspects of the particular example: the stochastic process that describes changing parameters, the particular distributions whose parameters change, and the number of repeated experiments. Under the conditions of Theorem 2.4.1, as the number of repeated experiments increases, the LR estimators eventually become greatly superior to standard Monte Carlo. Our experiment results suggest that green simulation is extremely promising: in the only two examples that we investigated, the MLR estimator achieved variance lower than standard Monte Carlo by factors of 17 and 61 after a moderate number of repeated experiments.

Because green simulation is a new paradigm, there are several good directions for future research. Here we call attention to a few that are most relevant to this chapter.

Some future research topics are relevant to the specific methods proposed in this article. We found the MLR estimator to be satisfactory for our purposes. However, further enhancements have been considered in the literature on importance sampling. For example, Hesterberg (1995) investigated different ways to normalize weights, and Owen and Zhou (2000) proposed to use likelihoods that appear in the MLR estimator as control

variates. At the end of Section 2.5.2, the experiment results raised the question of which old simulation output to reuse in green simulation. This question is worthy of investigation in connection with the methods proposed in this article and also with other methods. In general, there are two possible drawbacks to using all of the old simulation output. One is that if the amount of old simulation output is extremely large, reusing more of it generates diminishing returns in terms of improved estimator quality compared to the computational cost of reusing it. To limit the computational cost of reuse, one could employ hard thresholding: reusing only that subset of the old simulation output that is deemed to be sufficiently relevant for estimating the expected performance for the target state. The selection criterion could be, for example, the similarity of the sampling density $h(\cdot; X_i)$ to the target density $h(\cdot; x)$ for LR estimators, or the distance from the previous state X_i to the target state x. The other drawback to using all of the old simulation output is that some of the old simulation output makes an estimator worse if it is reused than if it is not reused, as seen in Section 2.5.2. To cope with this problem, one may employ hard thresholding or soft thresholding, meaning assigning weight to simulation output in proportion to its perceived relevance. For example, the WLR estimator can be viewed as a soft thresholding method which gives smaller weights to outputs that are associated with high variances. The theoretical and practical benefit of various green simulation methods could be enhanced by good rules for selecting the subset of old simulation output to reuse.

We focused on showing that when old simulation output is reused well, it provides greater accuracy when combined with a new simulation experiment than would be achieved by the same new simulation alone. We analyzed how the accuracy of an answer to the current question improves as the number of repeated experiments increases. However, it

might be possible to answer the current question sufficiently accurately with no further experiment. If a new simulation experiment was indeed required, one could design it in light of the current question and the currently available information. This leads to future research in experiment design not from a blank slate. Also, one might consider of possible future questions when designing the current experiment, in light of knowledge of the state process.

As a new paradigm, one may consider other green simulation methods to efficiently reuse simulation output, which may have different features and merits than the LR estimators studied in this chapter. The LR estimators are directly applicable only when simulation experiments are repeated with changes to parameters of distributions of random variables generated in the simulation, not when a changing parameter affects something other than a distribution, e.g., numbers of servers or sizes of buffers in a simulation of a queuing system. Even if these estimators are applicable, they would not be highly effective if it is unlikely to visit a state that is sufficiently similar to a previously visited state, where similarity is measured according to (2.5). Rubinstein and Shapiro (1993) address the likelihood ratio method's effectiveness and extensions of its applicability. More broadly applicable green simulation methods can be designed based on metamodeling, which is studied in the next chapter.

CHAPTER 3

Green Simulation via Metamodeling

3.1. Introduction

The essence of *green simulation* is to improve the computational efficiency of a simulation experiment by reusing the simulation output of repeated experiments; the LR method studied in Chapter 2 is only one of many possible method for this new simulation analysis and design paradigm. There are situations in which Assumptions (A.1)-(A.6) do not hold and the LR method is inapplicable because the current state affects the simulation model not only by affecting a parameter of a likelihood. It may also be that the LR method is applicable, but the target-X_n-sample-$X_{n'}$ variances $\sigma^2_{X_n}(X_{n'})$ are large, making the LR estimators ineffective.

In this chapter we consider a different green simulation method, which reuses simulation output via metamodeling. Specifically, we propose two metamodeling-based green estimators that showcase the generality of green simulation in two ways:

(1) Green simulation entails methods for reusing simulation outputs other than the LR method, whose requirements such as Assumptions (A.1)–(A.6) are not limitations of green simulation in general.

(2) Green simulation inspires reusing simulation output not only for repeated Monte Carlo experiments but also for more complicated experiments such as the nested simulation procedure proposed by Liu and Staum (2010).

Simulation metamodeling is a popular technique that can provide robust and fast decision support to enhance the effectiveness of the decision-making processes (see Barton and Meckesheimer, 2006, for a review). A metamodel (model of a simulation model) aims to provide an accurate and inexpensive approximation to the usually unknown input-output relationship, or the *response surface*, dictated by the underlying simulation model. By input-output relationship, we mean the relationship between an input state x and the expected value of the simulation output $\mu(x)$, or expected output for short.

Figure 3.1 depicts a setting of repeated experiment and defines suitable notation and terminology for the current chapter. It differs from Figure 1.1 in the amalgamation of random vector Y and the simulation logic F into a general simulation replication $\widehat{\mu}_j(x)$, which denotes the jth replication of an experiment with input x. These replications are then used to calculate the simulation output $\widehat{\mu}(x)$, which is an estimate of the *expected* simulation output $\mu(x)$. Unlike the LR method, metamodeling allows the input x to affect both the likelihood of the random vector and the parameters of the simulation logic, so long as the simulation input-output relationship μ remains unchanged.

A usual application of metamodeling involves first running a simulation experiment with different values of its inputs. Then, using the simulation output $\widehat{\mu}(x)$, which may be the sample average of the replications, a metamodel $\widetilde{\mu}(x)$ is fitted to approximate the input-output relationship $\mu(x)$. Finally the fitted metamodel is used for estimation, optimization, sensitivity analysis, or any other task that were originally asked of the simulation experiment (see Chambers and Mount-Campbell, 2002; Kleijnen, 1998, for example). There are choices of metamodeling techniques such as kriging (Kleijnen, 2009), neural networks (Fonseca et al., 2003), and stochastic kriging (Ankenman et al., 2010),

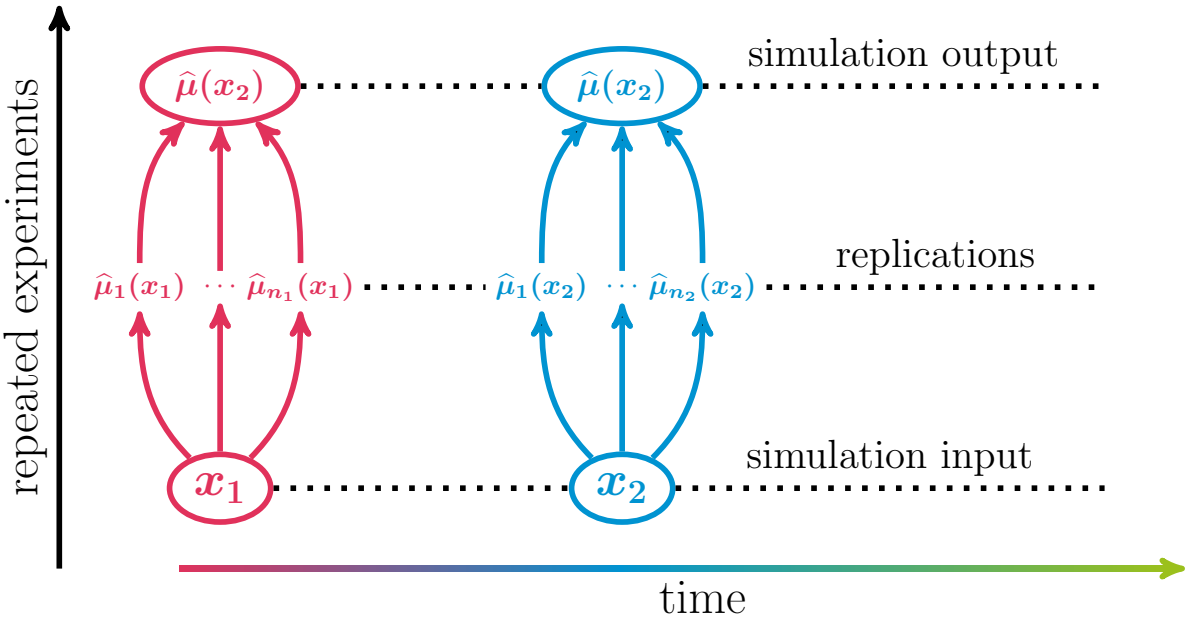


Figure 3.1. Setting of repeated simulation experiments for green simulation via metamodeling.

each with its own features and limitations. As a first proposal for green simulation via metamodeling, in this chapter we consider stochastic kriging for reusing simulation output. Section 3.2 provides a review of stochastic kriging. While stochastic kriging is the only metamodel considered in the following discussion, we acknowledge that there exist other metamodels and techniques, such as updating ordinary least squares regression (Aiken et al. (1991); Belsley et al. (2005)), that are suitable for sequential experiments.

In the setting of repeated experiments, we propose reusing the simulation output from the previous and current experiments in a metamodel to enhance the computational efficiency of the current experiment. In Section 3.3 we propose a green stochastic kriging (GSK) estimator that reuses the simulation output of repeated Monte Carlo experiments. We compare this GSK estimator to the LR estimators in the same two examples considered

in Section 2.5. We find that, in the catastrophe bond pricing example, the GSK estimator is usually worse than the LR estimators in terms of mean squared error (MSE), but it successfully reduces the MSE as more simulation outputs are reused. While the GSK estimator is unsuccessful in the credit risk example, it reveals limitations of the GSK estimator and inspires future research directions. In Section 3.4 we reuse the simulation output of a nested simulation procedure for estimating expected shortfall, which is proposed by Liu and Staum (2010). Our numerical study shows that, after reusing the output of 15 experiments, the accuracy of our green nested simulation procedure improves upon its baseline procedure by more than an order of magnitude.

3.2. Stochastic Kriging

In this section we review stochastic kriging based on Ankenman et al. (2010) and Liu and Staum (2010). We use stochastic kriging to reuse the output of repeated Monte Carlo experiment in Section 3.3 and develop a green nested simulation procedure based on the latter in Section 3.4.

Stochastic kriging is an interpolation-based metamodeling technique. It is suitable for simulation metamodeling because it takes account of both *extrinsic uncertainty* and *intrinsic uncertainty* of simulation outputs.

In stochastic kriging, the expected simulation output $\mu(x)$, or the response surface, is modeled as

$$\mu(x) = f(x)^\top \beta + \mathsf{M}(x) \tag{3.1}$$

where $f(x)$ is a vector of known functions of x, β is a vector of unknown parameters of compatible dimension, and M is a *Gaussian random field (GRF)* with mean zero. Treating M as a GRF captures the uncertainty about the response surface before running simulations, which is called the *extrinsic uncertainty*. In this chapter, we consider both the commonly used second-order stationary GRF with Gaussian covariance function (Ankenman et al., 2010) and the recently developed generalized integrated Brownian field (Salemi et al., 2013). While selection of appropriate GRF in different applications is not the focus of this thesis, in a catastrophe bond pricing example, we found that generalized integrated Brownian field produces markedly superior results than that of the Gaussian covariance function.

In addition to extrinsic uncertainty, the *intrinsic uncertainty* in stochastic kriging captures the Monte Carlo noise in a stochastic simulation. The jth simulation replication for given input x is modeled as

$$\mu_j(x) = f(x)^\top \beta + \mathsf{M}(x) + \epsilon_j(x) \tag{3.2}$$

where $\epsilon_1(x), \epsilon_2(x), \cdots$ are assumed to be normally distributed noise with mean zero and variance $V(x)$, and independent of each other and of M. The simulation output at input x_i after n_i replications is given by $\widehat{\mu}(x_i) = \frac{1}{n_i}\sum_{j=1}^{n_i} \mu_j(x_i)$, which is the standard Monte Carlo estimator of the expected simulation output, denoted by $\widehat{\mu}_{n_i}^{SMC}(x_i)$. The following assumption supports the use of stochastic kriging metamodel.

(B1) Given any input state x, the simulation output $\widehat{\mu}(x)$ is normally distributed with finite mean $\mu(x)$ and finite variance $\sigma_x^2(x)$.

The reason to make Assumption (B1) is that Ankenman et al. (2010) showed that in this case the stochastic kriging estimator has optimality properties.

Suppose that the simulation model has been run for k inputs $x_1, \ldots, x_k$, referred to as *design points*, and let $\widehat{\mu}^k = [\widehat{\mu}(x_1), \cdots, \widehat{\mu}(x_k)]^\top$ represent the corresponding vector of simulation outputs, with n_i replications for input x_i. We use stochastic kriging metamodel to estimate the expected simulation output for a set of input values $X_1, \ldots, X_K$, referred to as the *prediction points*. Denote the vector of expected simulation outputs at the design points and at the predictions by $\mu^k = [\mu(x_1), \ldots, \mu(x_k)]^\top$ and $\mu^K = [\mu(X_1), \ldots, \mu(X_K)]^\top$, respectively. Based on the GRF, one can compute the $k \times k$ matrix $\Sigma^{kk} = \mathrm{Cov}[\mu^k, \mu^k]$ of extrinsic covariances among μ^k, the $k \times K$ matrix $\Sigma^{kK} = (\Sigma^{Kk})^\top := \mathrm{Cov}[\mu^k, \mu^K]$ of extrinsic covariances μ^k and μ^K, and the $K \times K$ matrix $\Sigma^{KK} = \mathrm{Cov}[\mu^K, \mu^K]$ of extrinsic covariances among μ^K. Because experiments for different inputs are independent, the intrinsic covariance matrix for $\widehat{\mu}^k - \mu^k$ is diagonal. It equals $C := VN^{-1}$ where V and N are diagonal matrices whose ith elements are receptively $V(x_i)$ and n_i. For brevity we write $\Sigma := \Sigma^{kk} + C$ as the sum of intrinsic and extrinsic covariances for the design points. Ankenman et al. (2010) show that the stochastic kriging estimator for μ^K follows a multivariate normal distribution with mean

$$\widetilde{\mu}^K = [f(X_1)^\top \beta, \ldots, f(X_K)^\top \beta]^\top + \Sigma^{Kk}\Sigma^{-1}\left(\widehat{\mu}^k - [f(x_1)^\top \beta, \ldots, f(x_k)^\top \beta]^\top\right) \tag{3.3}$$

and variance

$$\mathbb{V}\mathrm{ar}\left[\widetilde{\mu}^K\right] = \mathbb{V}\mathrm{ar}\left[\mu^K | \widehat{\mu}^k\right] = \Sigma^{KK} - \Sigma^{Kk}\Sigma\Sigma^{kK}. \tag{3.4}$$

The parameters in Equations (3.3) and (3.4) such as β and those characterizing the GRF are unknown in practice. In practice, these unknown parameters are estimated based on simulation outputs, as detailed in Ankenman et al. (2010) and Salemi et al. (2013), a process known as *fitting the metamodel*. The output of the fitted stochastic kriging metamodel at the prediction points is given by Equation (3.3) with the estimated parameters plugged in. As discussed in Sections 3.3 and 3.5, the fitted metamodel is then used for solving the original task asked of the simulation model.

3.3. Green Stochastic Kriging for Repeated Monte Carlo Experiments

In this section we apply green simulation via metamodeling to improve the computational efficiency upon a standard Monte Carlo procedure. In particular, we propose reusing the simulation outputs of repeated Monte Carlo experiments in repeatedly fitting a stochastic kriging metamodel. As more and more simulation outputs are reused, we expect an improvement in the accuracy of the fitted metamodel, and as a result the efficiency of the simulation experiment improves over time. We examine the practical performance of the proposed GSK estimator in the catastrophe bond pricing example considered in Section 2.5.1. Our results show that this GSK estimator succeeds in reducing mean squared error (MSE) as more simulation output is reused.

Figure 3.2 illustrates a sequence of two repeated Monte Carlo experiments, whose simulation outputs are reused in a metamodel to produce an accurate answer for the second experiment. The data used for fitting the metamodel is shown in the dotted rectangle and the usage of fitted metamodel is highlighted by a dotted line.

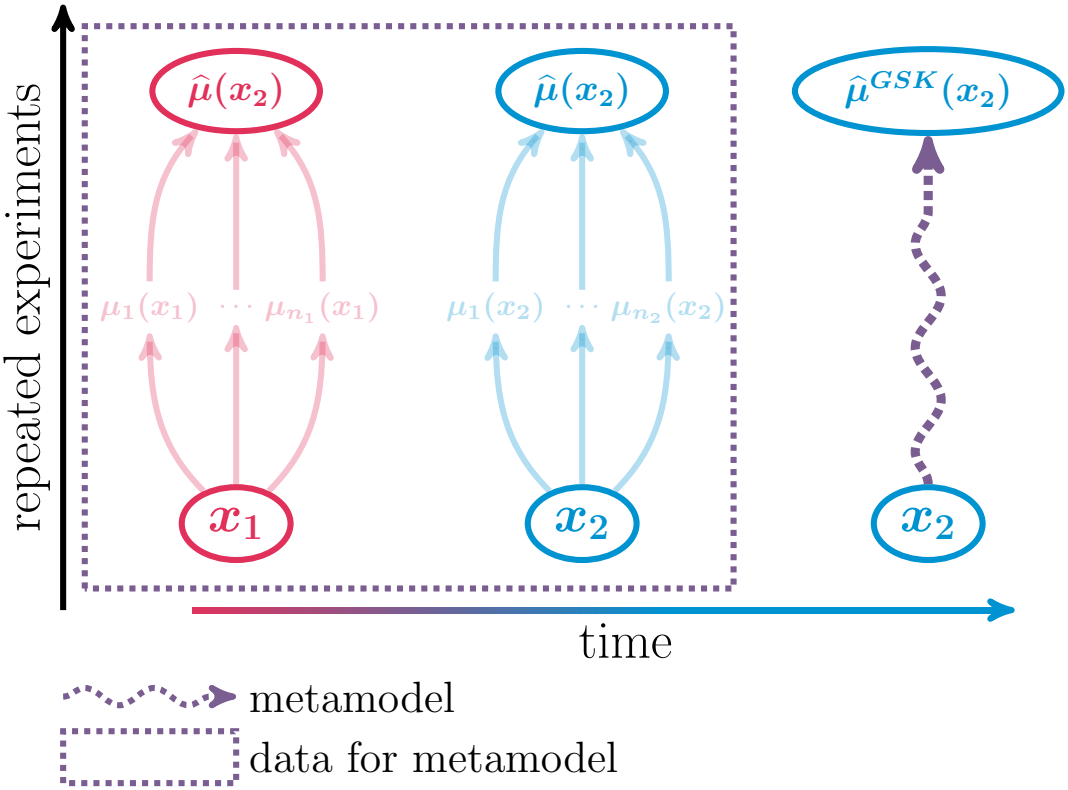


Figure 3.2. Green simulation via metamodeling for repeated Monte Carlo experiments.

As discussed above, the same response surface μ is considered in the repeated experiments with different values of inputs $x_1, x_2, \ldots$. As a result, reusing simulation outputs in a metamodel $\widetilde{\mu}$ can provide an accurate and efficient approximation of $\mu(x)$, even for an input value x for which no simulation has been run before. In addition, the quality of the metamodel is likely to improve over time, as more and more simulation output is reused. We propose and investigate a green stochastic kriging estimator in repeated Monte Carlo experiments in the following discussion.

3.3.1. Green Stochastic Kriging Estimator

Consider the setting of repeated experiments with a fixed number r of replications. Let $\widehat{\mu}_{n,r} = [\widehat{\mu}_r^{SMC}(x_1), \cdots, \widehat{\mu}_r^{SMC}(x_n)]^\top$ represent the vector of SMC estimators for the first n experiments. In the current (nth) experiment, we propose fitting a stochastic kriging metamodel based on $\widehat{\mu}_{n,r}$, and using the fitted model to estimate the current experiment's output. In particular, let x be the target state of the current experiment, based on Equation (3.3), we propose the following *green stochastic kriging (GSK)* estimator for $\mu(x)$

$$\widehat{\mu}_{n,r}^{GSK}(x) = f(x)^\top \widehat{\beta} + \widehat{\Sigma}(x)\widehat{\Sigma}^{-1}\left(\widehat{\mu}_{n,r} - [f(X_1)^\top \widehat{\beta}, \ldots, f(X_n)^\top \widehat{\beta}]^\top\right). \tag{3.5}$$

where $\widehat{\beta}$ is the trend model parameter vector, $\widehat{\Sigma}(x)$ is the $1 \times n$ vector of extrinsic covariances between $\mu(x)$ and $\mu(X_1), \ldots, \mu(X_n)$, and $\widehat{\Sigma}$ is $n \times n$ matrix of the sum of intrinsic and extrinsic covariances among $\mu(X_1), \ldots, \mu(X_n)$, all calculated based on fitted parameters of the stochastic kriging metamodel.

In our experiments, we used a linear model, $f(x) = x$; we found that the results were superior to using a constant model, $f(x) = 1$. In addition, we chose the GRF to be a generalized integrated Brownian field (Salemi et al., 2013) and found that the results were superior to the more common choice of GRF with a Gaussian correlation function.

For the implementation of the GSK estimator, we propose a green algorithm that reuses previously fitted parameters to warm-start the fitting of parameters in the current experiment. After the nth experiment, SK performs an optimization to find a vector of parameters that make the GRF fit best to the inputs and outputs of the first n experiments. We set the initial value in this optimization equal to the parameter vector chosen after the

$(n-1)$st experiment. In our numerical experiments, we found that this warm-starting significantly improved the speed and numerical stability of SK parameter-fitting.

3.3.2. Comparison between the GSK and LR estimators

In general, the GSK estimator is more broadly applicable than the LR estimators. Specifically, the GSK estimator does not require Assumptions (A.1)– (A.6), without which the LR estimators may be infeasible or have poor practical performance. Some of the assumptions are stringent or hard to verify in practical applications, and should be viewed as limitations of the LR method. In contrast, Assumption (B1) for the GSK estimator can usually be justified by the Central Limit Theorem when the number of replications is reasonable. Even if this assumption fails, the GSK estimator may be suboptimal but can still be implemented nevertheless. Moreover, unlike Assumption (A.1) for the LR estimator, the input state x for the GSK estimator can affect the the simulation experiment through the likelihood of the random vector, the parameters in the simulation logic, or both.

Consider a case where the GSK and the three LR estimators proposed in Chapter 2 are all applicable, i.e., Assumptions (A.1)– (A.6) and (B1) hold. By substituting Equation (2.2) into Equation (3.5), the GSK estimator can be seen as including a weighted sum of the simulation outputs $\left\{F\left(Y_k^{(j)}\right) : k = 1, \ldots, n; j = 1, \ldots, r\right\}$. The LR estimators are also weighted sums of the simulation outputs. However, the GSK weights behave quite differently from the LR weights. The LR weights depend on the likelihoods associated with the states $X_1, \ldots, X_n$, and they cannot be negative. Like weights in stochastic kriging in general, the GSK weights depend primarily on the locations of the states $X_1, \ldots, X_n$, as related by the extrinsic covariance function, and they can be negative.

From an implementation standpoint, the GSK estimator requires far less storage and is usually less computationally demanding than the LR estimators. As indicated in Figure 3.2, in the nth experiment the GSK estimator requires storage for the simulation input-output pairs $\left\{\left(x_k, \widehat{\mu}_r^{SMC}(x_k)\right) : k = 1, \ldots, n\right\}$ and the previously fitted parameter values for the warm-starting implementation. In contrast, all LR estimators require storing all the replications $\left\{\left(Y_k^{(j)}, F\left(Y_k^{(j)}\right)\right) : k = 1, \ldots, n; j = 1, \ldots, r\right\}$ for computing likelihoods and for reusing their values, not to mention storing density evaluations for green implementations. Depending on the number of replications in each experiment, the storage for the LR estimators may be substantially greater than that for the GSK estimator. The two most demanding computations in the GSK estimator are the optimization in the fitting process and the matrix inversion in the estimation step. These computations are not excessive for reasonable number of unknown parameters and moderate number of repeated experiments. In our experiments, in all 100 repeated experiments the GSK estimator is much faster than all the LR estimators.

3.3.3. Numerical Experiments

In this section, we use two numerical examples to examine the practical performance of the GSK estimator and compare it with the LR estimators. The two examples are identical to those in Section 2.5: catastrophe bond pricing and credit risk assessment of a loan portfolio. For brevity we do not repeat the example descriptions and focus only on the new results of GSK estimators. In the catastrophe bond pricing example, the GSK estimator succeeded in reducing mean squared error (MSE) as more simulation outputs are reused, but its MSE was larger than those of the LR estimators. In the credit risk example the

GSK estimator is was not successful: its MSE is larger than the SMC estimator. This failure indicates certain limitations of the GSK estimator and inspires potential future research.

To investigate the effectiveness of the GSK estimator, we performed similar experiments as those in Section 2.5:

- For the catastrophe bond pricing example, we performed 10,000 macro replications of sequence of $n = 100$ repeated experiments, each with $r = 100$ replications.
- For the credit risk assessment example, we performed 10,000 macro replications of sequence of $n = 52$ repeated experiments, each with $r = 1,000$ replications.

In each sequence of repeated experiments, we start evaluating the GSK estimator from $n = 5$ to avoid over-fitting in metamodeling. Similar to the experiments for the LR estimators, the estimated MSE of a fixed-state estimator $\widehat{\mu}(x)$ of $\mu(x)$ was $\sum_{k=1}^{10,000} (\widehat{\mu}^{(k)}(x) - \mu(x))^2/10,000$, where $\widehat{\mu}^{(k)}(x)$ is the value of the estimator on the kth macro-replication. Likewise, the estimated MSE of a current-state estimator $\widehat{\mu}(X_n)$ of $\mu(X_n)$ was $\sum_{k=1}^{10,000} (\widehat{\mu}^{(k)}(X_n^{(k)}) - \mu(X_n^{(k)}))^2/10,000$. Due to using 10,000 macro-replications, the standard errors of these estimated MSEs are less than 1% of the corresponding estimated MSE. Recall that MSE equals variance for the SMC and LR estimators, which are unbiased.

Figures 3.3 and 3.4 are log-log plots of the estimated MSEs for the SMC, ILR, EWLR, MLR, and GSK estimators for two fixed states and for the current state. They are otherwise identical to Figures 2.2 and 2.3 except the dotted line, showing MSE for the GSK estimators.

We see from both Figures 3.3 and 3.4 that the GSK MSE decreases as n increases. For the center state, after 10 experiments the GSK MSE is comparable to the ILR variance.

After 100 experiments, the GSK estimator's variance is over 36 times smaller than the SMC estimator's variance for x_{mi} and about 1.6 times smaller for x_{hi}. Because x_{hi} is always outside the range of $X_1, \ldots, X_n$, SK must extrapolate when it estimates $\mu(x_{hi})$. Extrapolation is challenging for SK, so it is a success for GSK that, for the extreme state, its MSE becomes lower than the SMC variance after about 40 experiments, and continues to decrease. While the MSE of the GSK estimator is larger than that of the LR estimators in this example, the decreasing MSE of the GSK estimator nevertheless illustrates the benefit of green simulation via metamodeling.

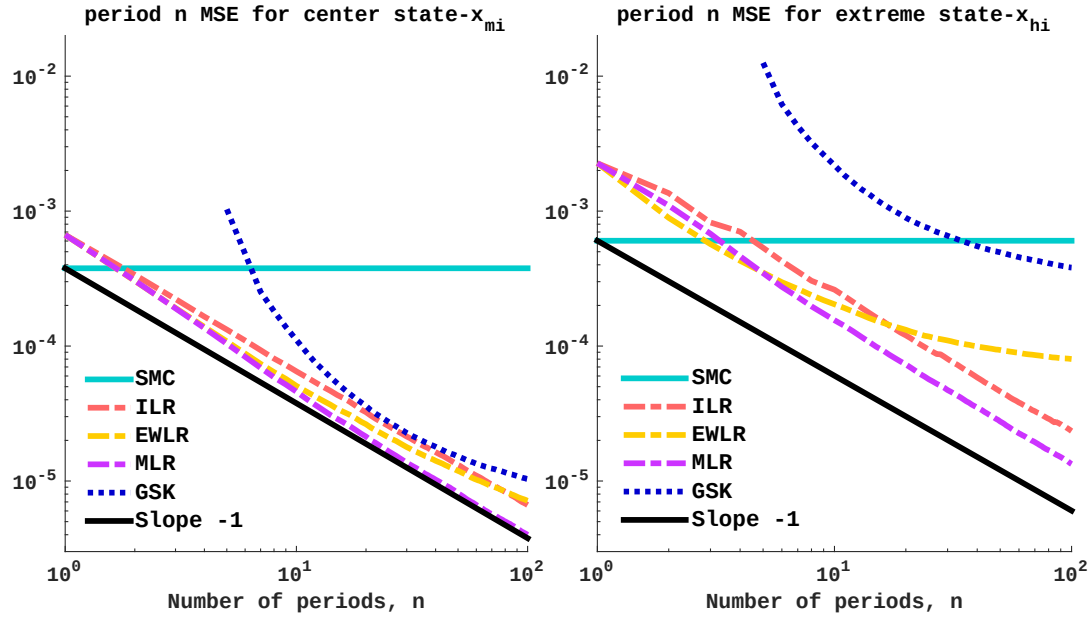


Figure 3.3. Log-log plots of estimated MSEs of fixed-state SMC, LR, and GSK estimators for CAT bond pricing example.

The catastrophe bond pricing example has provably amenable properties for the LR methods, such as the ergodicity of the underlying input state process, which may partially contribute to the decreasing GSK MSE too. In contrast, as discussed in Section 2.5.2 the credit risk example has a non-ergodic state process and poses difficulties to some of the

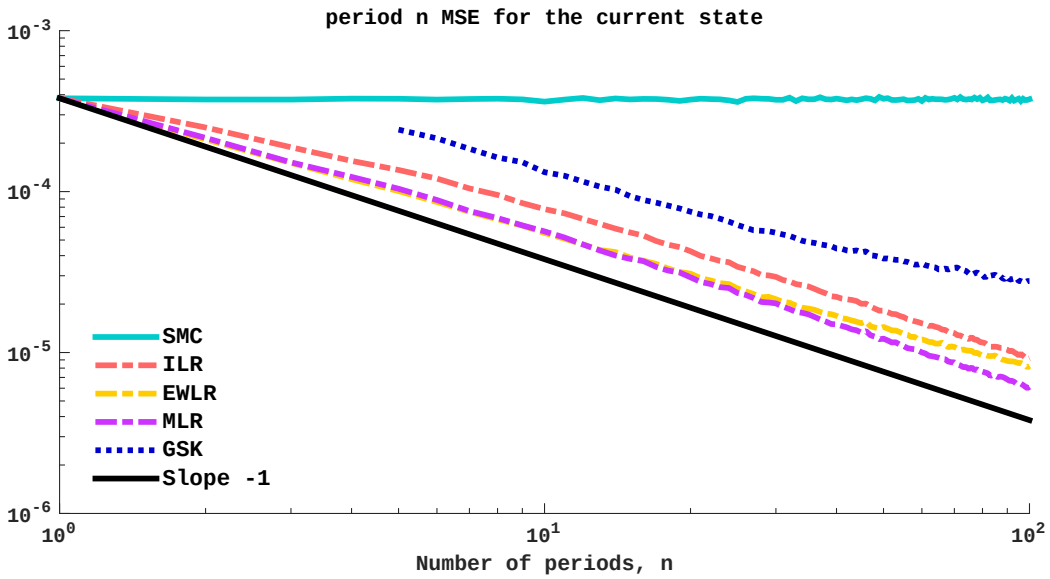


Figure 3.4. Log-log plots of estimated MSEs of current-state SMC, LR, and GSK estimators for CAT bond pricing example.

LR estimators. The MSE of the GSK estimator and its 95% error bars in the credit risk example is depicted in Figure 3.5. We see that the GSK estimator performed poorly in this example. The reason for the poor performance of GSK in this example is as follows. In this example, there is a strong correlation between the conditional expectation $\mu(X_n)$ and conditional variance $\sigma^2(X_n)$ of the simulation output. Furthermore, $\sigma^2(X_n)$ can be very small and difficult to estimate, so that the estimated conditional variance $\widehat{\sigma}^2(X_n)$ can even be zero. In stochastic kriging, simulation output with lower estimated variance receives higher weight; if lower estimated variance is correlated with lower estimated mean, bias can result (Staum, 2009). This stochastic kriging bias is large in this example, causing the poor performance of GSK.

The poor performance of GSK in this example reveals a drawback of green simulation via metamodeling: the need for *model validation*. This suggests that, when developing methods

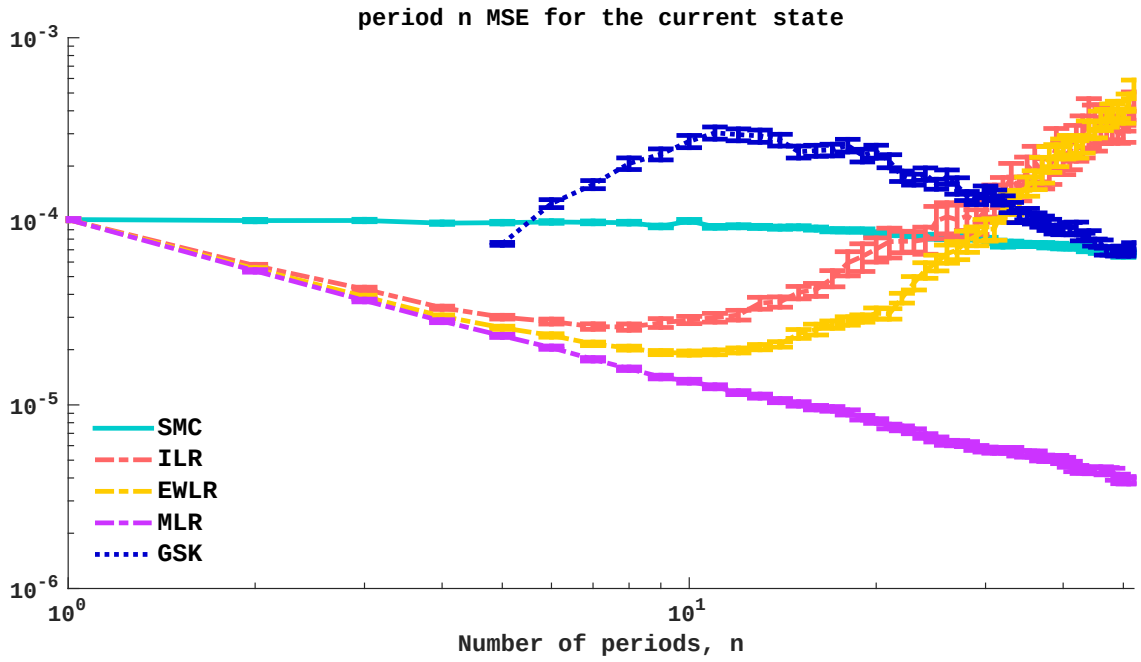


Figure 3.5. Log-log plot with error bars for estimated variances of the SMC, LR, and GSK estimators for current state estimators for credit risk example.

for green simulation via metamodeling, one cannot assume validity of the metamodel simply because of the increasing amount of reused simulation outputs. When assessing the effectiveness of the green metamodeling method, one should balance the improved efficiency due to reusing simulation outputs and the efforts for model validation.

3.4. Green Nested Simulation of Expected Shortfall

In this section we develop a green nested simulation (GNS) procedure for a repeated risk assessment problem. The work of Liu et al. (2010) and Broadie et al. (2011) both use metamodels in nested simulation, but for isolated experiments. The GNS procedure considers repeated experiments and reuses outputs of a nested simulation procedure, which is more complicated than standard Monte Carlo experiment where the LR and

GSK estimators were employed. In addition, unlike the LR and GSK estimators, which reuse simulation outputs without changing the design of individual repeated experiments, the GNS procedure has an innovative experiment design for reusing previous simulation outputs. In short, the GNS procedure showcases the generality and the potential of green simulation for different applications.

The GNS procedure is based on that of Liu and Staum (2010), which we refer to as the *baseline procedure*. In particular, the baseline procedure solves the risk assessment problem only once while the GNS procedure is suitable for the case where the risk assessment experiment is repeated. As shown in Liu and Staum (2010), the baseline procedure can be 50 times more accurate, as measured by root mean squared error (RMSE), than a standard nested simulation procedure with the same simulation budget. Our numerical study shows that, after reusing the simulation output of 15 experiments, the GNS estimator is 10 times more accurate than that of the baseline procedure, which does not reuse any simulation output.

3.4.1. Motivating Example and Baseline Procedure

In this section we summarize the motivating example and the baseline procedure, based on which our GNS procedure is developed.

The motivating example is an extension of that in Liu and Staum (2010). We consider a portfolio risk assessment problem that is repeated multiple times, each with updated input values. The portfolio in question is a portfolio of call options on the stocks Cisco (CSCO) and of Sun Microsystems (JAVA) as shown in Table 3.1. In the table, the position is expressed as the number of shares of stock the option owner is entitled to buy, where a

negative position means a short position in the call option. We refer to Liu et al. (2010) for the source of data.

Underlying Stock	Position	Strike	Maturity	Price	Interest Rate	Implied Volatility
CSCO	200	\$27.5	0.315	\$1.65	4.82%	26.66%
CSCO	-400	\$30	0.315	\$0.7	4.82%	25.64%
CSCO	200	\$27.5	0.564	\$2.5	5.01%	28.36%
CSCO	-200	\$30	0.564	\$1.4	5.01%	26.91%
JAVA	900	\$5	0.315	\$0.435	4.82%	35.19%
JAVA	1200	\$6	0.315	\$0.125	4.82%	35.67%
JAVA	-900	\$5	0.564	\$0.615	5.01%	36.42%
JAVA	-500	\$6	0.564	\$0.26	5.01%	35.94%

Table 3.1. Portfolio of call options. The prices for CSCO and JAVA are \$27.15 and \$5.01 in day 1.

The goal of each risk assessment experiment is to estimate the one-day expected shortfall (ES) of the portfolio. A two-level nested simulation experiment is illustrated in Figure 3.6: Suppose the experiment is launched at time n to facilitate later discussions on

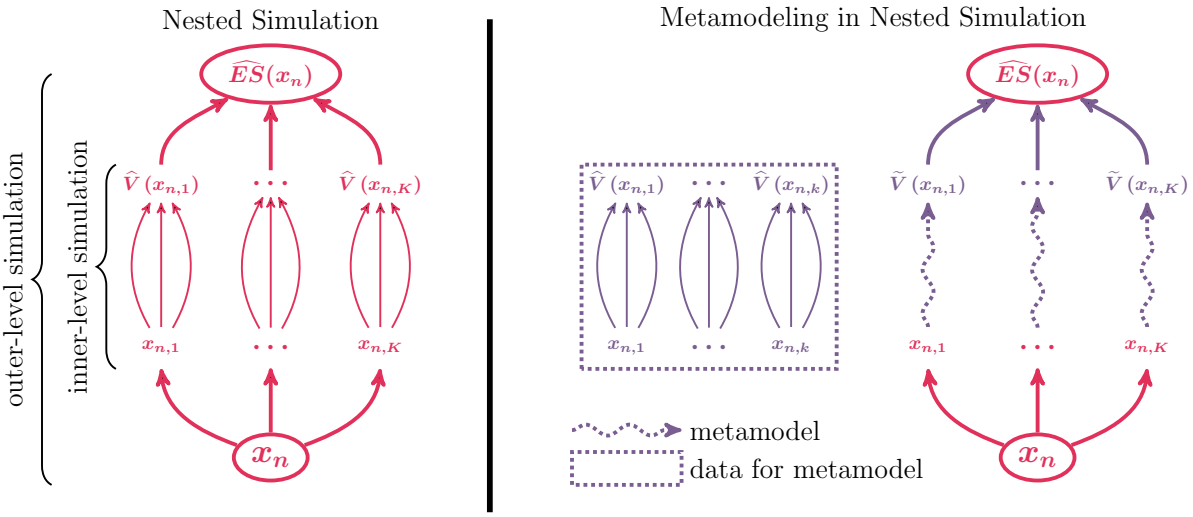


Figure 3.6. Left panel: Standard two-level nested simulation for estimating expected shortfall. Right panel: Application of metamodeling for the inner-level model of a two-level nested simulation.

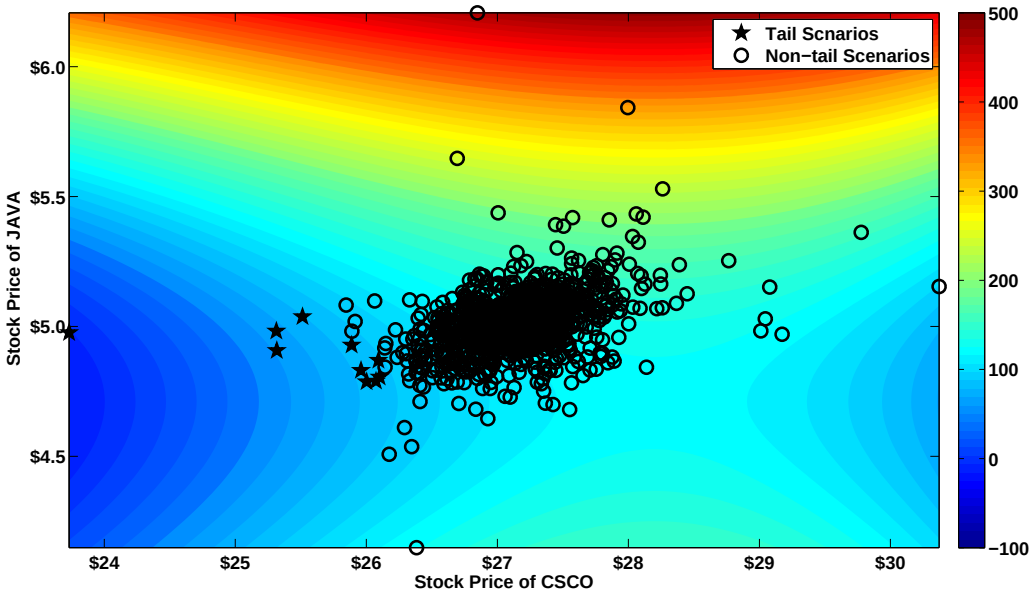
repeated experiments. Given the current (nth) input x_n, which encodes the current market prices for CSCO and JAVA, the outer-level simulation generates K possible *scenarios* $x_{n,i}$ for $i = 1, \cdots, K$. Each scenario is a pair of next-day stock prices, which may be based on historical stock return observations or Monte Carlo sampling from a stochastic stock price model. In our experiments, the outer-level simulation randomly samples 1000 historical return observations for CSCO and JAVA. The inner-level simulation conditional on scenario $x_{n,i}$ yields an estimate of the next-day portfolio value, $V(x_{n,i})$, in that scenario. The current portfolio value V_n is calculated by evaluating the options in the portfolio using Black-Scholes option pricing formula. The one-day profit and loss (P&L) in scenario $x_{n,i}$ is given by $V_n - V(x_{n,i})$. Given the set of K equally probable scenarios $x_{n,1}, \ldots, x_{n,K}$ and their respective P&L's, the one-day expected shortfall is the average of the largest losses in these scenarios. Specifically, suppose we are interested in a tail of probability α (e.g. $\alpha = 95\%$), where $K(1-\alpha)$ is assumed to be an integer. Then one-day α-ES in the at the current experiment is given by

$$ES_\alpha(x_n) = \frac{1}{K(1-\alpha)} \sum_{i=1}^{K(1-\alpha)} \left[V_n - V(x_{n,(i)})\right] \tag{3.6}$$

where $V(x_{n,(i)})$ is the ith smallest portfolio value among those in the K scenarios. For ease of discussions we refer to the scenarios with $K(1-\alpha)$ smallest portfolio values, i.e., $x_{n,(i)}$ for $i = 1, \ldots, K(1-\alpha)$, as the *tail scenarios* and the other scenarios as the *non-tail scenarios*.

Based on Liu and Staum (2010), the true model for the next-day portfolio value, i.e., the inner-level simulation model, is the sum of its option's values based on the Black-Scholes option pricing formula. As pointed out by Liu and Staum (2010), inner-level simulation is

not necessary in this example because the next-day's portfolio value is a known function of the scenario. For example, the contour plot of next-day's value, as a function of the stock prices for CSCO and JAVA, for the portfolio specified in Table 3.1 is depicted in Figure 3.7. Also, 1,000 scenarios based on historical observations of one-day stock returns are shown in Figure 3.7, where 10 tail scenarios (i.e., $\alpha = 99\%$) are specified as solid stars. We study this simple example because we are able to evaluate the accuracy of the



22

Figure 3.7. Contour plot of portfolio value as a function of scenario. Stars and circles are scenarios generated from historical data.

ES estimators of different procedures. However, both the baseline procedure proposed by Liu and Staum (2010) and our GNS procedure are designed to tackle problems where inner-level simulation is necessary. Therefore, instead of using the closed-form formula to evaluate options in the portfolio at each scenario, inner-level simulations (geometric Brownian motions for stock prices) are used to estimate the option prices. Note that the

Black-Scholes formula is nevertheless used to evaluate the current portfolio value V_n. This is because in practice the current portfolio value is often observed from the market thus no simulation is required.

We consider the case where the experiment above is performed repeatedly at times $n = 1, 2, \ldots$, each with observed market information x_n at time n. The experiments are repeated in the sense that, while the input values x_n and the scenarios $x_{n,i}$ may be different, the inner- and outer-level simulation models remain the same. Since we perform the same experiment for the same portfolio (specified in Table 3.1), the portfolio value $V(x)$ as a function of the scenario x is the same for all experiments. So we can view V as the unknown response surface and use metamodeling to approximate it. This viewpoint is illustrated in the right panel of Figure 3.6. We refer to the scenarios $x_{n,1}, \ldots, x_{n,k}$ where inner-level simulations are run to provide data for metamodeling as the *design points*. The scenarios $x_{n,1}, \ldots, x_{n,K}$ whose portfolio value is approximated by the calibrated metamodel are referred to as the *prediction points*. Let $\widetilde{V}(x_{n,i})$ be the metamodel output at prediction point $x_{n,i}$. The estimator for the α-ES in the baseline procedure and our GNS procedure are both given by

$$\widehat{ES}_\alpha(x_n) = \frac{1}{K(1-\alpha)} \sum_{i=1}^{K(1-\alpha)} \left[V_n - \widetilde{V}(x_{n,(i)}) \right] \tag{3.7}$$

where $\widetilde{V}(x_{n,(i)})$ is the ith lowest value among the stochastic kriging predictions at the predictions points.

We now motivate and summarize the baseline nested simulation procedure for the above risk assessment problem based on Liu and Staum (2010), which solves risk assessment problem only once and solves it efficiently. Given fixed simulation budget for

each experiment, say C inner-level replications, standard practice for nested simulation experiment is to allocate equal number of inner-level replications to all K scenarios, which is not a desirable allocation scheme for estimating ES. We see from Equation (3.6) that only the tail scenarios matter in estimating ES. Moreover, for common values of α (e.g. $\alpha = 95\%$ or 99%) there are many fewer tail scenarios than non-tail scenarios. So the standard equal allocation to all scenarios is inefficient because the majority of the simulation budget is wasted on non-tail scenarios. In this case, it is desirable to allocate more simulation budget to the tail scenarios than to the non-tail scenarios, provided that one can cheaply identify these scenarios.

The following summary of the baseline procedure is based on Liu et al. (2010), which is designed to solve a one-day ES risk assessment problem only once and solve it efficiently. The baseline procedure has three stages, each of which constructs or refines a stochastic kriging metamodel. An outline of the baseline procedure is given in Figure 3.8. While we only make remarks on some key components that are relevant to our development of the GNS procedure, readers are referred to Liu and Staum (2010) for details of the baseline procedure.

In the following discussion, we refer to figures that illustrate the performance of the baseline procedure. These figures are based on one run of the baseline procedure to estimate the 99%-ES of the scenarios randomly sampled from historical stock returns. The baseline procedure uses a simulation budget C of 0.5 million replications, $K = 1000$ prediction points, a target of $k_1 = 50$ Stage I design points and $k_2 = 30$ Stage II design points, $n_0 = 2000$ replications per design point in Stages I and II, and sampling $M = 300$ times from the posterior distribution of portfolio value at the design points. This choice of

Parameters: Simulation budget C, number of prediction points K, number of stage 1 design points k_1, number of stage 2 design points k_2, initial number of replications per design n_0, number of stage 2 posterior samples M. Observed market information x_n, current portfolio value V_n is calculated using Black-Scholes or observed from the market.

Stage I:

I.1. Generate K prediction points by outer-level simulation (historical data or Monte Carlo).

I.2. Generate roughly k_1 Stage I design points, using Latin hypercube sampling, within the convex hull of prediction points.

I.3. Simulate n_0 replications for every Stage I design point. Create a stochastic kriging metamodel based on the simulation outputs.

Stage II

II.1. Sample a vector of portfolio values for all prediction points from the multivariate normal distribution with mean and variance suggested by the Stage I metamodel. Based on M such samples, select at most k_2 prediction points that seem likeliest to be tail scenarios, and add them to the set of design points.

II.2. Simulate n_0 replications for every *new* Stage II design point. Create a stochastic kriging metamodel based on the simulation outputs.

Stage III

III.1. Based on the total simulation budget C and the simulations performed in the first two stages, allocate the remaining computational budget to all design points. The allocation scheme is found by minimizing the posterior variance of an ES estimator based on the likelihood of each design point being a tail scenario.

III.2. Perform further simulation at the design points based on the above allocation scheme. Create a stochastic kriging metamodel based on the simulation outputs.

III.3. Compute the ES estimator in Equation (3.7) using the final metamodel.

Figure 3.8. Baseline procedure for efficient nested simulation of one-period ES.

parameters is consistent with those in Liu and Staum (2010) when the simulation budget is small, in which case the benefit of green designs is apparent.

Kleijnen and Van Beers (2004) argue that kriging should not be used for extrapolation. So Stage I of the baseline procedure avoids extrapolation of stochastic kriging by selecting design points that lie on the boundary of convex hull of the prediction points. In addition, space-filling design points are also used to ensure an accurate metamodel in the interior of the convex hull. Figure 3.9 shows the Stage I design points chosen for the baseline procedure and the absolute error $|\widetilde{V} - V|$ of the Stage I metamodel. The error could be substantial in many regions in the design space in Stage I, but we shall see the error shrinks in the neighborhoods of the tail scenarios in later stages of the baseline procedure.

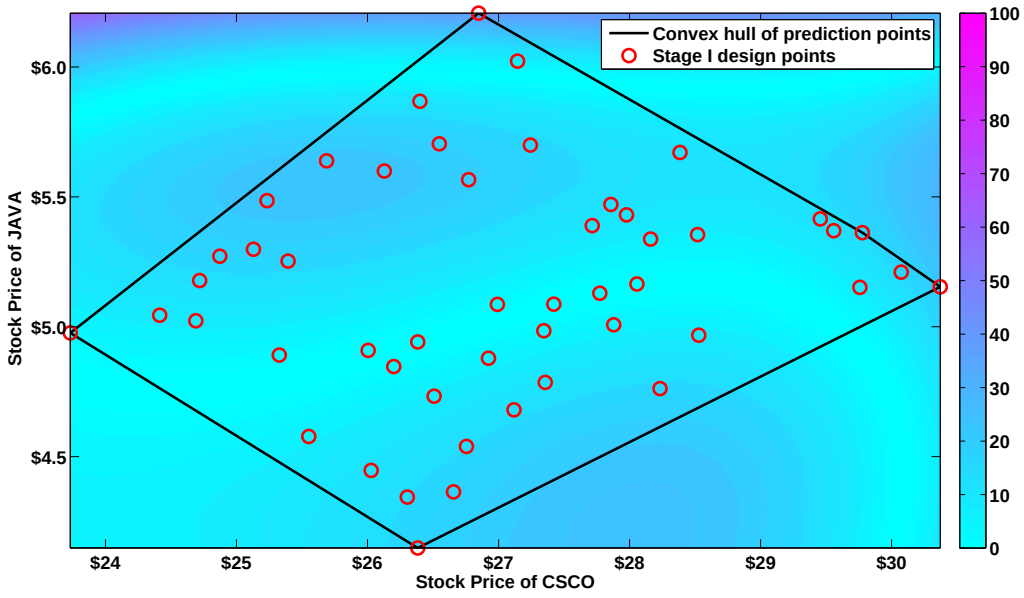


Figure 3.9. Absolute error of the Stage I metamodel on one run of the baseline procedure.

Step II.1 tries to guess which scenarios belong to the tail. Based on the Stage I stochastic kriging metamodel and in particular the posterior distribution of the portfolio values, M random samples of portfolio values can be generated $\widetilde{V}^{(1)}, \ldots, \widetilde{V}^{(M)}$. Let $T_i^{(j)}$ be

an indicator function that equals one if $\widetilde{V}_i^{(j)}$ is among the $K(1-\alpha)$ lowest components of $\widetilde{V}^{(j)}$. Based on the M samples, the likelihood of scenario i being a tail scenario is estimated by $\hat{q}_i := \sum_{j=1}^{M} T_{i(j)}/M$. Stage II design points are scenarios with highest non-zero estimated likelihoods. It is suggested that $k_2 > K(1-\alpha)$ to improve the chance of selecting the true tail scenarios.

Figure 3.10 shows the new Stage II design points and the absolute error of Stage II metamodel. Although we selected $k_2 = 30$, on this particular run only 27 design points were added in Stage II: the other scenarios' values were never among the 10 lowest in $M = 300$ samples. In this particular run, all 10 tail scenarios were selected as Stage II design points, which is a success for the baseline procedure. Although the Stage II metamodel worsens compared to Stage I metamodel in some regions, it has much smaller absolute error in the region around potential tail scenarios.

Given likelihood estimates $\hat{q}_i$, a reasonable estimator of ES is

$$\boldsymbol{w}^\top(\Upsilon^* \mathbf{1}^K - \mathbf{Y}^K), \text{ where } w_i = \frac{\hat{q}_i}{K(1-\alpha)} \text{ for } i = 1, \ldots, K. \tag{3.8}$$

The allocation of remaining simulation budget to design points is determined by minimizing the posterior variance of the ES estimator in Equation (3.8). Liu and Staum (2010) show the following optimization problem solves a simplified version of the desired variance minimization problem

$$\min_{n_1,\ldots,n_k} \left\{ \boldsymbol{U}^\top \boldsymbol{V} \boldsymbol{N}^{-1} \boldsymbol{U} \,\middle|\, \sum_{i=1}^{k} n_i = C, n_i \geq n_0, \forall i = 1, \ldots, k \right\} \tag{3.9}$$

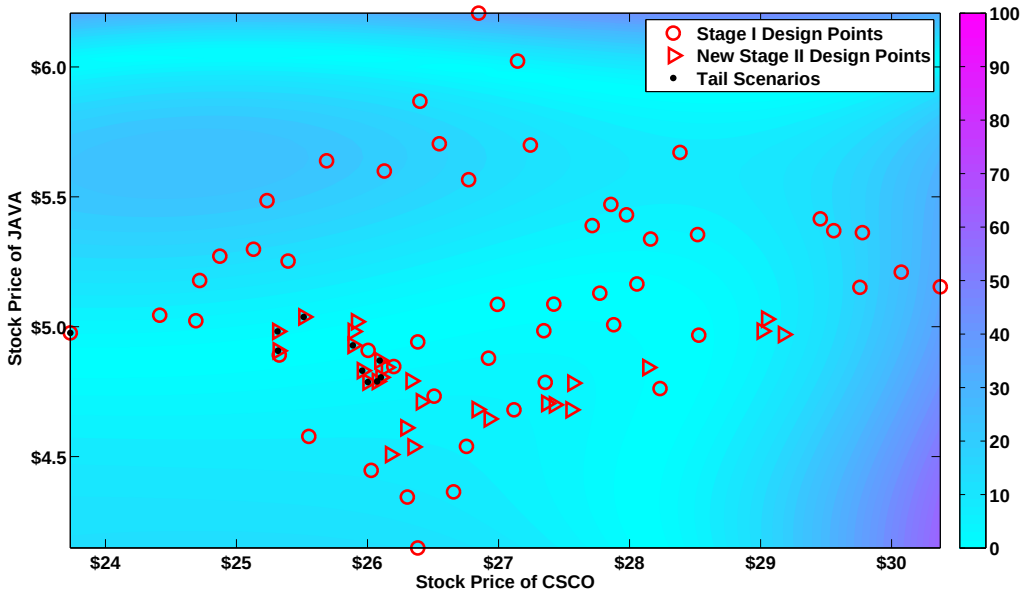


Figure 3.10. Absolute error of the Stage II metamodel on one run of the baseline procedure.

where $\boldsymbol{U} := (\boldsymbol{\Sigma}^{kk} + \boldsymbol{V}/n_0)^{-1}\boldsymbol{w}$ and the covariance matrices come from the Stage II metamodel. Problem (3.9) can be solved efficiently (Bitran and Hax (1981); Bretthauer et al. (1999)).

Figure 3.11 shows the allocation of simulation budget of the baseline procedure. The simulation budget is spent mostly on design points that are tail scenarios or in their neighborhoods. Simulation efforts near tail scenarios can improve the inference by stochastic kriging metamodel about the portfolio values in tail scenarios. Figure 3.12 shows the absolute error of the Stage III metamodel. Due to significant simulation efforts devoted to regions around tail scenarios, the error in estimation portfolio value of the tail scenario decreases dramatically from Stage II to Stage III. The error is large in some regions that are far from the tail scenarios, which does not affect the accuracy of estimating ES.

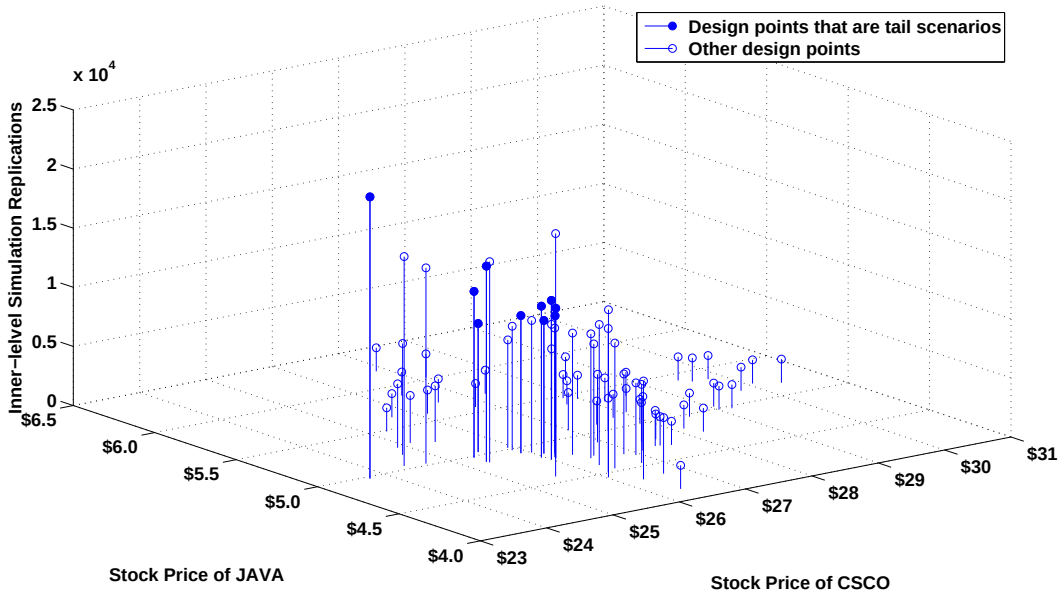


Figure 3.11. Number of simulation replications allocated to each design point on one run of the baseline procedure.

As shown in the numerical study in Liu and Staum (2010), the the baseline procedure attains a root mean squared error (RMSE) "dozens of times smaller than a standard simulation procedure." In the next section, we develop a green nested simulation procedure that can be significantly more accurate than the baseline procedure by reusing simulation outputs.

3.5. Green Nested Simulation Procedure of Expected Shortfall

In this section we present a nested green simulation (GNS) procedure for the repeated risk assessment problem. Note that the baseline procedure in Figure 3.8 always starts "from scratch", whereas our GNS procedure incorporates preexisting simulation outputs

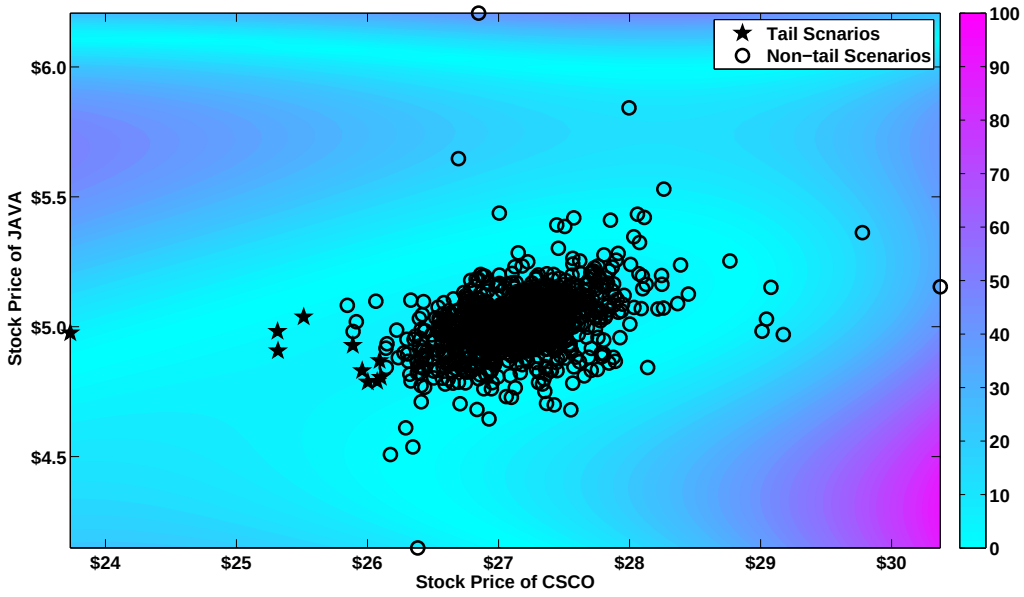


Figure 3.12. Absolute error of the Stage III metamodel on one run of the baseline procedure.

from previous experiments. As the amount of preexisting simulation outputs increase, the GNS procedure attains higher accuracy in estimating ES.

Given the success of the baseline procedure, when the risk assessment experiment is run repeatedly with different inputs, it may be a reasonable approach to simply apply the baseline procedure repeatedly, as outlined in Figure 3.13. By merely repeating the baseline procedure, which starts from scratch every time, we expect the same accuracy from its estimator regardless of how many times the experiment is repeated.

We propose a green nested simulation procedure outlined in Figure 3.14. Our procedure is similar to the baseline procedure: a three-stage procedure with similar or even identical steps in each stage. In particular, Figure 3.14 differs from Figure 3.13 only in Stage I: reuses previous simulation outputs as the starting point of the three-stage baseline

Parameters: C, K, k_1, k_2, n_0, M
Initialization: Observe the current market information x_n.
Stage I. Implement Stage I in Figure 3.8.
Stage II. Implement Stage II in Figure 3.8.
Stage III. Implement Stage III in Figure 3.8.

Figure 3.13. Repetition of baseline procedure to estimate ES in the nth experiment.

procedure. The simulation effort in Stage I of the baseline procedure is greatly reduced and the remaining, unused budget will be allocated to Stage III where the accuracy of the ES will be improved. In contrast, our green simulation procedure recycles all simulation outputs available. As time progresses more experiments are repeated so the amount of preexisting simulation outputs accumulates, we expect the accuracy of the resulting green ES estimator to improve.

Our green procedure in Figure 3.14 differs from Figure 3.13 in only two steps: step I.2 has an extra augmentation design and an extra recycling step after the experiment is done, both are ideas based on the general green simulation paradigm. The augmentation step is designed in a way to fully utilize all preexisting outputs but still avoid extrapolation of the resulting Stage I metamodeling. If without the recycling step, i.e., no preexisting simulation output is available, the three main stages of our procedure are identical to those in Figure 3.13. In the following discussion, we view the previous Figures 3.9- 3.12 as the first of a sequence of repeated experiments, whose outputs are recycled and reused in the future.

We argue that, in complicated experiments such as nested simulations, it is important to have thoughtful designs for green simulation or reusing simulation outputs may not be beneficial or effective. Figure 3.15 depicts a sequence of two repeated experiments: The

Parameters: C, K, k_1, k_2, n_0, M
Initialization: Observe the current market information x_n.
Stage I.

I.1. Implement I.1. in Figure 3.8

I.2. ***If no previous simulation output exists***, implement I.2. in Figure 3.8. ***Otherwise, if there is simulation output at some previous design points***, consider the convex hull of both those design points and the current prediction points. Add prediction points that lie on the boundary of this convex hull as new Stage I design points.

I.3. Simulate n_0 replication for every *new* Stage I design point. Augment the preexisting stochastic kriging metamodel with the new simulation output.

Stage II. Implement Stage II in Figure 3.8.
Stage III. Implement Stage III in Figure 3.8.
Recycling Simulation Outputs: Recycle all Stage III design points as well as their sample mean, variance, and number of replications. Also recycle the Stage III stochastic kriging model. These are the preexisting simulation outputs to be reused by future experiments.

Figure 3.14. Green simulation procedure to estimate ES for T periods..

heat map in Figure 3.15 shows the absolute error for the Stage III metamodel from the first experiment, the solid boxed area is the convex hull of prediction points of the first experiment, the arrow shows a possible change of the input values from the first to the second experiment, and the circles and stars shows the prediction points for the second experiment. We see that the Stage III metamodel from the first experiment has substantial absolute error outside the convex hull of its prediction points, which is reasonable because stochastic kriging metamodel is ineffective for extrapolation. In the second experiment, 9 out 10 tail scenarios lie outside the convex hull of the first experiment's prediction points. If one directly proceeded to Stage II with the previous Stage III metamodel, none of these 9 tail scenarios would be identified as potential tail scenarios. Then Stage III will mistakenly allocates significant simulation budget around non-tail scenarios. The resulting

ES estimator will be worse than simply repeating the baseline procedure in the second experiment.

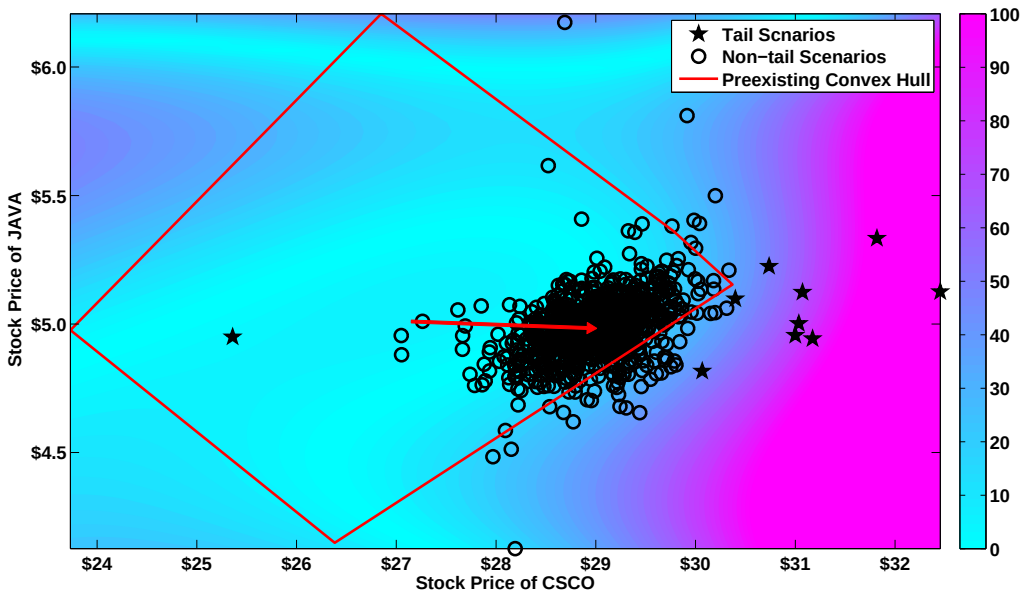


Figure 3.15. Absolute error of Stage III metamodel in the first experiment and 1,000 prediction points in the second experiment. Convex hull of the first experiment's prediction points is shown in the diamond-shape region. Change of input values is shown as a solid arrow.

The aforementioned weakness of the above simplistic approach suggests the importance of augmenting Stage I in the baseline procedure rather than skipping it entirely. In particular, based on the design principle of the baseline procedure, we propose an possible augmentation in the GNS procedure, i.e., Step I.2 in Figure 3.14, so as to avoid extrapolation of Stage I stochastic kriging metamodel. Specifically, in the second experiment, the Stage I metamodel is an augmentation or refinement of the Stage III metamodel in the first experiment: Consider the convex hull of previous design points and the current prediction

points. Prediction points that lie on this convex hull are added as new Stage I design points and used to augment the previous Stage III stochastic kriging metamodel. As a result, no prediction point requires extrapolation of this augmented metamodel.

Figure 3.16 shows that the augmentation step successfully handles this "adversarial" example. The absolute error $|\widetilde{V} - V|$ of the augmented Stage I metamodel is reasonable for *all* prediction points in the second experiment. Comparing Figure 3.15 and Figure 3.16 we see drastic differences in the absolute error around the tail scenarios. More importantly, this improvement is achieved by adding only 4 new Stage I design points, which, based on our choice of design parameters, is only a fraction of that in the baseline procedure ($k_1 = 50$) and requires less than 2% of the per-experiment simulation budget.

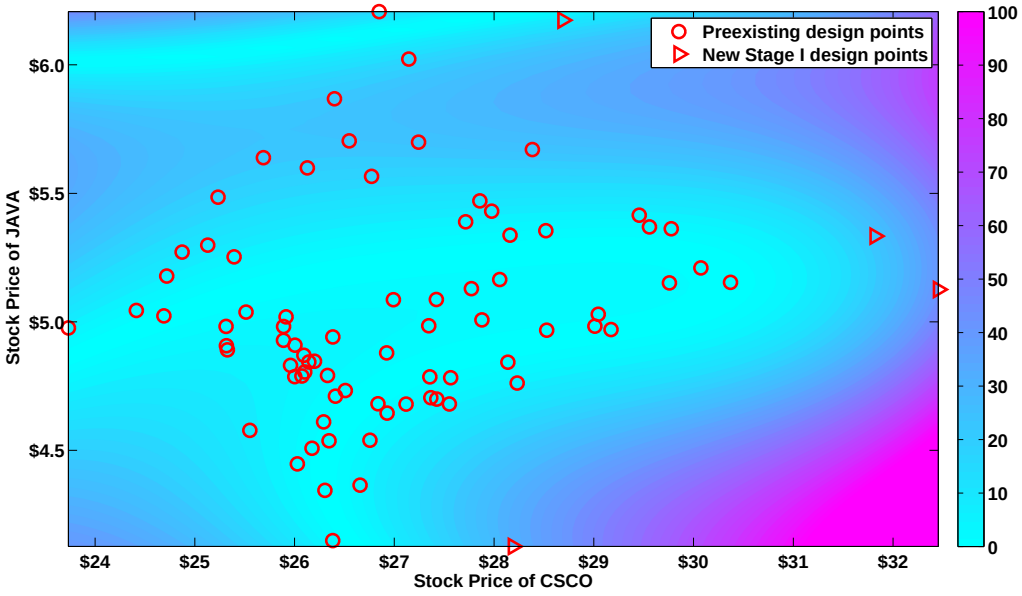


Figure 3.16. Absolute error of the augmented Stage I metamodel in the second experiment.

Figure 3.17 shows the new Stage II design points and the absolute error $|\widetilde{V} - V|$ of Stage II metamodel in the second experiment. The improvement over the augmented Stage I metamodel is not visually significant, which is a success of our augmentation design. Moreover, all 10 tail scenarios are included as design points in the Stage II metamodel, which is precisely the goal of Stage II.

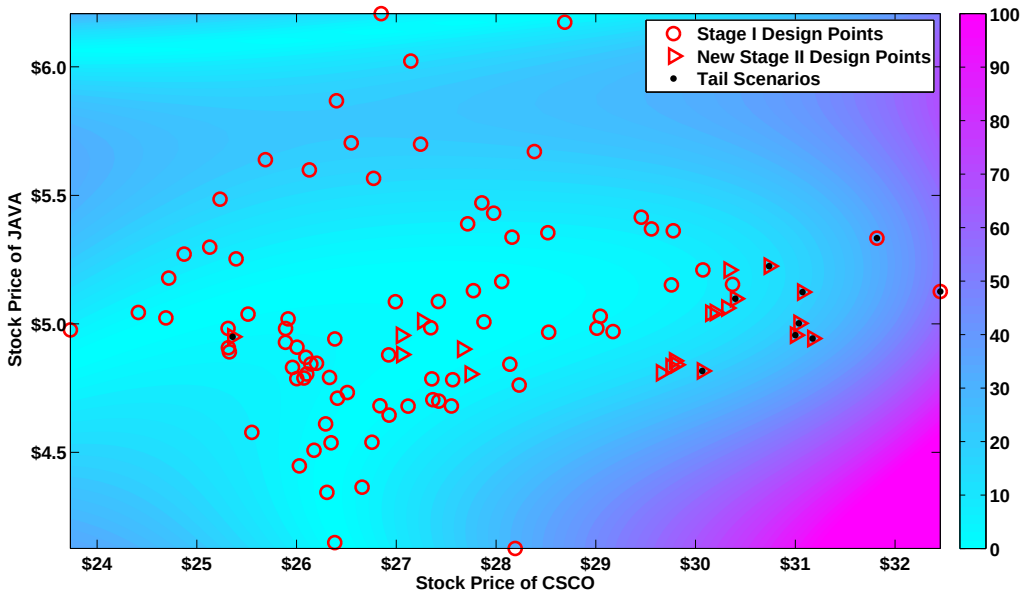


Figure 3.17. Absolute error of the Stage II metamodel on one run of the green procedure in the second experiment.

The strength of our GNS procedure is well illustrated in Figure 3.18, which depicts the allocation the second experiment's simulation budget among all design points considered in both experiments. First of all, similar to Stage III of the baseline procedure, most of the simulation budget is allocated to design points that are tail scenarios or near tail scenarios. Secondly, by reusing the first experiment's simulation output the GNS augmentation step

in Stage I uses far less simulation budget than the case without any previous simulation output (e.g. 4 new design points in the augmentation step versus $k_1 = 50$ new Stage I design points in the baseline procedure). These savings allow for a larger budget to be allocated in later stages of the GNS procedure. As discussed above, the augmented Stage I metamodel has high quality and leads to correct identification of tail scenarios. In the example that we have shown so far, some tail scenarios in the second experiment receives more than three times as much simulation budget than the tail scenarios in the first experiment. Such high concentration of simulation budget on tail scenarios significantly increases the accuracy of the ES estimator, which is the goal of this GNS procedure. Last but not least, Stage III in our GNS procedure has a flexible design that, if needed, can allocate simulation budget to previous design points, if doing so can improve the accuracy of the resulting ES estimator.

Figure 3.19 shows the absolute error $|\widetilde{V} - V|$ of Stage III metamodel in the second experiment. We see a significant improvement of the Stage III metamodel over the Stage II metamodel, especially in the neighborhoods of the tail scenarios. This improvement is a result of the intelligent budget allocation scheme of the baseline procedure.

To examine the practical performance of our GNS procedure, we conduct a numerical study for 15 repeated experiments on 100 independent sample paths of stock prices. In particular, the stock prices in each sample path are modeled as geometric Brownian motions whose drift and volatility parameters are estimated from historical return observations. In each sample path, the experiments are repeated daily for 15 days. Each experiment has 1,000 design points, all are obtained on historical return observations. For each sequence of repeated, we apply the repeated baseline procedure in Figure 3.13 and our GNS procedure

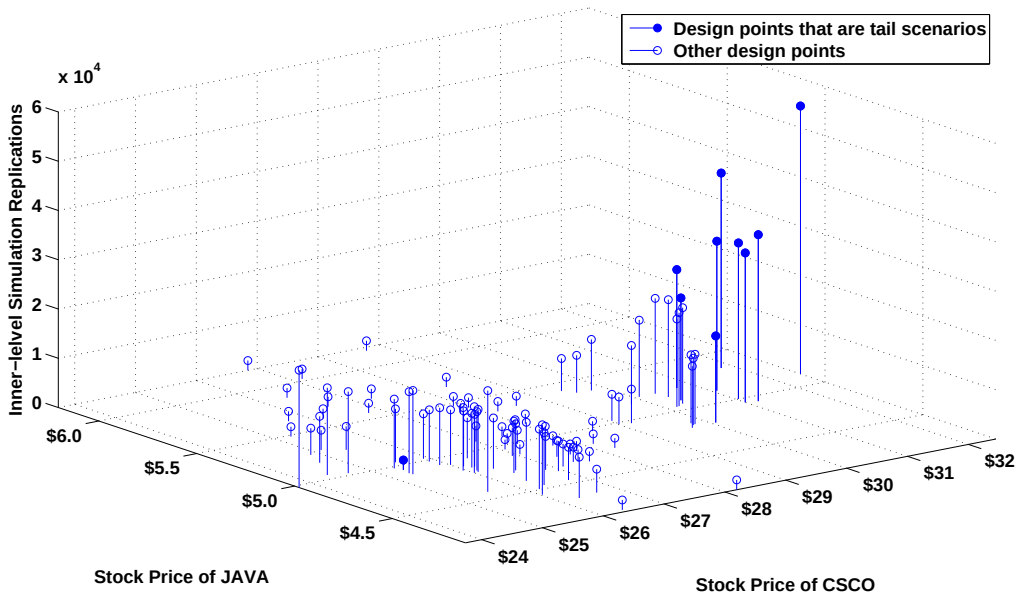


Figure 3.18. Number of simulation replications allocated to all design points considered in both the first and the second experiments.

in Figure 3.14. Design parameters of both procedures are the same and are consistent with the discussions above. We acknowledge that it is a conceptual flaw to perform daily risk assessment for the same portfolio of options, particularly with the same maturities in each experiment. Despite this conceptual flaw, this example is simple enough for intuitive graphical display yet sufficiently complicated to illustrate green simulation design in nested simulation experiments.

Figure 3.20 shows the relative root mean squared error (RRMSE) of the repeated baseline and the GNS procedures' ES estimators, along with their 95% error bars. The repeated baseline procedure's RRMSE is shown by black dotted line and the GNS procedure's RRMSE is shown by green solid line. The RRMSEs for both procedures coincide in the first experiment because there is no preexisting simulation output so the two procedures

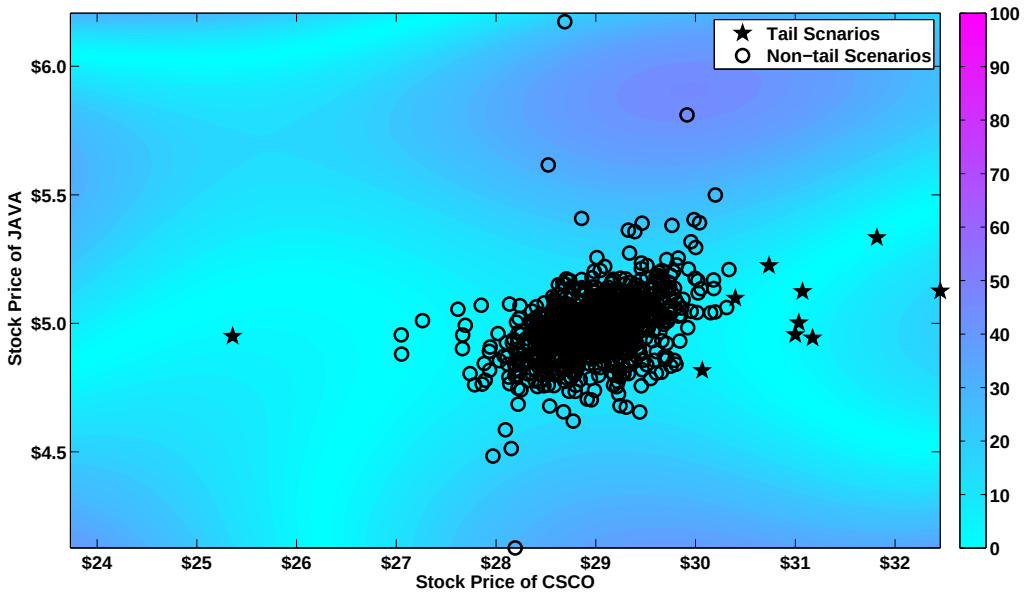


Figure 3.19. Absolute error of the Stage III metamodel on one run of the green procedure in the second experiment.

are identical. As a benchmark, the RRMSE of the repeated baseline procedure remains at a constant level. In contrast, the GNS procedure's estimator has improving accuracy as more simulation outputs are reused. In the 15th experiment, with the same simulation budget, the GNS procedure's is about 10 times more accurate than the baseline procedure. More importantly, the GNS procedure seems to have much room for improving accuracy beyond the 15th experiment.

3.6. Future Research Directions

Despite the success of our GNS procedure for its improving accuracy in repeated nested simulation experiments, it has a few weaknesses that we will address in the following discussions.

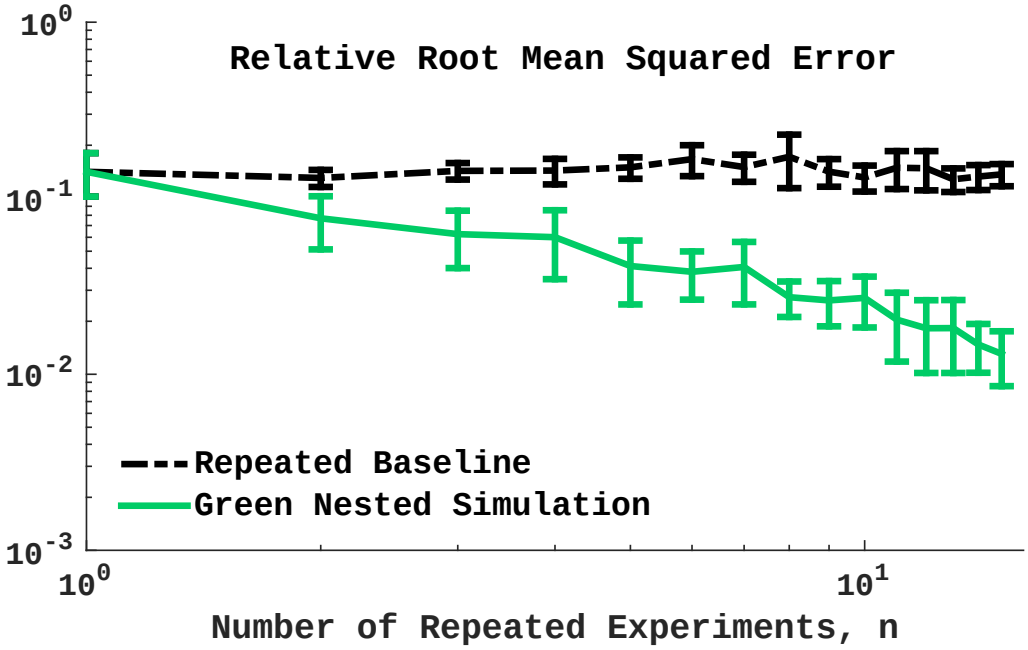


Figure 3.20. Log-log plot of relative mean squared error and error bar for 15 repeated experiments of repeated baseline procedure (dotted black line) and green nested simulation procedure (solid green line).

3.6.1. Metamodeling with many design points

Fitting a metamodel with many design points can be a challenging task, especially for stochastic kriging. Large numbers of design points usually leads to excessive computations and numerical difficulties. If some of the design points are close to each other, then the fitting and prediction may encounter numerical difficulty. In our GNS procedure, the number of design points is likely to increase from one experiment to the next. Moreover, for the motivating example that we consider, the design points are likely to be densely located in regions where the portfolio value is low (near tail scenarios). As the preexisting simulation output accumulates as more experiments are done, we will eventually be fitting

a stochastic kriging metamodel with many design points and using it for predictions. Consequently, in our numerical study, the GNS procedure in the 15th experiment can be over one hundred times slower than the repeated baseline procedure and this gap is likely to widen as more experiments are repeated.

In future research we may address this challenge using some of the following methods:

(1) Reuse simulation output selectively. When abundant simulation output is available from past experiments, it is not always a good idea to reuse all of it in the current experiment. The credit risk problem studied in Section 2.5.2 is one such example. In these cases it may be beneficial to discard the irrelevant or even misleading simulation output and only reuse the output that is most informative for the current experiment. By reusing simulation output selectively we can limit the number of design points when creating the stochastic kriging metamodels in the GNS procedure. Selecting the available simulation output in a meaningful way is an important research question that we will attempt to tackle in the future.

(2) Consider other metamodeling techniques. While we considered stochastic kriging exclusively in the GNS procedure, one can design similar nested simulation procedure that uses other metamodeling techniques such as regression, splines, and neural networks, among others. Each metamodeling technique has its own features and drawbacks. Neural networks, for example, is a popular technique in many big-data applications and can handle many design points.

(3) Select representative design points. When there are many design points that are clustered in different regions, it may be a good ideas to use a representative design point for each cluster when fitting a metamodel. Well-chosen representative design

points may contain almost as much information as all design points but lead to a much less computationally demanding metamodel.

3.6.2. Green simulation designs with fixed accuracy

The GNS procedure as well as the previous LR and GSK estimators are all fixed-budget simulation designs. There is also great interest in the literature, such as Tongarlak et al. (2011) and Chen and Li (2014), for fixed-accuracy simulation designs. To be specific, if the decision maker has a prespecified error tolerance and abundant computational resources then a fixed-accuracy simulation design is appropriate.

Green simulation as new paradigm is applicable to both fixed-budget and fix-accuracy designs. For example, we can easily modify our GNS procedure from a fix-budget to a fixed-accuracy procedure by changing the budget allocation scheme in Stage III. Specifically, instead of solving optimization problem (3.9), we can solve

$$\min_{n_1,\ldots,n_k} \left\{ \sum_{i=1}^{k} n_i \middle| \boldsymbol{U}^\top \boldsymbol{V} \boldsymbol{N}^{-1} \boldsymbol{U} \leq v^*, n_i \geq n_0, \forall i = 1, \ldots, k \right\} \tag{3.10}$$

where v^* is a given constant that relates to the maximum allowable posterior variance of the ES estimator. The constraint $\boldsymbol{U}^\top \boldsymbol{V} \boldsymbol{N}^{-1} \boldsymbol{U} \leq v^*$ may seem complicated at the first glance, but its left hand side can be written as $\sum_{i=1}^{k} U_i^2 V_i / n_i$ where $\boldsymbol{U}$ and $\boldsymbol{V}$ are constants. There may be efficient algorithm for solving (3.10) by examining its optimality conditions. Based on the encouraging results from the GNS procedure, we envision that this fixed-accuracy procedure can also be benefited from reusing preexisting simulation outputs.

CHAPTER 4

Green Simulation via Database Monte Carlo

4.1. Introduction

In this chapter, we develop and analyze a new green simulation procedure that reuses simulation outputs using Database Monte Carlo (Borogovac and Vakili, 2008). Similar to the green LR estimators studied in Chapter 2, we consider the convergence properties of the proposed green Database Monte Carlo (DBMC) estimator.

When simulation is used as a decision support tool for routine tasks, its usage often follows a cyclic pattern: Firstly, the current state of the system, such as market price of a stock, is observed and used as the input for the simulation model. Based on the observed input, a time-constrained simulation experiment is run so that an answer is delivered within the given time limit, e.g. half a working day. When an experiment is repeated routinely, e.g, the decision support cycle finishes when an answer is delivered for the current task, the computational resource remains idle until the next experiment begins. We propose adding a *simulation investment* step in the above cycle so that such idle resource can be invested to improve the computational efficiency of future experiments. The split of available computations into simulation experiment and simulation investment fits the framework of Database Monte Carlo (Borogovac and Vakili, 2008), which constructs databases of simulation outputs and then use them as quasi-control variates to improve the accuracy of subsequent experiments. A fundamental building block for our green Database Monte

Carlo (GreenDB) procedure is the recent method of Database Monte Carlo for simulation on demand by Rosenbaum and Staum (2015), which views the database construction as one-time computational investment that benefits all future experiments. Particularly, after the constructing a "good" database, it can be used to provide good quasi-control variates for multiple experiments to improve their speed and accuracy. In the DBMC for simulation on demand framework, the database remains unchanged after initial construction. So the efficiency of their method depends heavily on the quality of the database. In contrast, our GreenDB procedure repeatedly make simulation investments into database to enhance it over time. Consequently, in addition to improved efficiency due to DBMC, the our GreenDB procedure's efficiency improves over time, as more experiments are repeated and more simulation investment is made.

As alluded above, the efficiency of our GreenDB depends on how well we invest the idle resources. A simulation budget allocation problem is considered to maximize the variance reduction of our GreenDB estimator. This problem is similar to those studied in security pricing by simulating stochastic differential equations, such as Duffie and Glynn (1995) and Boyle et al. (1997). The solution leads to a dynamic allocation policy of the idle computational resource. The optimal objective value also suggests a possible rate of convergence of our GreenDB estimator. Results in our numerical study show that the GreenDB estimator has improving accuracy as more experiments are repeated and more simulation investment is made, despite a fixed time constraint and fixed investment in each experiment.

4.2. Mathematical Framework

This section specifies a mathematical framework for developing a green DBMCsimulation procedure in the remainder of this chapter. We first describe simulation output as a function of inputs using the concept of random fields then present a setting of repeated experiments where the same simulation model is run repeatedly with changing inputs.

Let $\mathcal{X} \subseteq \mathbb{R}^s$ be the set of all possible inputs, or the *input space*, for a simulation model of interest. Given an input $x \in \mathcal{X}$, or a *point* in the input space, a simulation experiment is run and the simulation output is regarded as a random variable: denote it $F(x)$. In this article we consider the simulation output F as a random field (Staum, 2009). Specifically, the simulation output is a measurable function $F : \mathcal{X} \times \Omega \mapsto \mathbb{R}$, where Ω is a probability space with probability measure $\mathbb{P}$ on it. The mean function $\mu : \mathcal{X} \mapsto \mathbb{R}$, which is also known as the response surface, is defined by

$$\mu(x) = \mathbb{E}[F(x)] = \int_{\Omega} F(x, \omega) d\mathbb{P}(\omega), \quad \forall x \in \mathcal{X}. \tag{4.1}$$

The covariance function $\sigma : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ and the correlation function $\rho : \mathcal{X} \times \mathcal{X} \mapsto [-1, 1]$ of the random field are given by

$$\sigma(x, x') := \int_{\Omega} F(x, \omega) F(x', \omega) d\mathbb{P} - \mu(x)\mu(x') \tag{4.2}$$

and

$$\rho(x, x') := \frac{\sigma(x, x')}{\sqrt{\sigma_x^2 \sigma_{x'}^2}}, \quad \text{where} \quad \sigma_x^2 = \sigma(x, x). \tag{4.3}$$

As is standard in analyzing Monte Carlo simulations, we assume that $\sigma_x^2 < \infty$ for all $x \in \mathcal{X}$ so that all simulation outputs have finite variance.

Given a point $x \in \mathcal{X}$, the goal is to estimate $\mu(x)$ by a simulation experiment. In some cases, such as rare event simulations, one may estimate $\mu(x)$ by running the simulation model at a different point x'. To avoid ambiguity, we refer to the point at which μ is estimated, say x, as the *prediction point* and the point at which the simulation model is run as the *design point*, whether or not these points coincide. A useful feature of the random field description is the probabilistic dependence between the simulation outputs $F(x)$ and $F(x')$ when the simulation is run at different points $x, x' \in \mathcal{X}$. Such dependence represents the effect of common random numbers in simulating the same model at different points. Control variates is a popular variance reduction technique that improves simulation efficiency by leveraging dependency between random variables. Database Monte Carlo (Borogovac and Vakili, 2008) extends control variates to more general settings when the control variate's mean is replaced by its estimate. Database Monte Carlo for Simulation on Demand (Rosenbaum and Staum, 2015) articulates the order of implementation of Database Monte Carlo so that an accurate answer can be delivered quickly. We provide a brief review of these methods in Section 4.3.

We now present a setting of repeated experiment where the same simulation model is run with different prediction points at different times. Suppose that one experiment is run at time n, $n = 1, 2, \cdots$; the goal of this nth experiment is to estimate $\mu(X_n)$, where X_n be the nth prediction point. We treat $\{X_n : n = 1, 2, \cdots\}$ as a discrete time stochastic process taking values in the input space $\mathcal{X}$. We assume that X_n is observable at time n,

but was not observable earlier. Moreover, the prediction points can only be observed but the user of the simulation model cannot control their dynamics.

We now introduce terminologies and notations for databases in repeated experiment setting. As alluded before, in the nth experiment the simulation model may be run at a design point that is different from the nth prediction point. In particular we may run simulation at a design point $\widetilde{x}$ and store the outputs in one experiment then reuse these outputs to improve efficiency of future experiments. The storage of such outputs is called the *database* and the point $\widetilde{x}$ is called the *database point*; for convenience we refer to a database by its database point. Let $\{\widetilde{X}_k : k = 1, 2, \cdots\}$ be a database sequence taking values in the input space $\mathcal{X}$. The input space may be continuous or discrete, but in this article we assume that it is bounded so that the dispersion of given point set is well-defined. This sequence may be deterministic (e.g., low discrepancy sequence) or stochastic (e.g., a discrete time stochastic process). In later discussions it is desirable to run simulation at the same database at different times so we shall describe how the database sequence is visited. Let $s(k) : \mathbb{N} \mapsto \mathbb{N}$ be a *visit schedule* such that $\widetilde{X}_{s(k)}$ is the database point at which the kth experiment is run. The visit schedule may be a deterministic function of time $k \in \mathbb{N}$ (e.g., $s(k) = \lfloor\sqrt{k}\rfloor$) or be chosen at each time k according to some policy as in Section 4.3.2. For simplicity we assume that $s(1) = 1$ thus the first experiment is run at the first database in the sequence and that $s(n) \leq \max\{s(k) : k < n\} + 1$ so the visit schedule does not "skip" any database in the sequence $\widetilde{X}_n$; it can "revisit" any database that has been visited before. The database sequence $\{\widetilde{X}_k : k = 1, 2, \cdots\}$ and the design schedule $s(k)$ are both modeling choices by the user. Consequently, in contrast to prediction points, the user can choose the database point $\widetilde{X}_{s(n)}$ for the nth experiment.

For example if one chooses $\widetilde{X}_n = X_n$ and $s(n) = 1$ for $n = 1, 2, \cdots$ then all experiments are run at the first prediction point X_1. One contribution in this chapter is providing insights to choosing good database sequence $\{\widetilde{X}_k\}$ and visit schedule $s(k)$ that improves computation efficiency for repeated experiments.

4.3. Green Simulation via Database Monte Carlo

In this section we develop a green simulation procedure that reuses simulation outputs to improve computational efficiency in repeated experiments. In each experiment, the proposed procedure not only delivers accurate answers quickly but also invests simulation efforts to improve the efficiency of future experiments. As more simulation investments are made, the answer delivered can be arbitrarily accurate.

We first review Database Monte Carlo for Simulation on Demand and related methods. Given a prediction point $x \in \mathcal{X}$, the *Standard Monte Carlo* (SMC) estimator for $\mu(x)$ is given by

$$\widehat{\mu}_r^{SMC}(x) = \frac{1}{r}\sum_{j=1}^{r} F^{(j)}(x) \tag{4.4}$$

where the outputs $F^{(1)}(x), \cdots, F^{(r)}(x)$ are r independent replications of the simulation model run at point x. The SMC estimator is often used in practice for its simplicity and theoretical justifications such as the Law of Large Numbers and the Central Limit Theorem. The SMC estimator has variance σ_x^2/r but may be reduced by various variance reduction techniques that take advantages of certain properties of the simulation model. For example, control variate (CV) is a variance reduction technique that takes advantages of correlation between random variables. For simplicity of exposition we consider only a

single CV. Suppose the $F(x')$ is chosen as a CV for $F(x)$, a *classical CV* estimator of $\mu(x)$ has the form

$$\widehat{\mu}_r^{CV}(x;x') = \widehat{\mu}_r^{SMC}(x) - \beta\left[\widehat{\mu}_r^{SMC}(x') - \mu(x')\right] \tag{4.5}$$

where:

- The simulation is run for r independent replications: those in $\widehat{\mu}_r^{SMC}(x)$ are dependent with those $\widehat{\mu}_r^{SMC}(x')$ since they are simulated using common random numbers.
- The CV coefficient β is usually chosen to minimize the variance of $\widehat{\mu}_r^{CV}(x;x')$. The optimal coefficient is $\beta^* = \mathbb{C}\text{ov}[F(x), F(x')]/\mathbb{V}\text{ar}[F(x')]$, provided that the required population quantities are available. The resulting estimator has zero bias and minimized variance $(1-(\rho(x,x'))^2)\sigma_x^2/r$, i.e., variance reduction of $1/(1-(\rho(x,x'))^2)$.

The classical CV (4.5) assumes that the CV expectation $\mu(x')$ is known, which may not be true in all applications. When $\mu(x')$ is not known, studies show that $F(x')$ can still be used as a quasi-CV (Emsermann and Simon, 2002) or CV with estimated mean (Pasupathy et al., 2012) by replacing $\mu(x')$ with its estimate. Database Monte Carlo with control variate (DBCV) is one such method that replaces $\mu(x')$ with $\widehat{\mu}_R^{SMC}(x')$ for some large R. Mathematically, a DBCV estimator of $\mu(x)$ has the form

$$\widehat{\mu}_{r,R}^{DBCV}(x;x') = \widehat{\mu}_r^{SMC}(x) - \beta\left[\widehat{\mu}_r^{SMC}(x') - \widehat{\mu}_R^{SMC}(x')\right]. \tag{4.6}$$

We see that R independent replications of $F(x')$ are required for estimating $\widehat{\mu}_R^{SMC}(x')$ in (4.6). These R replications constitute a database at x'. DBCV is effective when $R \gg r$, which is assumed in subsequent discussions. In practice the database stores the SMC estimate $\widehat{\mu}_R^{SMC}(x')$ instead of the individual replications, nevertheless we say that the database has size R. Due to the use of quasi-CV, the variance-minimizing coefficient β and the resulting minimum variance are different from those of classical CV. Readers are encouraged to find details in Emsermann and Simon (2002), Borogovac and Vakili (2008), and Pasupathy et al. (2012), among others. In this chapter we consider an approximate variance of $\widehat{\mu}_r^{DBCV}(x;x')$ for simplicity in later analysis. Let $\beta = \beta(x;x')$ be the optimal CV coefficient when $F(x')$ is used as a classical CV for $F(x)$, then

$$\mathbb{V}\mathrm{ar}[\widehat{\mu}_{r,R}^{DBCV}(x;x')] = \mathbb{V}\mathrm{ar}[\widehat{\mu}_r^{CV}(x;x')] + \beta^2 \mathbb{V}\mathrm{ar}[\widehat{\mu}_R^{SMC}(x')] \tag{4.7a}$$

$$= \frac{(1-(\rho(x,x'))^2)}{r}\sigma_x^2 + \frac{\beta^2}{R}\sigma_{x'}^2 \tag{4.7b}$$

$$\approx \left[\frac{1}{R} + \frac{(1-(\rho(x,x'))^2)}{r}\right]\sigma_x^2 \tag{4.7c}$$

where the approximation follows if $F(x)$ and $F(x')$ are similar so that $\sigma_{x'}^2 \approx \sigma_x^2$ and $\beta \approx 1$. Despite being an approximation,(4.7) showcases a main feature of the minimum variance of DBCV estimator (see Equation (6) in Rosenbaum and Staum (2015)): it converges to zero as the database's size R grows to infinity and the magnitude of the correlation $\rho(x,x')$ converges to 1.

At the first glance, the DBCV estimator (4.6) may be computationally expensive because it requires running $R+2r$ replications of the simulation model. Database Monte Carlo for Simulation on Demand (DBSD) is an attractive procedure that uses the DBCV

estimator to deliver accurate answers quickly by separating the *construction* of the database from the *estimation* of expected simulation output. A basic version of DBSD can be summarized in the following two steps:

(1) *Before* the prediction point x is observed, choose a *database point* x' and run $r + R$ replications at x'(construction of a database).

(2) *After* the prediction point x is observed, run r replications at x and deliver $\widehat{\mu}^{DBMC}(x)$ in (4.6) as an estimate for $\mu(x)$ (estimation of $\mu(x)$).

In the procedure above, it takes only r replications between observing x and delivering an estimate of $\mu(x)$. Database construction is done prior to observing the prediction point and hence is not considered part of the estimation task. Moreover, database construction can potentially improve the efficiency of many future experiments and hence is one form of *simulation investment*. The computation patterns in the DBSD procedure is applicable in many practical settings: usually there is a time limit (e.g., during working hours) between observing the input of a simulation and delivering an answer via simulation, but computational resources may be available after an answer is delivered (e.g., after working hours), when one can made simulation investments that can benefit future experiments.

4.3.1. Basic Green Database Monte Carlo with Control Variate

In the original proposal of DBSD, database construction is a one-time investment, i.e., the locations and sizes of databases remain unchanged once constructed. Consequently the DBCV estimator is conditionally biased given constructed databases and the variance reduction it provides is limited by the quality of the available databases. In the context of repeated experiment, we propose a green simulation procedure in which simulation

investments are made repeatedly: new databases maybe constructed and existing databases may grow in size over time. At each time n, we suppose that the simulation model can run a total of $r + R$ replications: an answer must be delivered after r replications while the remaining R more replications, the simulation investment, can be run before the next experiment begins. For clarity, we consider two types of simulation investment: *augmentation*, which adds more replications to an existing database and *initiation*, which runs simulation at a new database point and constructs a new database. We propose the following *Green Database Monte Carlo* procedure for the nth experiment:

(1) Suppose at the beginning of time n there are k_n databases $\{\widetilde{X}_1, \cdots \widetilde{X}_{k_n}\}$ with sizes $R_1 + r, \cdots, R_{k_n} + r$; all these databases have r replications that were run using common random numbers.

(2) After prediction point X_n is revealed, choose one database, say $\widetilde{X}_{\widetilde{k}_n}$, as a quasi-CV.

(3) Run r replications at X_n using the same common random numbers as those in the databases.

(4) Deliver

$$\widehat{\mu}_r^{GreenDB}\left(X_n; \widetilde{X}_{\widetilde{k}_n}\right) = \widehat{\mu}_{r,R_{k_n^*}}^{DBCV}\left(X_n; \widetilde{X}_{\widetilde{k}_n}\right) \tag{4.8}$$

as an estimate of $\mu(X_n)$, where the parameter β_n is chosen to minimize the variance of $\widehat{\mu}_r^{GreenDB}\left(X_n; \widetilde{X}_{\widetilde{k}_n}\right)$. Estimation of β_n is discussed in Section 4.3.2.

(5) After delivering the answer, run R replications at $\widetilde{X}_{s(n)}$ based on the database sequence $\{\widetilde{X}_n\}$ and visit schedule $s(n)$. This maybe an augmentation to an existing database if $s(n) \in \{1, \cdots, k_n\}$ or initiation of a new database at $\widetilde{X}_{s(n)}$ otherwise.

Similar to DBSD, GreenDB delivers accurate answers quickly after a prediction point is revealed. In contrast to the one-time investment in DBSD, GreenDB makes simulation investment repeatedly in repeated experiments. As more investments are made, either existing databases grow in size or more databases are constructed, or both. As a result, GreenDB is superior than DBSD in repeated experiments.

4.3.2. Designing Practical Green Database Monte Carlo Procedures

Under some conditions, a well-designed GreenDB procedure can produce converging estimators similar to those in Feng and Staum (2015), which can achieve arbitrary accuracy as more and more simulation investment is made. We will supply more details to the basic GreenDB procedure and consider designing an efficient procedure. In particular, we will specify: Construction of the initial database; selection of quasi-CV among existing databases; and estimation of the CV coefficient $\widehat{\beta}$. Lastly we consider a simulation investment allocation problem whose solution provides useful insights in choosing database sequence $\{\widetilde{X}_k\}$ and visit schedule $s(k)$.

For the first database, consider the first experiment with prediction point X_1, when there is no preexisting database. Since an answer is required within r replications of the simulation model, without any other variance technique (e.g. importance sampling) one should simply run r replications at X_1 and estimate $\mu(X_1)$ by $\widehat{\mu}_r^{SMC}(X_1)$. Given a database sequence $\{\widetilde{X}_k\}$ and visit schedule $s(k)$, the remaining R replications of simulation investment should be run at $\widetilde{X}_{s(1)} = \widetilde{X}_1$. When a database is first initiated, its size is R. Note that r of these R replications should serve as common random numbers for estimations at future prediction points. The database initiated at time 1 may be a

quasi-CV for future experiments. However, as discussed later, a new database may not always be candidate as quasi-CV immediately after initiation.

While selection of quasi-CV among existing databases and estimation of $\widehat{\beta}$ are not our focuses, we review some theoretical criteria and practical suggestions for these tasks. Theoretically, in classical CV one should select a database $\widetilde{X}_{\widetilde{k}_n}$ such that $|\rho(X_n, \widetilde{X}_{\widetilde{k}_n})|$ is maximized to achieve maximum variance reduction. However, in general the correlation function $\rho(x, x')$ for a given simulation model may be prohibitively complicated, if possible at all, to optimize. In many practical applications, outputs of the same simulation model are highly correlated when run at similar inputs. Then, as a common practical alternative, one may choose the "nearest neighbor" database of the prediction point

$$\widetilde{X}_{\widetilde{k}_n} = \underset{\widetilde{X}_k \in \{\widetilde{X}_1, \cdots, \widetilde{X}_{k_n}\}}{\arg\min} \{\|\widetilde{X}_k - X_n\|_2\}. \tag{4.9}$$

For example, the above rule is used in the numerical examples of Rosenbaum and Staum (2015).

Estimation of $\widehat{\beta}$ in DBCV has been studied by Pasupathy et al. (2012) that accounts for the use of estimated mean and Avramidis and Wilson (1993) that eliminates the bias, among others. We did not consider applying those techniques to GreenDB in this chapter, it remains for future research. As suggested by Rosenbaum and Staum (2015), we estimate $\widehat{\beta}$ by linear regression of $F^{(1)}(X_n), \cdots, F^{(r)}(X_n)$ on $F^{(1)}(\widetilde{X}_{\widetilde{k}_n}), \cdots, F^{(r)}(\widetilde{X}_{\widetilde{k}_n})$.

The database sequence $\{\widetilde{X}_k\}$ and the visit schedule $s(k)$ are the two key factors in designing an efficient GreenDB procedure. In particular, we are interested in developing GreenDB estimator such that $\mathbb{V}\mathrm{ar}[\widehat{\mu}_r^{GreenDB}\left(X_n; \widetilde{X}_{\widetilde{k}_n}\right)] \to 0$ as $n \to \infty$. We will consider some requirements for $\{\widetilde{X}_k\}$ and $s(k)$ so that the convergence is fast. Assume that $\rho(x, x') \to 1$

as $\|x - x'\|_2 \to 0$ for all $x, x' \in \mathcal{X}$. Based on (4.7) we have $\mathbb{V}\text{ar}\left[\widehat{\mu}_r^{GreenDB}\left(X_n; \widetilde{X}_{\widetilde{k}_n}\right)\right] \to 0$ as $n \to \infty$ if

(c1) $\|X_n - \widetilde{X}_{\widetilde{k}_n}\|_2 \to 0$ as $n \to \infty$, and

(c2) $R_{k_n^*} \to \infty$ as $n \to \infty$.

Consider the bounded metric space[1] $(\mathcal{X}, \|\cdot\|_2)$, the nth *dispersion* (Niederreiter, 1992, Definition 6.2) of the database sequence $\{\widetilde{X}_k\}$ by

$$d_n := \sup_{x \in \mathcal{X}} \min_{1 \le k \le n} \|x - \widetilde{X}_k\|_2.$$

Then we have $\|X_n - \widetilde{X}_{\widetilde{k}_n}\|_2 \le d_n$ for all $n = 1, 2, \cdots$. Therefore a sufficient condition for (c1) is $d_n \to 0$ as $n \to \infty$, which implies that the database sequence is space-filling and more and more databases are initiated as time progresses. Based on studies of Quasi-Monte Carlo (QMC), we propose using low-discrepancy sequence in $\mathcal{X}$ for $\{\widetilde{X}_n : n = 1, 2, \cdots\}$. Since a fixed simulation investment of R replications is made to database construction in each experiment. The condition (c2) requires that $s(k)$ revisits all databases infinitely many times as time progresses. We now solve a simulation investment allocation problem whose optimal solution leads to a visit schedule that satisfies the requirements above and fast convergence of the resulting GreenDB estimator. Let $C_n = nR$ be the total simulation investment budget for database construction by the end of the nth experiment. For simplicity, we assume that the decision variables k_n and R_n are continuous and that $k_n \cdot R_n = C_n$. Effectively, we are assuming that all databases have the same size. In future research, we may investigate more sophisticated resource allocation schemes such that databases have different sizes in different regions of the input space. The goal is to find an

[1]The input space $\mathcal{X}$ is assumed to be bounded.

optimal allocation so as to maximize variance reduction in (4.7). Then a visit schedule $s(k)$ can be formulated using a dynamic policy that approximately matches the optimal solution k_n^* and R_n^* at each time n.

Assume that $1-(\rho(x,x'))^2 \leq c_1\|x-x'\|_2^\alpha$ for some $c_1, \alpha \in \mathbb{R}_+$ for all $x, x' \in \mathcal{X}$. It follows from Niederreiter (1992) that the dispersion of some s-dimensional low-discrepancy sequence (Solbol and Halton sequences, for example) is bounded by $d_n \leq c_2 k_n^{-1/s}$ for some $c_2 \in \mathbb{R}_+$. Then we have $1-(\rho(X_n, \widetilde{X}_{k_n^*}))^2 \leq c_1\|x-x'\|_2^\alpha \leq d_n \leq cN_n^{-\alpha/s}$ where $c = c_1 \cdot c_2^\alpha$. Consequently the simulation investment allocation problem is given by

$$\begin{aligned} \min_{k_n, R_n \in \mathbb{R}_+} \quad & \tfrac{1}{R_n} + \tfrac{c}{r} k_n^{-\alpha/s} \\ \text{s.t.} \quad & R_n k_n = C_n \end{aligned} \tag{4.10}$$

Substituting $R_n = C_n/k_n = (nR)/k_n$ into the objective then we may minimize the scaler-valued function $f(k_n) = \frac{k_n}{(nR)} + \frac{c}{r}k_n^{-\alpha/s}$. Clearly f is a convex function therefore its minimizer is given by

$$\frac{df(k_n)}{dk_n} = 0 \Rightarrow k_n^* = (nR)^{\frac{s}{\alpha+s}} \left(\frac{c\cdot\alpha}{r\cdot s}\right)^{\frac{s}{\alpha+s}}, \quad R_n^* = (nR)^{\frac{\alpha}{\alpha+s}} \left(\frac{r\cdot s}{c\cdot\alpha}\right)^{\frac{s}{\alpha+s}}. \tag{4.11}$$

The optimal solution (4.11) shows that:

(1) For any $\alpha, c_1, c_2, s > 0$ and for any $r, R > 0$, $k_n^* \to \infty$ and $R_n^* \to \infty$ as $n \to \infty$. In other words, as more simulations are done and more simulation investments are made, both the number and the size of databases increase indefinitely. As shown in the previous discussions, these are sufficient conditions for $\mathbb{V}\text{ar}[\widehat{\mu}_r^{GreenDB}\left(X_n; \widetilde{X}_{\widetilde{k}_n}\right)] \to 0$ as $n \to \infty$. In other words, the GreenDB estimator can be arbitrarily accurate as more and more simulation investment is made.

(2) For large n and fixed r, R, c_1, and c_2, the k_n^* decreases with α and R_n^* increases with α. This suggest that for larger α one should construct less databases that are sparser but with larger size. This is because for larger α the prediction point and the chosen quasi-CV may be further apart but still achieve a particular correlation level, which is bounded by the dispersion of the database sequence in the allocation problem. Note that α measures the correlation of simulation outputs associated with two similar inputs: the larger the α the higher the correlation.

(3) The optimal objective, and hence the variance reduction, is of order $\mathcal{O}(n^{-\alpha/(\alpha+s)})$. This suggests that, if $\sup\{\sigma_x^2 : x \in \mathcal{X}\} < \infty$, then the variance of our green estimator converges at rate $\mathcal{O}(n^{-\alpha/(\alpha+s)})$.

Exact implementation of optimal solution (4.11) suggests that, in the $(n+1)$th repeated experiment, invest $(R_{n+1}^* - R_{n^*})$ replications in the k_n^* existing databases and R_{n+1}^* replications in $k_{n+1}^* - k_n^*$ new databases, which raises practical difficulties: The optimal solution may not be integral; More importantly investing in multiple databases in each experiment may not be desirable. Instead, we suggest the following dynamic policy for the visit schedule Let $\{\widetilde{X}_1, \cdots, \widetilde{X}_{k_n}\}$ be the existing databases time n and let $R_1 + r, \cdots, R_{k_n} + r$ be their sizes. The design schedule $s(n)$ at time n satisfies:

(1) If $k_n < k_n^*$ where k_n^* is the optimal solution in (4.11), then $s(n) = k_n + 1$ so a new database is initiated with size R.

(2) Otherwise, if $k_n \geq k_n^*$, then $s(n) = \min\{k : R_k = \min\{R_{k'} : k' = 1, \cdots, k_n\}\}$ so one existing database with minimum size is augmented (break ties in order of initiations).

According to the policy, simulation investment is made to one database in each experiment, which simplifies the implementation of the GreenDB procedure. By adding investment into the database with the minimum size, the policy aims to balance the size in all existing database. One drawback of the policy is that any newly initiated database has size R and only increments by at most R in each subsequent experiment. When n is large, new database may be significantly smaller than other existing databases, which results in much worse variance reduction. As a remedy, we suggest that a database can only be a candidate for quasi-CV if its size is no more than R less than that of the first database. This additional safeguard ensures improving quality of the quasi-CV and fast convergence of the resulting GreenDB estimator. We will examine the practical performance of the GreenDB procedure by numerical examples in Section 4.4.

4.4. Illustration

To illustrate the essences of our procedure, we consider a repeated experiment that estimates the probability of a random variable exceeds a given threshold, where the random variable in question changes in each experiment. The simulation model in this experiment is simple so that the mean and covariance function of the corresponding random field can be derived analytically, which enables us to closely examine the practical performance of the proposed GreenDB procedure.

Let $\mathcal{X} = [x_{min}, x_{max}] \subset \mathbb{R}$ be the input space and $\omega \sim \text{Unif}(0, 1)$ be the underlying source of randomness for a given simulation model F. For given input $x \in \mathcal{X}$ and realization ω, the simulation output is given by $F(x, \omega) = \mathbf{1}\{x \cdot \omega > \gamma\}$ where $\mathbf{1}$ is the indicator function and $\gamma \in \mathbb{R}$.

Clearly this is a 1-dimensional problem and therefore $s = 1$. For any $x, x' \in [x_{min}, x_{max}]$, the mean function for the above simulation model is given by

$$\mu(x) = Pr\left(U > \frac{\gamma}{x}\right) = 1 - \frac{\gamma}{x}, \quad \forall x \in [x_{min}, x_{max}]$$

and the covariance function is given by

$$\begin{aligned}
\sigma(x, x') &= E[F(x) \cdot F(x')] - E[F(x)] \cdot E[F(x')] \\
&= P\left(U > \tfrac{\gamma}{\min\{x,x'\}}\right) - P\left(U > \tfrac{\gamma}{x}\right) \cdot P\left(U > \tfrac{\gamma}{x'}\right) \\
&= \left(1 - \tfrac{\gamma}{\min\{x,x'\}}\right) - \left(1 - \tfrac{\gamma}{x}\right) \cdot \left(1 - \tfrac{\gamma}{x'}\right) \\
&= \left(\tfrac{\min\{x,x'\}-\gamma}{\min\{x,x'\}}\right) \cdot \left(\tfrac{\gamma}{\max\{x,x'\}}\right).
\end{aligned}$$

Then one can show that for any $x, x' \in [x_{min}, x_{max}]$

$$\begin{aligned}
(\rho(x, x'))^2 &= \frac{\sigma(x,x')}{\sqrt{\sigma_x^2 \sigma_{x'}^2}} = \frac{\left(\frac{\min\{x,x'\}-\gamma}{\min\{x,x'\}}\right)^2 \cdot \left(\frac{\gamma}{\max\{x,x'\}}\right)^2}{\left(\frac{\min\{x,x'\}-\gamma}{\min\{x,x'\}}\right) \cdot \left(\frac{\gamma}{\min\{x,x'\}}\right) \cdot \left(\frac{\max\{x,x'\}-\gamma}{\max\{x,x'\}}\right) \cdot \left(\frac{\gamma}{\max\{x,x'\}}\right)} \\
&= \frac{\min\{x,x'\}-\gamma}{\max\{x,x'\}-\gamma} = 1 - \frac{\max\{x,x'\}-\min\{x,x'\}}{\max\{x,x'\}-\gamma}
\end{aligned}$$

so $1 - (\rho(x, x'))^2 \leq c_1 |x - x'|^\alpha$ for $c_1 = (x_{min} - \gamma)^{-1}$ and $\alpha = 1$. The database sequence is chosen as $\widetilde{X}_n = x_{min} + (x_{max} - x_{min}) h_n$ for all $n = 1, 2, \cdots$ where $h_1, h_2, \cdots$ is the Halton sequence. This means the nth dispersion of the database sequence is bounded by $d_n \leq 2(x_{max} - x_{min})/n$; so $c_2 = 2(x_{max} - x_{min})$. Moreover, the optimal control variate coefficient is given by

$$\beta^* = \frac{\sigma(x, x')}{\sigma_{x'}^2} = \begin{cases} \frac{x' \cdot (x-\gamma)}{x \cdot (x'-\gamma)} & \text{if} \quad x' \geq x \\ \frac{x'}{x} & \text{if} \quad x' < x \end{cases}$$

This optimal coefficient is used in the following numerical illustration. Note that for general simulation models the mean and covariance functions are too complicated to be derived analytically, so the parameters c_1, c_2, α, and β need to be estimated. This remains for future research.

Given prediction point X_n at time n, the goal of the nth simulation experiment is to estimate

$$\mu(X_n) = \mathrm{E}[F(X_n)] = Pr(X_n \cdot \omega > \gamma), \text{ where } \omega \sim \mathrm{Unif}(0,1). \tag{4.12}$$

For simplicity, the prediction points are $X_n \sim \mathrm{Unif}(x_{min}, x_{max})$, which forms a trivial stationary stochastic process with stationary distribution $X_n \sim \mathrm{Unif}(x_{min}, x_{max})$. We suppose that $r = 200$ and $R = 1000$ to emulate practical situations (4 hours to perform estimation task and 20 hours for simulation investment). General features of the GreenDB procedure, as discussed below, remains the same for wide range of parameters $0 < \gamma < x_{min} < x_{max} < \infty$. For illustration purpose we choose $x_{min} = 10$, $x_{max} = 12$, and $\gamma = 8$ and so $k_n^* = \sqrt{10n}$.

To investigate the effectiveness of the GreenDB procedure, we performed a sequence of 1,000 experiments, each with $r = 200$ replications for estimation and $R = 1000$ for simulation investment. Using the same sample path $\{X_n : n = 1, 2, \cdots, 1000\}$, we evaluated two estimators for $\mu(X_n)$: $\mu_r^{SMC}(X_n)$ and $\mu_r^{GreenDB}(X_n; \widetilde{X}_{\widetilde{k}_n})$. To accurately estimate the unconditional variance, we performed such a sequence of experiments 10,000 times. These 10,000 macro-replications of the sequence of experiments have independent sample paths and simulation output. The estimated variance of an estimator $\widehat{\mu}(X_n)$ of $\mu(X_n)$ was $\sum_{i=1}^{10,000}(\widehat{\mu}^{(i)}(X_n^{(i)}) - \mu(X_n^{(i)}))/10,000$, where $\widehat{\mu}(X_n^{(i)})$ is the value of the estimator

in the nth experiment on the ith macro-replication. Due to using $10,000$ macro-replications, the standard errors of these estimated variances are less than 0.01% of the corresponding estimated variances (error bars are too narrow to show clearly in the figure).

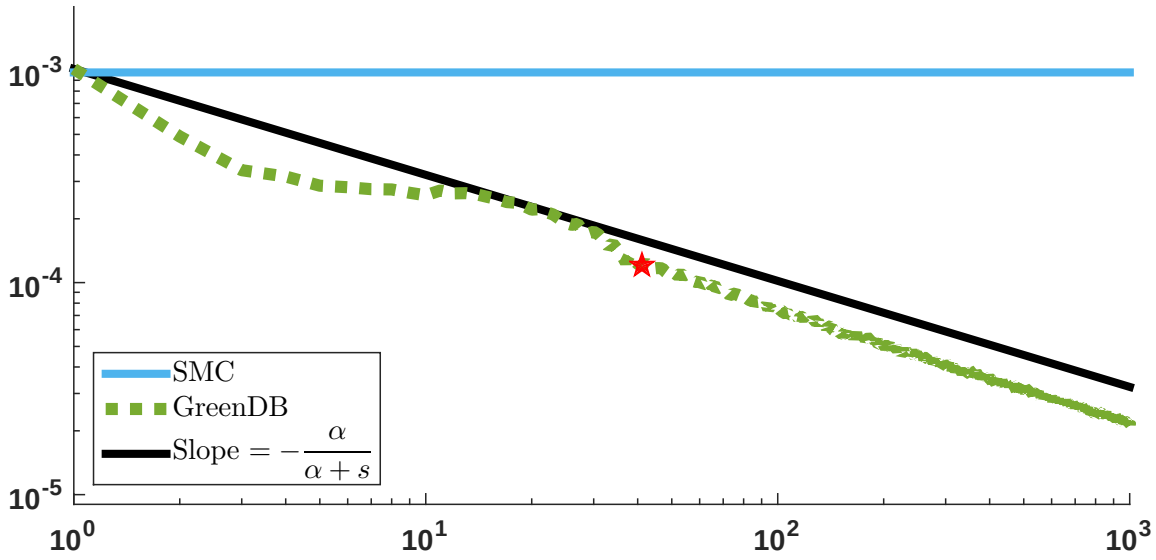


Figure 4.1. Log-log plot of estimated variances for Standard Monte Carlo and Green Simulation estimators for random prediction points X_n.

Figure 4.1 is a log-log plot of the variances of two estimators for each experiment $n = 1, 2 \cdots, 1,000$. The horizontal solid blue line is the variance of SMC estimator with r replications, and the dotted green line is the variance of our GreenDB estimator. The SMC variance forms a horizontal line because all experiments are identical the simulation model is the same and all prediction points have the same distribution. We first compare the dotted green line against the horizontal line to examine the effectiveness of our green simulation procedure. For the first experiment ($n = 1$), there is no database constructed from a previous experiment, so the GreenDB coincides with the SMC estimator. For all

later experiments ($n \geq 2$), when databases are available, the GreenDB variance is less than that for the SMC variance. It is clear that the variance of the GreenDB estimator decreases over time while the SMC estimators' variance remain the same. In the 1,000th experiment, the GreenDB estimator's variance is about 50 times smaller than the SMC estimator's variance, or equivalently a variance reduction of 50. This high accuracy for the GreenDB is achieved with the same estimation time as that for the SMC estimator, which showcases the advantage of our green simulation procedure.

A black solid line with slope $-\frac{\alpha}{\alpha+s}$ and the same intercept as the solid blue line is plotted in Figure 4.1 for reference. This slope is expressed in log-log scale, which indicates a convergence rate of $\mathcal{O}(n^{-\alpha/(\alpha+s)})$ for the variance of an estimator that has such slope in the figure. This is the convergence rate of the optimal objective for the simulation investment allocation problem (4.10). By comparing the right part of the green line (say $n \geq 100$) with the black line, it seems the GreenDB variance converges at rate $\mathcal{O}(n^{-\alpha/(\alpha+s)})$. We elaborate this point in the next paragraph.

A red star in Figure 4.1 splits the dotted green line into two segments that are visually different: the left segment forms a curve and the right segment forms a line. On one hand, the left segment is the warm up period in our procedure where the number and size of the databases constructed by our procedure are significantly different from the optimal solution (4.11). In addition, in this warm up period there are few databases so the prediction point and its closest database could be significantly different, rendering (4.7) an unreliable approximation. In this case the actual variance reduction is hard to predict by the optimal objective of (4.10). On the other hand, after warming up the right segment

forms a straight line whose slope is close to the predicted value of $-\alpha/(\alpha + s)$, when comparing to the solid black line.

References

Aiken, Leona S, Stephen G West, Raymond R Reno. 1991. *Multiple regression: Testing and interpreting interactions*. Sage.

Ankenman, Bruce, Barry L. Nelson, Jeremy Staum. 2010. Stochastic kriging for simulation metamodeling. *Operations Research* **58**(2) 371–382.

Avramidis, Athanassios N, James R Wilson. 1993. A splitting scheme for control variates. *Operations Research Letters* **14**(4) 187–198.

Barton, Russell R, Martin Meckesheimer. 2006. Metamodel-based simulation optimization. *Handbooks in operations research and management science* **13** 535–574.

Barton, Russell R., Barry L. Nelson, Wei Xie. 2014. Quantifying input uncertainty via simulation confidence intervals. *INFORMS Journal on Computing* **26**(1) 74–87.

Beckman, Richard J., Michael D. McKay. 1987. Monte Carlo estimation under different distributions using the same simulation. *Technometrics* **29**(2) 153–160.

Belsley, David A, Edwin Kuh, Roy E Welsch. 2005. *Regression diagnostics: Identifying influential data and sources of collinearity*, vol. 571. John Wiley & Sons.

Bitran, Gabriel R, Arnoldo C Hax. 1981. Disaggregation and resource allocation using convex knapsack problems with bounded variables. *Management Science* **27**(4) 431–441.

Blake, Eric S., Ethan J. Gibney. 2011. *The Deadliest, Costliest, and Most Intense United States Tropical Cyclones from 1851 to 2010 (and Other Frequently Requested Hurricane Facts)*. NOAA Technical Memorandum NWS NHC-6, National Weather Service, National Hurricane Center, Miami, Florida.

Borogovac, Tarik, Pirooz Vakili. 2008. Control variate technique: A constructive approach. *Proceedings of the 2008 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Inc., Piscataway, New Jersey, 320–327.

Boyle, Phelim, Mark Broadie, Paul Glasserman. 1997. Monte carlo methods for security pricing. *Journal of economic dynamics and control* **21**(8) 1267–1321.

Bretthauer, Kurt M, Anthony Ross, Bala Shetty. 1999. Nonlinear integer programming for optimal allocation in stratified sampling. *European Journal of Operational Research* **116**(3) 667–680.

Broadie, Mark, Yiping Du, Ciamac C Moallemi. 2011. Efficient risk estimation via nested sequential simulation. *Management Science* **57**(6) 1172–1194.

Chambers, M, CA Mount-Campbell. 2002. Process optimization via neural network metamodeling. *International Journal of Production Economics* **79**(2) 93–100.

Chen, E Jack, Min Li. 2014. Design of experiments for interpolation-based metamodels. *Simulation Modelling Practice and Theory* **44** 14–25.

Coval, Joshua D., Jakub W. Jurek, Erik Stafford. 2009. Economic catastrophe bonds. *The American Economic Review* **99**(3) 628–666.

Dassios, Angelos, Ji-Wook Jang. 2003. Pricing of catastrophe reinsurance and derivatives using the Cox process with shot noise intensity. *Finance and Stochastics* **7**(1) 73–95.

Duffie, Darrell, Peter Glynn. 1995. Efficient monte carlo simulation of security prices. *The Annals of Applied Probability* 897–905.

Efron, Bradley, Carl Morris. 1973. Stein's estimation rule and its competitorsan empirical bayes approach. *Journal of the American Statistical Association* **68**(341) 117–130.

Efron, Bradley, Carl Morris. 1975. Data analysis using stein's estimator and its generalizations. *Journal of the American Statistical Association* **70**(350) 311–319.

Emsermann, Markus, Burton Simon. 2002. Improving simulation efficiency with quasi control variates. *Stochastic Models* **18**(3) 425–488.

Feng, Mingbin, Jeremy Staum. 2015. Green simulation designs for repeated experiments. L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, M. D. Rossetti, eds., *Proceedings of the 2015 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Inc., Piscataway, New Jersey, 403–413.

Fonseca, DJ, DO Navaresse, GP Moynihan. 2003. Simulation metamodeling through artificial neural networks. *Engineering Applications of Artificial Intelligence* **16**(3) 177–183.

Glasserman, Paul, Xingbo Xu. 2014. Robust risk measurement and model risk. *Quantitative Finance* **14**(1) 29–58.

Hesterberg, Tim. 1995. Weighted average importance sampling and defensive mixture distributions. *Technometrics* **37**(2) 185–194.

Hesterberg, Timothy Classen. 1988. Advances in importance sampling. Ph.D. thesis, Stanford University.

James, William, Charles Stein. 1961. Estimation with quadratic loss. *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, vol. 1. 361–379.

Kleijnen, Jack PC. 1998. Experimental design for sensitivity analysis, optimization, and validation of simulation models. *Handbook of simulation* 173–223.

Kleijnen, Jack P.C. 2009. Kriging metamodeling in simulation: A review. *European Journal of Operational Research* **192**(3) 707–716.

Kleijnen, Jack P.C., Reuven Y. Rubinstein. 1996. Optimization and sensitivity analysis of computer simulation models by the score function method. *European Journal of Operational Research* **88** 413–427.

Kleijnen, Jack PC, Wim CM Van Beers. 2004. Application-driven sequential designs for simulation experiments: Kriging metamodelling. *Journal of the Operational Research Society* **55**(8) 876–883.

Klugman, Stuart A., Harry H. Panjer, Gordon E. Willmot. 2012. *Loss Models: From Data to Decisions*. 4th ed. John Wiley & Sons.

Laforgia, Andrea. 1991. Bounds for modified Bessel functions. *Journal of Computational and Applied Mathematics* **34**(3) 263–267.

Liu, Ming, Barry L Nelson, Jeremy Staum. 2010. An efficient simulation procedure for point estimation of expected shortfall. *Simulation Conference (WSC), Proceedings of the 2010 Winter*. IEEE, 2821–2831.

Liu, Ming, Jeremy Staum. 2010. Stochastic kriging for efficient nested simulation of expected shortfall. *Journal of Risk* **12**(3) 3.

Luke, Yudell L. 1972. Inequalities for generalized hypergeometric functions. *Journal of Approximation Theory* **5**(1) 41–65.

Maggiar, Alvaro, Andreas Wächter, Irina S. Dolinskaya, Jeremy Staum. 2015. A derivative-free algorithm for the optimization of functions smoothed via Gaussian convolution using multiple importance sampling. *Optimization Online* Last accessed on Nov 11, 2015.

Martino, Luca, Victor Elvira, David Luengo, Jukka Corander. 2014. An adaptive population importance sampler: Learning from uncertainty. *IEEE Transactions on Signal Processing* **63**(16) 4422 – 4437.

McNeil, Alexander J., Rüdiger Frey, Paul Embrechts. 2005. *Quantitative Risk Management: Concepts, Techniques, and Tools*. Princeton University Press, Princeton, NJ, USA.

Merton, Robert C. 1974. On the pricing of corporate debt: The risk structure of interest rates. *The Journal of Finance* **29**(2) 449–470.

Meyn, Sean, Richard L. Tweedie. 2009. *Markov Chains and Stochastic Stability*. 2nd ed. Cambridge University Press, New York, NY, USA.

Munich Re Geo Risks Research. 2015. Loss events worldwide 1980 - 2014, 10 costliest events ordered by overall losses. Tech. rep., Munich Re.

Niederreiter, Harald. 1992. *Random Number Generation and Quasi-Monte Carlo Methods*, vol. 63. Society for Industrial and Applied Mathematics.

Nummelin, Esa. 2004. *General Irreducible Markov Chains and Non-Negative Operators*, *Cambridge Tracts in Mathematics*, vol. 83. Cambridge University Press.

Owen, Art B., Yi Zhou. 2000. Safe and effective importance sampling. *Journal of the American Statistical Association* **95**(449) 135–143.

Pasupathy, Raghu, Bruce W Schmeiser, Michael R Taaffe, Jin Wang. 2012. Control-variate estimation using estimated control means. *IIE Transactions* **44**(5) 381–385.

Rosenbaum, Imry, Jeremy Staum. 2015. Database Monte Carlo for simulation on demand. L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, M. D. Rossetti, eds., *Proceedings of the 2015 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Inc., Piscataway, New Jersey, 679–688.

Rubinstein, Reuven Y., Alexander Shapiro. 1993. *Discrete event systems: sensitivity analysis and stochastic optimization by the score function method*. Wiley.

Salemi, Peter, Jeremy Staum, Barry L Nelson. 2013. Generalized integrated brownian fields for simulation metamodeling. A. Tolk R. Hill R. Pasupathy, S.-H. Kim, M. E. Kuhl, eds., *Proceedings of the 2013 Winter Simulation Conference: Simulation: Making*

Decisions in a Complex World. Institute of Electrical and Electronics Engineers, Inc., 543–554.

Staum, Jeremy. 2009. Better simulation metamodeling: The why, what, and how of stochastic kriging. *Proceedings of the 2009 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Inc., Piscataway, New Jersey, 119–133.

Stein, Charles, et al. 1956. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. *Proceedings of the Third Berkeley symposium on mathematical statistics and probability*, vol. 1. 197–206.

Tongarlak, Mustafa H, Bruce E Ankenman, Barry L Nelson. 2011. Relative error stochastic kriging. *Proceedings of the Winter Simulation Conference*. Winter Simulation Conference, 3628–3640.

Veach, Eric. 1997. Robust Monte Carlo methods for light transport simulation. Ph.D. thesis, Stanford University.

Veach, Eric, Leonidas J Guibas. 1995. Optimally combining sampling techniques for Monte Carlo rendering. *SIGGRAPH 95 Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. ACM, New York, NY, USA, 419–428.

Xie, Wei, Barry L. Nelson, Russell R. Barton. 2014. A Bayesian framework for quantifying uncertainty in stochastic simulation. *Operations Research* **62**(6) 1439–1452.

APPENDIX A

Verifying the Conditions of Theorem 2.4.1 for the Catastrophe Bond Pricing Example

In this appendix, we verify the conditions of Theorem 2.4.1 and for the catastrophe bond (CAT bond) pricing example. In this example, the underlying state process $\{X_n : n = 1, 2, \ldots\}$ is AR(1), so it is ergodic. Recall that the loss in the CAT bond pricing example is given by $Y_n = \sum_{i=1}^{M_n} Z_n^i$. Conditional on $X_n = x$, and denoting $\phi(x) = (\lambda, \theta)$, the number M_n of claims is Poisson distributed with mean λ and independent of $\{Z_n^i : i = 1, \ldots, M_n\}$, which are independent random variables, exponentially distributed with mean θ. The conditional distribution of Y_n given X_n places probability mass $h(0; x) = e^{-\lambda}$ on $y = 0$ and has probability density

$$h(y; x) = \sum_{m=1}^{\infty} \frac{\lambda^m}{m!} e^{-\lambda} \frac{y^{m-1} e^{-y/\theta}}{\Gamma(m)\theta^m} = \sqrt{\frac{\lambda}{\theta y}} e^{-\lambda - y/\theta} I_1\left(2\sqrt{\frac{\lambda y}{\theta}}\right)$$

for $y > 0$, where Γ is the Gamma function and I_1 is the modified Bessel function of the first kind of order 1.

We will first establish two lemmas that are useful for verifying the conditions of Theorem 2.4.1 for the CAT bond pricing example. Define the domain $\mathcal{K} = (0, \infty) \times (0, \infty) \times \mathbb{R}$ and the functions $A : \mathcal{K} \mapsto \mathbb{R}$ and $a : \mathcal{K} \times (0, \infty) \mapsto \mathbb{R}$ by

$$A(k) = \int_0^{\infty} a(k, y)dy, \quad \text{where} \quad a(k, y) = \frac{e^{-k_1 y + k_3\sqrt{y}}}{\sqrt{y}} I_1(k_2\sqrt{y}) \geq 0. \tag{A.1}$$

Lemma A.0.1. *The function A is continuous on $\mathcal{K}$.*

Proof. For any $k, k' \in \mathcal{K}$, we have

$$|A(k) - A(k')| \leq \int_N^\infty a(k,y)dy + \int_N^\infty a(k',y)dy + \int_0^N |a(k,y) - a(k',y)|\, dy.$$

For any $k \in \mathcal{K}$ and $\epsilon > 0$, we will show that there exist $N > 0$ and $\delta > 0$ such that the right side is bounded above by ϵ if $\|k - k'\|_2 < \delta$. First, we will show that, for any $k \in \mathcal{K}$ and $\epsilon > 0$, there exists $N_1 > 0$ such that $\int_{N_1}^\infty a(k,y)dy \leq \epsilon/3$. Applying the same argument to $k' \in \mathcal{K}$, there exists N_2 such that $\int_{N_2}^\infty a(k',y)dy \leq \epsilon/3$. We then let $N = \max\{N_1, N_2\}$. Finally, we show that for any $k \in \mathcal{K}$, $\epsilon > 0$, and $N > 0$, there exists some $\delta > 0$ such that $\int_0^N |a(k,y) - a(k',y)|\, dy \leq \epsilon/3$ if $\|k - k'\|_2 \leq \delta$.

First, it is proved by Luke (1972) that $\Gamma(\nu+1)(2/y)^\nu I_\nu(y) < \cosh(y)$ for $y > 0$ and $\nu > -1/2$. Taking $\nu = 1$ in this inequality, and observing that $\cosh(y) < e^y$, we have $I_1(y) < (y/2)e^y$. Let $\tilde{k}_1 = k_1/2 > 0$. Then

$$\begin{aligned} a(k,y) &= \frac{e^{-k_1 y + k_3\sqrt{y}}}{\sqrt{y}} I_1(k_2\sqrt{y}) \\ &< \frac{e^{-k_1 y + k_3\sqrt{y}}}{\sqrt{y}} \left(\frac{k_2\sqrt{y}}{2} e^{k_2\sqrt{y}}\right) \\ &\leq \frac{k_2 e^{(k_2+k_3)^2/(4\tilde{k}_1)}}{2} e^{-\tilde{k}_1 y} =: C e^{-\tilde{k}_1 y}, \end{aligned}$$

where the second inequality holds because $(k_2 + k_3)\sqrt{y} - \tilde{k}_1 y \leq (k_2+k_3)^2/(4\tilde{k}_1)$, and the constant $C > 0$ is defined for ease of notation. Therefore $\int_N^\infty a(K,y)dy \leq C\int_N^\infty e^{-\tilde{k}_1 y}dy = C(e^{-N\tilde{k}_1}/\tilde{k}_1)$. Take

$$N_1 = -\frac{\ln[\epsilon\tilde{k}_1/3C]}{\tilde{k}_1} = -\frac{2\ln[(\epsilon k_1)/(6C)]}{k_1}.$$

Then $\int_{N_1}^\infty a(K,y)dy \leq \epsilon/3$.

The function I_1 is a solution of Bessel's differential equation, so it is continuous on $(0,\infty)$. Consequently, the function $\tilde{a}:\mathcal{K}\times(0,\infty)\mapsto\mathbb{R}$ defined as $\tilde{a}(k,y):=e^{-k_1y+k_3\sqrt{y}}I_1(k_2\sqrt{y})=a(k,y)\sqrt{y}$ is continuous on $\mathcal{K}\times[0,\infty)$. Choose any $\delta_0>0$ and define the compact neighborhood $\mathcal{N}_k(\delta_0):=\{k'\in\mathcal{K}:\|k-k'\|_2\leq\delta_0\}$. The function $\tilde{a}$ is continuous on the compact set $\mathcal{N}_k(\delta_0)\times[0,N]$. Therefore, it is uniformly continuous on $\mathcal{N}_k(\delta_0)\times[0,N]$. Consequently, there exists $\delta\in(0,\delta_0]$ such that, for all $(k,y),(k',y')\in\mathcal{N}_k(\delta_0)\times[0,N]$ that satisfy $\|(k,y)-(k',y')\|_2\leq\delta$, we have $|\tilde{a}(k,y)-\tilde{a}(k',y')|\leq\epsilon/(6\sqrt{N})$. Therefore, for any $k'\in\mathcal{K}$ such that $\|k-k'\|_2\leq\delta$, we have

$$\int_0^N|a(k,y)-a(k',y)|\,dy=\int_0^N\frac{1}{\sqrt{y}}|\tilde{a}(k,y)-\tilde{a}(k',y)|\,dy\leq\frac{\epsilon}{6\sqrt{N}}\int_0^N\frac{1}{\sqrt{y}}dy=\frac{\epsilon}{3}.$$

□

Define the domain $\mathcal{K}^B=(0,\infty)\times(0,\infty)\times(0,\infty)$ and the functions $B:\mathcal{K}^B\mapsto\mathbb{R}$ and $b:\mathcal{K}^B\times(0,\infty)\mapsto\mathbb{R}$ by

$$B(k)=\int_0^\infty b(k,y)dy,\quad\text{where}\quad b(k,y)=\frac{e^{-k_1y}}{\sqrt{y}}\frac{[I_1(k_2\sqrt{y})]^2}{I_1(k_3\sqrt{y})}\geq 0.$$

Lemma A.0.2. *If $\bar{\mathcal{K}}\subset\mathcal{K}$ is compact, then $\sup\{A(K)|K\in\bar{\mathcal{K}}\}<\infty$. If $\bar{\mathcal{K}}^B\subset\mathcal{K}^B$ is compact, then $\sup\{B(K)|K\in\bar{\mathcal{K}}^B\}<\infty$.*

Proof. Because A is continuous in $\mathcal{K}$, by Lemma A.0.1, and $\bar{\mathcal{K}}\subset\mathcal{K}$ is compact, it follows that $\sup\{A(K)|K\in\bar{\mathcal{K}}\}=\max\{A(K)|K\in\bar{\mathcal{K}}\}<\infty$.

For any $k \in \mathcal{K}^B$, let $k^* = \max\{k_2, k_3\}$ and $k_* = \min\{k_2, k_3\}$. Then it follows from Theorem 2.1 of Laforgia (1991) that

$$b(k_1, k_2, k_3, y) < \frac{e^{-k_1 y}}{\sqrt{y}} \left[e^{2(k^* - k_*)\sqrt{y}} \frac{k^*}{k_*} \right] I_1(k_2\sqrt{y}) = a(k_1, k_2, 2(k^* - k_*), y).$$

Therefore $B(k_1, k_2, k_3) \leq A(k_1, k_2, 2(k^* - k_*))$ for any $k \in \mathcal{K}^B$. Moreover, the compactness of $\bar{\mathcal{K}}^B$ implies the compactness of the set

$$\mathcal{K}^* := \{(k_1, k_2, 2(k^* - k_*)) | (k_1, k_2, k_3) \in \bar{\mathcal{K}}^B, k^* = \max\{k_2, k_3\}, k_* = \min\{k_2, k_3\}\},$$

which is a subset of $\mathcal{K}$. Therefore $\sup\{B(K) | K \in \bar{\mathcal{K}}^B\} \leq \sup\{A(K) | K \in \mathcal{K}^*\} < \infty$. $\square$

Proposition A.0.1. *In the catastrophe bond example, if* $\bar{\lambda} \geq \underline{\lambda} > 0$ *and* $2\underline{\theta} > \bar{\theta} \geq \underline{\theta} > 0$*, then for any target state* $x \in \mathbb{R}^2$*,* $\int_{\mathbb{R}^2} \sigma_x^2(x')\, d\pi(x') < \infty$ *and the sequence* $\{\sigma_x^2(X_n), n = 1, 2 \ldots\}$ *is uniformly integrable.*

Proof. Consider any target state $x \in \mathbb{R}^2$ and any sampling state $x' \in \mathbb{R}^2$. The likelihood ratio is $\ell_x(y; x') = h(y; x)/h(y; x')$. Because the simulation output $F(Y_n)$ is between 0 and 1, for all n, the target-x-sample-x' variance $\sigma_x^2(x')$ defined in Equation (2.5) satisfies

$$0 \leq \sigma_x^2(x') \leq \mathbb{E}\left[(F(Y_n)\, \ell_x(Y_n; x'))^2 | X_n = x' \right] \leq \mathbb{E}\left[(\ell_x(Y_n; x'))^2 | X_n = x' \right].$$

To establish the desired conclusions, it suffices to show that this conditional second moment has a finite upper bound over $x' \in \mathbb{R}^2$, for any fixed $x \in \mathbb{R}^2$. Denote $(\lambda, \theta) = \varphi(x) > 0$ and

$(\lambda', \theta') = \varphi(x')$. We have

$$\mathbb{E}\left[(\ell_x(Y_n; x'))^2 | X_n = x'\right] = \left(\frac{e^{-\lambda}}{e^{-\lambda'}}\right)^2 e^{-\lambda'} + \int_0^\infty \left(\frac{h(y;x)}{h(y;x')}\right)^2 h(y;x')dy.$$

The first term is bounded above by $e^{\bar{\lambda}-2\underline{\lambda}}$. For the second term, we have

$$\begin{aligned} \int_0^\infty \left(\tfrac{h(y;x)}{h(y;x')}\right)^2 h(y;x')dy &= \int_0^\infty \frac{\left[\sqrt{\frac{\lambda}{\theta y}} e^{-\lambda - \frac{y}{\theta}} I_1\left(2\sqrt{\frac{\lambda y}{\theta}}\right)\right]^2}{\sqrt{\frac{\lambda'}{\theta' y}} e^{-\lambda' - \frac{y}{\theta'}} I_1\left(2\sqrt{\frac{\lambda' y}{\theta'}}\right)} dy \\ &= \sqrt{\tfrac{\lambda^2\theta'}{\lambda'\theta^2}} e^{\lambda' - 2\lambda} \int_0^\infty \tfrac{1}{\sqrt{y}} e^{-\left(\frac{2\theta'-\theta}{\theta\theta'}\right)y} \frac{\left[I_1\left(2\sqrt{\frac{\lambda y}{\theta}}\right)\right]^2}{I_1\left(2\sqrt{\frac{\lambda' y}{\theta'}}\right)} dy \\ &= \sqrt{\tfrac{\lambda^2\theta'}{\lambda'\theta^2}} e^{\lambda' - 2\lambda} B\left(\tfrac{2\theta'-\theta}{\theta\theta'}, 2\sqrt{\tfrac{\lambda}{\theta}}, 2\sqrt{\tfrac{\lambda'}{\theta'}}\right). \end{aligned} \tag{A.2}$$

For all $x' \in \mathbb{R}^2$, we have that $(\lambda', \theta') = \varphi(x')$ is in a compact set $[\underline{\lambda}, \bar{\lambda}] \times [\underline{\theta}, \bar{\theta}]$ that does not contain zero. On this set, the arguments of B are all bounded so

$$\left\{ \left(\frac{2\theta' - \theta}{\theta\theta'}, 2\sqrt{\frac{\lambda}{\theta}}, 2\sqrt{\frac{\lambda'}{\theta'}}\right) | (\lambda', \theta') \in [\underline{\lambda}, \bar{\lambda}] \times [\underline{\theta}, \bar{\theta}] \right\} = \bar{\mathcal{K}}^B$$

is compact. Thus, it follows from the second claim of Lemma A.0.2 that Equation (A.2) has a finite upper bound over $x' \in \mathbb{R}^2$. □

APPENDIX B

Ineffective Output Reusage by Shrinkage Estimator

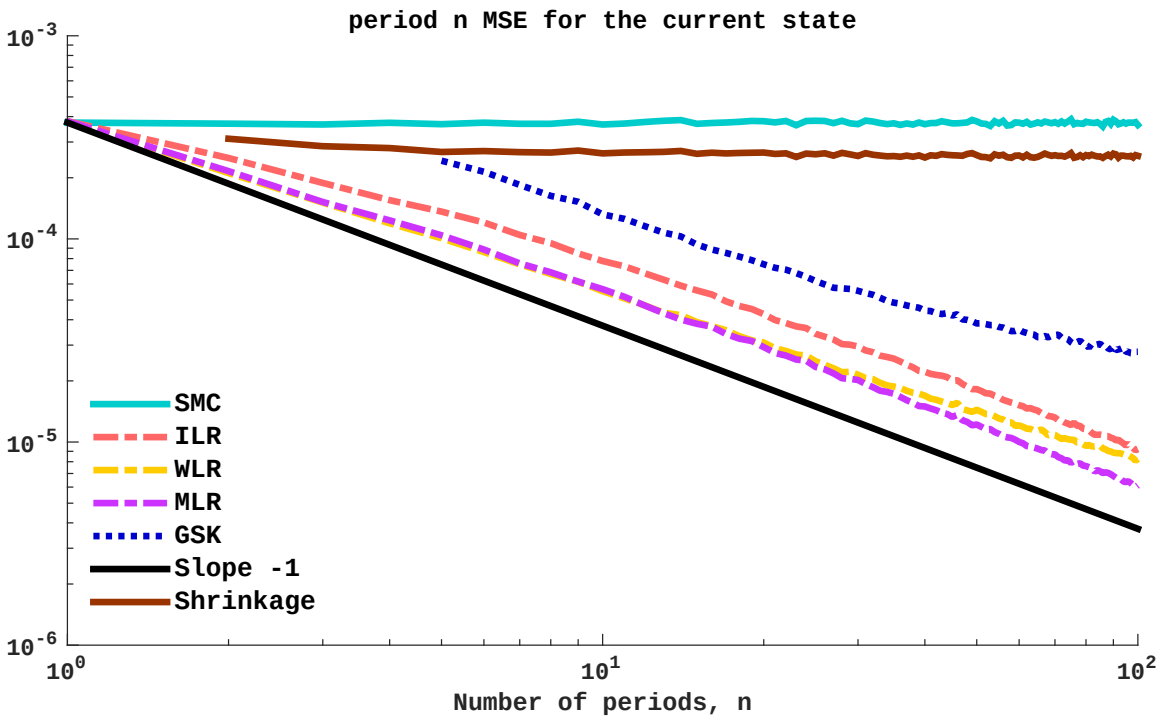


Figure B.1. Log-log plot of estimated variances for current-state estimators for CAT bond pricing example.

In this appendix we consider a simple shrinkage estimator with an empirical Bayes derivation and examine its practical performance in the repeated experiment setting. Shrinkage estimation is initially studied in Stein et al. (1956) and James and Stein (1961) then it was studied in a empirical Bayes framework in Efron and Morris (1973) and Efron and Morris (1975).

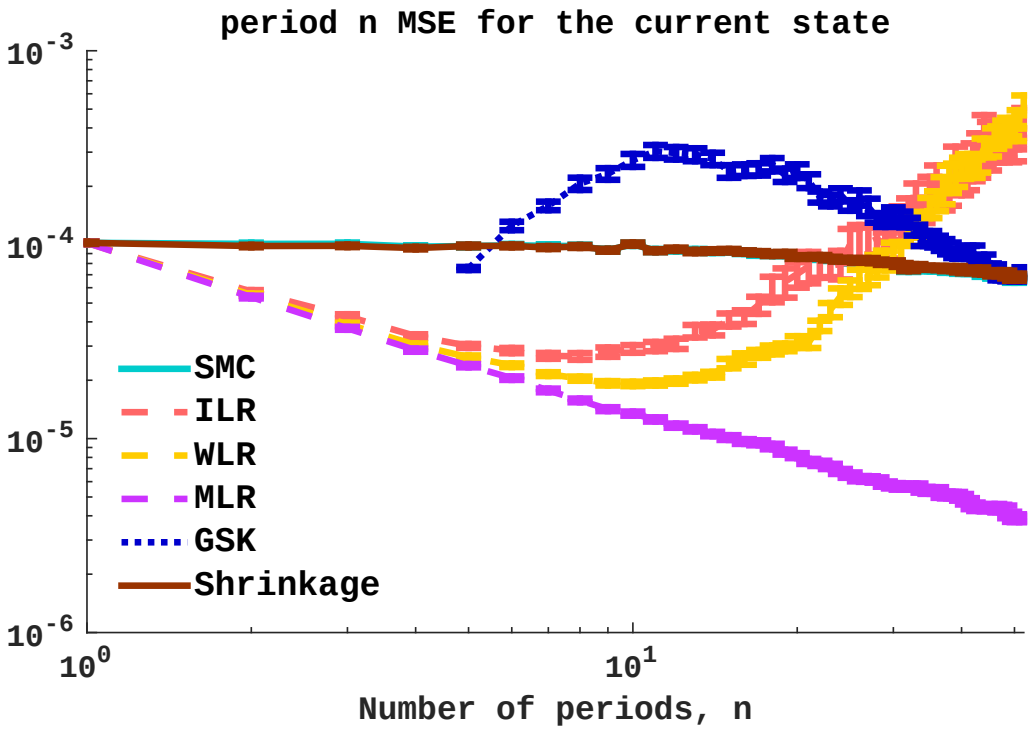


Figure B.2. Log-log plot with error bars for estimated MSEs for current state estimators for credit risk example.

In the empirical Bayes framework, the unknown quantities $\mu(X_1), \ldots, \mu(X_n)$ are assumed to be i.i.d. samples from a some prior distribution, whether parametric or not. If this prior distribution was known, then one can apply classical Bayesian estimation to derive the posterior distribution of $\mu(X_n)$ given the data, or in our setting the nth experiment outputs $\{F(Y_n^{(j)}) : j = 1, \cdots, r\}$. The essence of empirical Bayes is to learn the prior distribution directly from data, or from all simulation outputs $\{F(Y_k^{(j)}) : k = 1, \cdots, n; j = 1, \cdots, r\}$ in the repeated experiment setting then infer the posterior distribution of $\mu(X_n)$ using this learned prior and the nth experiment outputs $\{F(Y_n^{(j)}) : j = 1, \cdots, r\}$. Often the posterior mean in the parametric empirical Bayes framework turns out to be a weighted

average between the sample mean of the current simulation outputs (the local estimator) and the sample mean of all simulation outputs (the global estimator), so the local average is shrunk towards the global average hence the name "shrinkage estimator".

We thought that empirical Bayes and the resulting shrinkage estimator as described above reuses simulation outputs inefficiently. First of all, the i.i.d. assumptions of the expected performances $\mu(X_1), \ldots, \mu(X_n)$ does not take into account of their dependence due to the stochastic process $\{X_n\}$. Secondly, the global average may be different from the current expected performance, so the local average may be shrunk towards a wrong quantity. Lastly and most importantly, in the repeated experiment setting, the total number of simulation outputs increases over time but the number of simulation output for each experiment is fixed. Therefore, as more and more simulation outputs are reused, one has better estimate of the prior distribution so the empirical Bayes estimator will converge to a classical Bayes estimator. So as more simulation outputs are reused, the posterior variance of such empirical Bayes will converge to a fixed level but does not converge to zero. We will illustrate our points using a variant of the well-known James-Stein estimator, which can be viewed as the posterior mean of a hierarchical normal-normal model in a empirical Bayes framework.

Assume that the current expected performance follows a normal distribution with mean μ and variance σ^2, $\mu(X_n) \sim N(\mu, \sigma^2)$. Given the current state X_n, or equivalently given $\mu(X_n)$, assume that the sample average of the current simulation outputs is normally distributed with mean $\mu(X_n)$ and variance $\sigma^2(X_n)$, i.e., $\widehat{\mu}_r^{SMC}(X_n) = \frac{1}{r}\sum_{j=1}^{r} F(Y_n^{(j)}) \sim N(\mu(X_n), \sigma^2(X_n))$. Then in a classical Bayesian framework the posterior distribution of

$\mu(X_n)$ given the sample average $\widehat{\mu}_r^{SMC}(X_n)$ is

$$\mu(X_n)|\widehat{\mu}_r^{SMC}(X_n) \sim N\left(B\mu + (1-B)\widehat{\mu}_r^{SMC}(X_n), (1-B)\sigma^2\right), \quad B = \frac{\sigma^2}{\sigma^2 + \sigma^2(X_n)}.$$

Using the empirical Bayes ideas, we estimate the hyperparameters μ and σ^2 by the sample mean and sample variance of all simulation outputs $\{F(Y_k^{(j)}) : k = 1, \cdots, n; j = 1, \cdots, r\}$. In addition $\sigma^2(X_n)$ is estimated by the sample variance of the current simulation outputs $\{F(Y_n^{(j)}) : j = 1, \cdots, r\}$. The resulting shrinkage estimator is the empirical Bayes posterior mean, i.e.,

$$\mu_{n,r}^{Shrinkage}(X_n) = \widehat{B}\left(\frac{1}{nr}\sum_{k=1}^{n}\sum_{j=1}^{r}F(Y_k^{(j)})\right) + (1-\widehat{B})\left(\frac{1}{r}\sum_{j=1}^{r}F(Y_n^{(j)})\right), \quad \widehat{B} = \frac{\widehat{\sigma}^2}{\widehat{\sigma}^2 + \widehat{\sigma}^2(X_n)}$$

where $\widehat{\sigma}^2$ and $\widehat{\sigma}^2(X_n)$ are the global and local sample variances, respectively.

Figure 2.3 show the performance of the shrinkage estimator for fixed states in the CAT bond pricing example, as compared to the other estimators proposed in the new manuscript. While the other lines in the figure are identical to those in Figure 4 of the manuscript, the additional brown solid line shows the MSE of the shrinkage estimator. In this example, the global sample average is indicative of the average level of the expected performance $\mu(X)$ for $X \sim \pi(x)$ where $\pi(x)$ is the stationary distribution of the state process $\{X_n\}$. So the local estimator shrinks towards a correct global mean on average and hence the shrinkage estimator is somewhat better than the standard Monte Carlo estimator in terms of MSE. However, since the amount of local information remains fixed, i.e., r replications, the variance of the shrinkage estimator only converges to a fixed level. Therefore the MSE of the shrinkage estimator levels off over time, which was what we

expected. As shown in Figure 2.3, the shrinkage estimator is much worse than the other estimators proposed in the manuscript in terms of MSE.

Figure 2.4 show the performance of the shrinkage estimator for fixed states in the credit risk example. In this example, the performance of shrinkage estimator is almost distinguishable to that of the standard Monte Carlo estimator. This is because we are estimating a very small default probability and the variance of the local sample average is very small compared to the global sample variance. Consequently the shrinkage parameter $\widehat{B}$ is close to zero so the shrinkage estimator is close to the standard Monte Carlo estimator.

Both examples illustrates that shrinkage estimator is not appropriate for repeated experiments, therefor we decided not to consider it as a tool for reusing simulation outputs in green simulation.