Subsampling and other considerations for efficient risk estimation in large portfolios

Journal of Computational Finance **26**(1), 113–140
DOI: 10.21314/JCF.2022.019

Research Paper

Subsampling and other considerations for efficient risk estimation in large portfolios

Michael B. Giles[1] and Abdul-Lateef Haji-Ali[2]

[1]Mathematical Institute, Andrew Wiles Building, University of Oxford, Woodstock Road, Oxford OX2 6GG, UK; email: mike.giles@maths.ox.ac.uk

[2]Department of Actuarial Mathematics and Statistics, Heriot-Watt University, Edinburgh EH14 4AS, UK; email: a.hajiali@hw.ac.uk

(Received August 18, 2020; revised July 27, 2021; accepted March 14, 2022)

ABSTRACT

Computing risk measures of a financial portfolio comprising thousands of derivatives is a challenging problem both because it involves a nested expectation requiring multiple evaluations of the loss of the financial portfolio for different risk scenarios and because evaluating the loss of the portfolio is expensive and the cost increases with portfolio size. We apply multilevel Monte Carlo simulation with adaptive inner sampling to this problem and discuss several practical considerations. In particular, we discuss a subsampling strategy whose computational complexity does not increase with the size of the portfolio. We also discuss several control variates that significantly improve the efficiency of multilevel Monte Carlo in our setting.

Keywords: risk estimation; Monte Carlo simulation; nested simulation; multilevel Monte Carlo simulation; variance reduction; control variates.

1 INTRODUCTION

Various risk measures are computed to assess the risk of a financial portfolio. These measures include the probability of a large loss, value-at-risk (VaR) and conditional

Corresponding author: A.-L. Haji-Ali

Print ISSN 1460-1559 | Online ISSN 1755-2850

VaR (CVaR), also called expected shortfall. Computing these risk measures on a large portfolio usually involves two challenges: a nested expectation and a large sum. More precisely, consider computing the probability that the expected loss exceeds some given $\mathcal{K}_\eta \in \mathbb{R}$; ie, we want to compute

$$\eta := \mathbb{P}[\mathbb{E}[\Lambda \mid R_\tau] > \mathcal{K}_\eta] = \mathbb{E}[H(\mathbb{E}[\Lambda \mid R_\tau] - \mathcal{K}_\eta)], \tag{1.1}$$

where $\mathbb{E}[\Lambda \mid R_\tau]$ is the risk-neutral expected loss given some risk scenario R_τ at some short risk horizon τ and $H(\cdot)$ is the Heaviside function. For example, when considering market risk, the risk scenario includes the values of the underlying assets at some risk horizon τ that affect the loss incurred by the portfolio at maturity. The loss is usually an aggregate of many losses from different financial derivatives depending on a set of common underlying assets; ie,

$$\Lambda \equiv \frac{1}{P}\sum_{i=1}^{P} \Lambda_i, \tag{1.2}$$

where P is the total number of derivatives in the portfolio and Λ_i is the loss incurred by the ith derivative. The $1/P$ factor is a normalization factor that ensures boundedness as P increases. In realistic portfolios, the derivatives are heterogeneous in their evaluation. Some derivatives can be computed analytically, other derivatives have to be approximated by simulating the underlying assets and others still depend on assets that can only be sampled approximately. Moreover, the nominal values of these derivatives can vary greatly; a few derivatives might have large nominal values and thus contribute significantly to the total loss compared with the majority of derivatives.

A straightforward method to approximate the probability of a large expected loss is to simulate the nested expectation in (1.1) using Monte Carlo simulation. That is, M independent scenarios of the risk parameter R_τ are sampled and, for each risk scenario, N independent samples of the total loss Λ are sampled by evaluating the sum in (1.2). This method was explored by Gordy and Juneja (2010), who showed that the bias in the outer expectation is related to the variance of the estimator of the inner expectation (see also Giorgi *et al* (2017) for a sharper and extended analysis of the Gordy and Juneja results). Hence, when using N samples to estimate each inner expectation $\mathbb{E}[\Lambda_i \mid R_\tau]$, the bias in the outer estimator is $\mathcal{O}(N^{-1}P^{-1})$. Setting $N = \mathcal{O}(\max(1, \varepsilon^{-1}P^{-1}))$ and $M = \mathcal{O}(\varepsilon^{-2})$ to achieve a root mean square error (RMSE) ε, and since evaluating Λ is an $\mathcal{O}(P)$ operation, the total computational complexity is $\mathcal{O}(\max(P\varepsilon^{-2}, \varepsilon^{-3}))$. In addition, Gordy and Juneja (2010, Section 3.4) propose handling heterogeneous derivatives with different nominal values or a different computational cost in the portfolio by dividing the N samples proportionally among the different derivatives instead of evaluating the sum (see also Section 2.1).

In previous work (Giles and Haji-Ali 2019) we showed how to combine multilevel Monte Carlo (MLMC) simulation, as introduced by Giles (2008), with adaptive sampling, as introduced by Broadie *et al* (2011), to estimate quantities of the form $\mathbb{E}[H(\mathbb{E}[X \mid Y])]$ for two random variables X and Y. Using this strategy, for $Y \equiv R_\tau$ and $X \equiv \Lambda - \mathcal{K}_\eta$, the probability of a large expected loss can be estimated with a reduced computational complexity of $\mathcal{O}(\max(P\varepsilon^{-2}, \varepsilon^{-2}|\log\varepsilon|^2))$. This computational complexity is an improvement on that of the Monte Carlo simulation but it still suffers from the dependence on the number of derivatives, P, which, as mentioned earlier, can be significant for large portfolios.

The aim of this paper is twofold: to introduce random subsampling in the context of pricing derivatives or computing risk measures, and to show how several computational strategies can be combined in a unified framework for the efficient computation of risk measures in large financial portfolios. First, in Section 2 we discuss subsampling strategies to handle large sums of heterogeneous terms and present a method whose computational complexity does not depend on the number of terms in the sum. Then, in Section 3 we apply this method to our motivating problem involving a large portfolio, discuss several variance reduction techniques and show how to handle different computational models for $\mathbb{E}[\Lambda_i \mid R_\tau]$. In Section 4 we discuss how to apply MLMC and adaptive sampling to obtain a method whose computational complexity to achieve RMSE ε is $\mathcal{O}(\varepsilon^{-2}|\log\varepsilon|^2)$ and is independent of the number of derivatives. Finally, in Section 5, we apply our results to fictitious portfolios with heterogeneous derivatives to illustrate the benefit of the methods that are presented in the current work.

2 RANDOM SUBSAMPLING

In this section, we discuss unbiased methods to estimate an expectation involving a sum of terms $\{f_i\}_{i=1}^P$, for a large, fixed number of terms P:

$$\mathbb{E}\left[\frac{1}{P}\sum_{i=1}^{P} f_i\right]. \tag{2.1}$$

We focus on this generic problem in this section and later apply the discussed strategies to approximate the inner conditional expectation in (1.1) for a given risk scenario R_τ and discuss how to relate the terms $\{f_i\}_{i=1}^P$ to the losses $\{\Lambda_i\}_{i=1}^P$, depending on the computational model of $\mathbb{E}[\Lambda_i \mid R_\tau]$. We will initially assume that the terms $\{f_i\}_i$ are mutually independent (or, in the case of considering conditional expectation, conditionally independent) and discuss the general case later.

A naive Monte Carlo estimator of (2.1) with $N \geqslant 1$ samples of the sum requires a minimum budget equal to the cost to compute the sum once. The minimum budget

thus increases with the number of terms, P. Instead, we use a random subsampler based on the observation that

$$\frac{1}{P}\sum_{i=1}^{P}\mathbb{E}[f_i] = \mathbb{E}\left[\frac{f_j}{Pp_j}\right],$$

where j is a random integer with

$$\mathbb{P}[j = i] = \begin{cases} p_i & \text{for } i \in \{1,\dots,P\}, \\ 0 & \text{otherwise.} \end{cases}$$

Using N samples in a Monte Carlo estimator to estimate $\mathbb{E}[f_j/(Pp_j)]$ gives the resulting estimator

$$\frac{1}{NP}\sum_{n=1}^{N} f_{j^{(n)}}^{(n)} p_{j^{(n)}}^{-1},$$

where $j^{(n)}$ is the nth sample of the random integer j and $f_i^{(n)}$ is the nth sample of f_i. The variance of this estimator, which is equal to the mean squared error (MSE) since the estimator is unbiased, is

$$\begin{aligned}\mathrm{Var}\left[\frac{1}{NP}\sum_{n=1}^{N} f_{j^{(n)}}^{(n)} p_{j^{(n)}}^{-1}\right] &= \frac{1}{NP^2}\,\mathrm{Var}\left[\frac{f_j}{p_j}\right] \\ &= \frac{1}{NP^2}\left(\sum_{i=1}^{P} g_i^2 p_i^{-1} - \left(\sum_{i=1}^{P}\mathbb{E}[f_i]\right)^2\right),\end{aligned}$$

where $g_i^2 := \mathbb{E}[f_i^2]$. On the other hand, the expected total work is $N\sum_{i=1}^{P} p_i W_i$, where W_i is the work required to sample the term f_i. Minimizing the variance of the estimator subject to fixed expected total work leads to the choice $p_i \propto g_i/W_i^{1/2}$. By using an estimate of g_i, denoted by $\tilde{g}_i$, and imposing the constraint of the probabilities summing to 1, we set

$$p_i \equiv \frac{\tilde{g}_i/W_i^{1/2}}{\sum_{j=1}^{P}\tilde{g}_j/W_j^{1/2}}. \tag{2.2}$$

The work of this random subsampler is

$$N\frac{\sum_{i=1}^{P}\tilde{g}_i W_i^{1/2}}{\sum_{i=1}^{P}\tilde{g}_i/W_i^{1/2}}.$$

Assuming we have a total budget B to approximate (2.1), we set

$$N \equiv B\frac{\sum_{i=1}^{P}\tilde{g}_i/W_i^{1/2}}{\sum_{i=1}^{P}\tilde{g}_i W_i^{1/2}}.$$

Here, we ignore the restriction that $N \in \mathbb{Z}$, and treat N as a real number instead. Note that rounding the number of samples up increases the total computational cost by $\max_i W_i$. In any case, using the previous real value of N, the optimal variance can then be bounded as

$$\begin{aligned}\frac{1}{NP^2}\operatorname{Var}\left[\frac{f_j}{p_j}\right] &\leqslant \frac{1}{N}\left(\frac{1}{P}\sum_{i=1}^{P}\frac{g_i^2}{\tilde{g}_i}W_i^{1/2}\right)\left(\frac{1}{P}\sum_{i=1}^{P}\frac{\tilde{g}_i}{W_i^{1/2}}\right)\\ &\leqslant \frac{1}{B}\left(\frac{1}{P}\sum_{i=1}^{P}\frac{g_i^2}{\tilde{g}_i}W_i^{1/2}\right)\left(\frac{1}{P}\sum_{i=1}^{P}\tilde{g}_i W_i^{1/2}\right).\end{aligned} \tag{2.3}$$

If we further assume that $g_i \leqslant c\tilde{g}_i$ for some constant $c > 0$ and that

$$P^{-1}\sum_{i=1}^{P}\tilde{g}_i W_i^{1/2} \leqslant C$$

for some $C > 0$, then the variance of the estimator is $\mathcal{O}(B^{-1})$ and is independent of P, while the total cost of the estimator is B, up to the rounding of N. Under these same conditions, the previous discussion applies, even in the limit as $P \to \infty$. For finite P, we note that in the typical case, when for every $i \in \{1, \ldots, P\}$ we have that g_i and the work estimate W_i do not increase with P and $\tilde{g}_i$ is bounded from below, we can simply use $c \equiv \max_i(g_i/\tilde{g}_i)$ and $C \equiv \max_i \tilde{g}_i W_i^{1/2}$.

2.1 Mixed subsampling

Another way to handle heterogeneous terms is to use deterministic, stratified subsampling. This was explored in the current context of computing probabilities of a large loss by Gordy and Juneja (2010, Section 3.4). Applied to our setting, we write

$$\mathbb{E}\left[\frac{1}{P}\sum_{i=1}^{P}f_i\right] \approx \sum_{i=1}^{P}\frac{1}{PN_i}\sum_{n=1}^{N_i}f_i^{(n)}, \tag{2.4}$$

where $N_i \geqslant 1$ is the number of samples of the ith term. The variance of this unbiased estimator is

$$P^{-2}\sum_{i=1}^{P}\frac{\sigma_i^2}{N_i},$$

where $\sigma_i^2 := \operatorname{Var}[f_i]$, while the work is $\sum_{i=1}^{P} N_i W_i$. Similar to random subsampling, we minimize the variance subject to a budget constraint, B, to find the optimal number of samples for the ith term:

$$N_i \equiv B\frac{\tilde{\sigma}_i/W_i^{1/2}}{\sum_{j=1}^{P}\tilde{\sigma}_j W_j^{1/2}}, \tag{2.5}$$

assuming we have estimates of σ_i denoted by $\tilde{\sigma}_i$. Note that we again ignore the restriction that $N \in \mathbb{Z}$ and treat it as a real number. The optimal variance is bounded by

$$\frac{1}{B}\left(\frac{1}{P}\sum_{i=1}^{P}\frac{\sigma_i^2}{\tilde{\sigma}_i}W_i^{1/2}\right)\left(\frac{1}{P}\sum_{i=1}^{P}\tilde{\sigma}_i W_i^{1/2}\right), \tag{2.6}$$

assuming $N_i \geqslant 1$ for all i. If we further assume that $\sigma_i \leqslant c\tilde{\sigma}_i$ for some constant c and that

$$P^{-1}\sum_{i=1}^{P}\tilde{\sigma}_i W_i^{1/2} \leqslant C$$

for some $C > 0$, then the variance is $\mathcal{O}(B^{-1})$ and is independent of P and similar to random sampling. However, a crucial constraint is that the budget, B, must be sufficiently large so that $N_i \geqslant 1$ in (2.5) for all i, otherwise the estimator (2.4) is biased. In particular, the budget must be at least $\sum_{i=1}^{P} W_i$ to have at least one sample per term. This leads to a computational complexity that depends on the number of terms in the sum, unlike random subsampling. On the other hand, the variance of the stratified subsampler in (2.6) is always smaller than the variance of the random subsampler in (2.3). The variance reduction roughly scales with $P^{-1}\sum_{i=1}^{P}(\tilde{g}_i - \tilde{\sigma}_i)W_i^{1/2}$, which is bounded independently of P. In other words, in our setting, using random subsampling rather than stratified subsampling increases the error by a constant independent of P.

We can also combine random and stratified subsampling as follows:

$$\mathbb{E}\left[\frac{1}{P}\sum_{i=1}^{P} f_i\right] = \mathbb{E}\left[\frac{1}{P}\sum_{i=1}^{K} f_i\right] + \frac{1}{P}\mathbb{E}\left[\frac{f_j}{p_j}\right],$$

where

$$\mathbb{P}[j = i] = \begin{cases} p_i & \text{for } j \in \{K+1, \ldots, P\}, \\ 0 & \text{otherwise.} \end{cases}$$

Then the sum of the first K-terms is approximated using stratified subsampling, while the sum of the remaining $(P - K)$ terms is approximated using random subsampling. Compared with random subsampling, this new subsampler evidently leads to smaller variance for a fixed budget when the K-terms are themselves deterministic (ie, $\mathbb{E}[f_i] = f_i$ for $i \leqslant K$). In this case, evaluating the sum of the K-terms directly increases the work by $\sum_{i=1}^{K} W_i$ but decreases the variance by approximately $\sum_{i=1}^{K}\tilde{g}_i W_i^{1/2}$. Assuming the budget is larger than $\sum_{i=1}^{K} W_i$ and by picking those K-terms that have large $\tilde{g}_i / W_i^{1/2}$ (ie, large nominal value or small cost), we can ensure the increase in cost is small compared with the decrease in the error. To further illustrate this point, consider the case when $W_i = 1$ for all $i = 1, \ldots, P$ and

$\{f_i\}_{i=1}^P$ are all deterministic (ie, we are simply estimating the average $P^{-1}\sum_{i=1}^P f_i$ using a computational budget $B \leqslant P$); when $B \geqslant P$ we can compute the average directly. The mixed subsampler can then be written as

$$\frac{1}{P}\sum_{i=1}^{P} f_i \approx \frac{1}{P}\sum_{i=1}^{K} f_i + \frac{P-K}{P(B-K)}\sum_{n=1}^{B-K} f_{j(n)}$$

for $K \geqslant 0$ and where j is a random integer over $\{K+1,\ldots,P\}$. The variance is given by

$$\frac{(P-K)^2}{P^2(B-K)}\operatorname{Var}[f_j] \leqslant \Big(\max_i f_i^2\Big)\frac{(P-K)^2}{P^2(B-K)}.$$

The optimal value of K that minimizes the variance bound is $\min(0, 2B-P)$, and the corresponding variance is bounded by

$$\Big(\max_i f_i^2\Big)\begin{cases}\dfrac{4(P-B)}{P^2}, & \dfrac{P}{2} \leqslant B \leqslant P,\\[2ex] \dfrac{1}{B}, & 0 < B < \dfrac{P}{2}.\end{cases}$$

This is consistent with intuition: when the computational budget passes a certain threshold (in this case $P/2$), subsampling some terms deterministically leads to smaller variance for the same computational budget.

More generally, determining if a particular term f_i should be subsampled deterministically or randomly for a given budget B requires good estimates of both $\tilde{g}_i \approx g_i$ and $\tilde{\sigma}_i \approx \sigma_i$ (compare (2.3) and (2.6)) and hence of $\mathbb{E}[f_i]$, the quantity we are trying to estimate. If the optimal strategy is to subsample f_i deterministically instead of randomly, the variance reduction roughly scales with the difference, $(\tilde{g}_i - \tilde{\sigma}_i)W_i^{1/2}$. Considering the need for additional estimates, the optimization of the subsampling strategy for a term f_i is worthwhile only when the budget is sufficiently large compared with the number of terms P and we know that $(\tilde{g}_i - \tilde{\sigma}_i)W_i^{1/2}$ is large, which is maximal when f_i is deterministic. Hence, when considering a portfolio of terms, the variance reduction will be significant if the portfolio contains mostly deterministic terms or terms with small variability. In addition, using mixed subsampling complicates the analysis and precludes the application of other computational methods, such as using antithetic subsampling in MLMC (see Section 4.1). Based on these observations, and on several numerical experiments, we have found that mixed subsampling is not worthwhile in most practical cases, including the examples we consider in Section 5.

2.2 Dependent f_i

At the beginning of this section we assumed that $\{f_i\}_{i=1}^P$ are mutually independent. In real applications, including the ones we consider in this work, some of these terms

might depend on a set of common underlying random variables. Nevertheless, we can use independent samples of these underlying random variables when sampling f_i to get independent samples of f_i, and the previous discussion applies. Clearly, such resampling introduces an additional overhead since we have to resample the common underlying random variables.

On the other hand, this resampling has several advantages. Gordy and Juneja (2010, Section 3) argue that, in addition to simplifying analysis and implementation and making the parallelization of the sampler easier, resampling the common random factors is advisable to ensure that the Monte Carlo errors cancel out at the portfolio level. Another advantage is that this resampling allows us to optimize the number of samples per term based on estimates of the second moments or variance of $\{f_i\}_{i=1}^P$. Because of these advantages, we argue that resampling is the prudent choice in most situations. It should be noted, however, that terms that are known to be negatively correlated should be sampled together to reduce the overall variance and hence the computational cost. In Section 3 we will suggest additional strategies to reduce the variability of the loss variables Λ_i in certain settings.

3 PROBABILITY OF LOSS AS A NESTED EXPECTATION

In this section, we focus on our motivating problem of evaluating the probability of a large loss of a financial portfolio under market risk. We will focus on a model for the loss of a derivative that can be written as a difference between $V_{i,\tau}$, the discounted value of the derivative given the risk scenario R_τ at the risk horizon τ, and $V_{i,0}$, the risk-neutral discounted value at initial time; ie,

$$\begin{aligned}\mathbb{E}_{\mathbb{Q}}[\Lambda_i \mid R_\tau] &= V_{i,0} - V_{i,\tau}\\ &= \mathbb{E}_{\mathbb{Q}}[h_i(S)] - \mathbb{E}_{\mathbb{Q}}[h_i(S) \mid S(\tau) = R_\tau].\end{aligned}$$

Here, $\mathbb{Q}$ is the risk-neutral measure and h_i is the discounted payoff functional, which depends on the asset process, S. We will also assume that S is a stochastic process satisfying an Itô stochastic differential equation (SDE)

$$\mathrm{d}S(t) = a(t, S(t))\,\mathrm{d}t + b(t, S(t))\,\mathrm{d}B(t) \tag{3.1}$$

for some sufficiently smooth coefficients a and b and a Brownian process $\{B(t)\}_{t\geq 0}$. Recall that we are interested in computing

$$\begin{aligned}\eta &= \mathbb{P}[\mathbb{E}_{\mathbb{Q}}[\Lambda \mid R_\tau] > \mathcal{K}_\eta]\\ &= \mathbb{E}_{\mathbb{P}}[H(\mathbb{E}_{\mathbb{Q}}[\Lambda - \mathcal{K}_\eta \mid R_\tau])]\\ &= \mathbb{E}_{\mathbb{P}}\left[H\left(\mathbb{E}_{\mathbb{Q}}\left[\frac{1}{P}\sum_{i=1}^{P}\Lambda_i - \mathcal{K}_\eta \;\middle|\; R_\tau\right]\right)\right]\end{aligned}$$

for a given $\mathcal{K}_\eta \in \mathbb{R}$ with $\mathbb{Q}$ and $\mathbb{P}$ being the risk-neutral and physical measures, respectively. Since we consider the market risk, the risk parameter R_τ is the asset value $S(\tau)$ in the physical measure $\mathbb{P}$ at the risk horizon τ.

We will consider three common categories of computational models for $\mathbb{E}_\mathbb{Q}[\Lambda_i \mid R_\tau]$ and, for each computational model, we will discuss different strategies to reduce the variability of Λ_i, which in turn reduces the bias of a Monte Carlo estimator of η, as discussed in Section 1. At the end of this section, we will construct a "portfolio of terms", $\{f_i\}_{i=1}^P$, such that

$$\mathbb{E}_\mathbb{Q}\left[\frac{1}{P}\sum_{i=1}^{P}\Lambda_i \;\middle|\; R_\tau\right] = \mathbb{E}_\mathbb{Q}\left[\frac{1}{P}\sum_{i=1}^{P} f_i \;\middle|\; R_\tau\right]. \tag{3.2}$$

Then we can apply the subsampling strategies that were discussed in Section 2 when computing the inner expectation of the sum. Recall that when using a random subsampler to estimate the right-hand side of (3.2) the optimal probabilities depend on estimates of the work required to sample f_i and of $g_i^2 = \mathbb{E}_\mathbb{Q}[f_i^2 \mid R_\tau]$ for every $i \in \{1, 2, \ldots, P\}$ (ie, estimating g_i ultimately depends on the risk scenario). For an estimator of η that is based on sampling many risk scenarios this is clearly too costly, with a cost that grows with P, which is counter to our original objective of devising a method whose computational complexity does not depend on P. Instead, we propose to use estimates $\tilde{g}_i \approx g_i$ that do not depend on the risk scenario. For example, we may assign them to values that represent the relative importance of a derivative compared with the others, or we may assign $\tilde{g}_i = \mathbb{E}_\mathbb{Q}[f_i^2]$ for all i and all risk scenarios.

3.1 Exact, deterministic evaluation

For some derivatives, Λ_i might be deterministic when conditioned on the risk scenario R_τ, or we may be able to directly, with unit cost, compute $\mathbb{E}_\mathbb{Q}[\Lambda_i \mid R_\tau]$ exactly, or almost exactly, given the risk scenario R_τ. For example, when considering put or call options on assets that follow geometric Brownian processes, we may be able to solve the Black–Scholes partial differential equation (PDE) analytically or numerically with sufficient accuracy. Note that the Black–Scholes PDE needs to be solved only once to compute $\mathbb{E}_\mathbb{Q}[\Lambda_i \mid R_\tau]$ for all risk scenarios R_τ; hence, we may consider approximating the solution to the PDE as "offline work". In this case, we set $f_i \equiv \mathbb{E}_\mathbb{Q}[\Lambda_i \mid R_\tau]$ for a given R_τ. Note that, for a given risk scenario R_τ, f_i is deterministic with zero variance and the cost to compute it is $\mathcal{O}(1)$.

3.1.1 Delta control variate

Use of the Delta Greek to construct a control variate for the probability of large loss is well known (see Glasserman 2003; Gou 2016), and we recall the basic idea here.

Recall that the expected loss incurred by derivative i given a risk scenario R_τ is written as a difference (ie, $\mathbb{E}_\mathbb{Q}[\Lambda_i \mid R_\tau] \equiv V_{i,0} - V_{i,\tau}$). Then, using an Itô expansion yields

$$\mathbb{E}_\mathbb{Q}[\Lambda_i^2 \mid R_\tau] = ((R_0 - R_\tau)\nabla_{R_0} V_{i,0})^2 + \mathcal{O}(\tau^2),$$

where $R_0 \equiv S(0)$ and, for R_τ the price of the underlying asset, $\nabla_{R_0} V_{i,0}$ is the Delta Greek. The first term dominates in the above expression since the risk parameter is an Itô process (ie, $R_\tau \equiv S(\tau)$) yielding $\mathbb{E}_\mathbb{P}[|R_\tau - R_0|^2] = \mathcal{O}(\tau)$. By subtracting this term, we can define a new loss variable,

$$\hat{\Lambda}_i := \Lambda_i - (R_0 - R_\tau)\nabla_{R_0} V_{i,0},$$

for a given risk scenario R_τ and a new loss threshold, $\hat{\mathcal{K}}_\eta$, which depends on the risk scenario:

$$\hat{\mathcal{K}}_\eta := \mathcal{K}_\eta - (R_0 - R_\tau)\nabla_{R_0} V_0, \quad \text{where } \nabla_{R_0} V_0 = \frac{1}{P}\sum_{i=1}^{P} \nabla_{R_0} V_{i,0}, \tag{3.3}$$

so that

$$\mathbb{E}_\mathbb{Q}\left[\frac{1}{P}\sum_{i=1}^{P} \hat{\Lambda}_i - \hat{\mathcal{K}}_\eta \;\middle|\; R_\tau\right] = \mathbb{E}_\mathbb{Q}\left[\frac{1}{P}\sum_{i=1}^{P} \Lambda_i - \mathcal{K}_\eta \;\middle|\; R_\tau\right],$$

with $\mathbb{E}_\mathbb{Q}[\hat{\Lambda}_i^2 \mid R_\tau] = \mathcal{O}(\tau^2)$. Hence, we have the deterministic term $f_i \equiv \mathbb{E}_\mathbb{Q}[\hat{\Lambda}_i \mid R_\tau]$ with a second moment $\mathcal{O}(\tau^2) \ll \mathcal{O}(\tau)$ since $\tau \ll 1$. Note that $\nabla_{R_0} V_{i,0}$ is independent of the risk scenario, R_τ, for all i and can be computed once for all risk scenarios as "offline work". If the portfolio is delta-hedged, then $\nabla_{R_0} V_0 = 0$.

3.2 Exact simulation

In some settings, we might be able to sample Λ_i exactly for a given risk scenario R_τ but not to compute $\mathbb{E}_\mathbb{Q}[\Lambda_i \mid R_\tau]$ exactly. This is the case, for example, for exotic options or underlying assets involving high-dimensional Itô processes when we might still be able to solve the underlying SDEs analytically to sample Λ_i exactly for a given R_τ (eg, when the SDE solution is a geometric Brownian motion). In this case, we simply set $f_i \equiv \Lambda_i$. Note that, for a given risk scenario R_τ, the term f_i has nonzero variance and the cost to compute it is again $\mathcal{O}(1)$.

3.2.1 Reducing the variance of Λ_i

Denote by $S_{t,x}$ the solution of (3.1) given $S(t) = x$. Then we can write

$$\begin{aligned}\Lambda_i &= h_i(S_{0,R_0}) - h_i(S_{\tau,R_\tau}) \\ &= h_i(S_{\tau,S(\tau)}) - h_i(S_{\tau,R_\tau}).\end{aligned}$$

Hence, to sample Λ_i for a given risk scenario R_τ, we need to first sample $S(\tau)$, which requires sampling a Brownian path $\{B(t)\}_{0\leqslant t\leqslant\tau}$. Then, we sample $\{S_{\tau,S(\tau)}(t)\}_{t\geqslant\tau}$ and $\{S_{\tau,R_\tau}(t)\}_{t\geqslant\tau}$ starting from $S(\tau)$ and R_τ, respectively, which requires sampling one shared Brownian path $\{B(t)\}_{t\geqslant\tau}$. While we could use two independent Brownian paths to sample two independent paths $S^{(1)}_{\tau,S(\tau)}$ and $S^{(2)}_{\tau,R_\tau}$, this would yield a larger second moment. For example, when $h_i(S)\equiv h_i(S(T))$ for some maturity $T\gg\tau$ (ie, the payoff is a function of the asset value at maturity) and for a sufficiently smooth payoff functional h_i, we have

$$\mathbb{E}_{\mathbb{Q}}[\Lambda_i^2\mid R_\tau]=\mathbb{E}_{\mathbb{Q}}[|S(\tau)-R_\tau|^2\mid R_\tau]+\mathcal{O}(\mathbb{E}_{\mathbb{Q}}[|S^{(1)}_{\tau,R_\tau}(T)-S^{(2)}_{\tau,R_\tau}(T)|\mid R_\tau]).$$

Here, the second term dominates since $T\gg\tau$. Using a shared Brownian path in order to sample $\{S_{\tau,S(\tau)}(t)\}_{t\geqslant\tau}$ and $\{S_{\tau,R_\tau}(t)\}_{t\geqslant\tau}$, and for a sufficiently smooth payoff functional h_i, we write

$$\begin{aligned}\mathbb{E}_{\mathbb{Q}}[\Lambda_i^2\mid R_\tau]&\leqslant 2\mathbb{E}_{\mathbb{Q}}[(h_i(S_{0,R_0})-h_i(S_{0,R_\tau}))^2\mid R_\tau]\\&\quad+2\mathbb{E}_{\mathbb{Q}}[(h_i(S_{0,R_\tau})-h_i(S_{\tau,R_\tau}))^2\mid R_\tau]\\&=2\mathbb{E}_{\mathbb{Q}}[((R_0-R_\tau)\nabla_{R_0}h_i(S_{0,R_0}))^2\mid R_\tau]\\&\quad+\mathcal{O}(\mathbb{E}_{\mathbb{Q}}[|S_{0,R_\tau}-S_{\tau R_\tau}|^2\mid R_\tau])+\mathcal{O}(\tau^2),\end{aligned}$$

where S_{0,R_τ} is the solution of (3.1) given $S(0)=R_\tau$. Here, both $\mathbb{E}_{\mathbb{P}}[|R_\tau-R_0|^2]$ and $\mathbb{E}_{\mathbb{Q}}[|S_{0,R_\tau}-S_{\tau,R_\tau}|^2\mid R_\tau]$ are $\mathcal{O}(\tau)$. Hence, to reduce the variance of Λ_i, we will use control variates to eliminate the terms involving these factors, starting with the second term, where we use an antithetic variates approach. As a general methodology, this is a standard approach to variance reduction (Glasserman 2003), which has been used previously for pricing American options (Broadie *et al* 1997) and also for nested simulation within MLMC (Bujok *et al* 2013; Giles 2018; Giles and Haji-Ali 2019). However, the specific treatment used here for estimating portfolio losses does not appear to have been published previously. We denote by $S^+(\tau)$ and $S^-(\tau)$ the two antithetic Itô processes that start from $S^+(0)=S^-(0)=R_0$ and that depend on the Brownian paths $(B(t))_{0\leqslant t\leqslant\tau}$ and $(-B(t))_{0\leqslant t\leqslant\tau}$, respectively. Then we set

$$\hat{\Lambda}_i:=\tfrac{1}{2}(h_i(S_{\tau,S^+(\tau)})+h_i(S_{\tau,S^-(\tau)}))-h_i(S_{\tau,R_\tau}),\tag{3.4}$$

where all three processes S_{τ,R_τ}, $S_{\tau,S^+(\tau)}$ and $S_{\tau,S^-(\tau)}$ use the same Brownian path $\{B(t)\}_{t\geqslant\tau}$. Then, we have that

$$\mathbb{E}_{\mathbb{Q}}[\hat{\Lambda}_i\mid R_\tau]=\mathbb{E}_{\mathbb{Q}}[\Lambda_i\mid R_\tau].$$

Defining $S^+_{t,x}$ to be the solution of (3.1) given $S^+(t)=x$ and using the Brownian path $\{B(s)\}_{s\geqslant t}$, while $S^-_{t,x}$ uses the Brownian paths $\{-B(s)\}_{t\leqslant s\leqslant\tau}$ and

$\{B(s)\}_{\max(\tau,t)\leqslant s}$, for a sufficiently smooth payoff h_i, we then have

$$\mathbb{E}_{\mathbb{Q}}[(\hat{\Lambda}_i)^2 \mid R_\tau] = 2\mathbb{E}_{\mathbb{Q}}[(\tfrac{1}{2}(R_0 - R_\tau)(\nabla_{R_0} h_i(S^+_{0,R_0}) + \nabla_{R_0} h_i(S^-_{0,R_0})))^2 \mid R_\tau] + \mathcal{O}(\mathbb{E}_{\mathbb{Q}}[(\tfrac{1}{2}(S^+_{0,R_\tau} + S^-_{0,R_\tau}) - S_{\tau,R_\tau} \mid R_\tau)^2]) + \mathcal{O}(\tau^2).$$

Here, assuming the SDE coefficients are sufficiently smooth, the second term is now $\mathcal{O}(\tau^2) \ll \mathcal{O}(\tau)$ since τ is assumed small.

Finally, similarly to in Section 3.1, we can use the Delta control variate to eliminate the remaining $\mathcal{O}(\tau)$ term by defining

$$\hat{\hat{\Lambda}}_i := \hat{\Lambda}_i - \tfrac{1}{2}(R_0 - R_\tau) D_i, \quad D_i := \nabla_{R_0} h_i(S^+_{0,R_0}) + \nabla_{R_0} h_i(S^-_{0,R_0}), \tag{3.5}$$

where we assume here that $h_i(S)$ is differentiable with respect to the initial state R_0. We also modify the loss threshold, $\mathcal{K}_\eta$, as in (3.3) so that

$$\mathbb{E}_{\mathbb{Q}}\left[\frac{1}{P}\sum_{i=1}^{P} \hat{\hat{\Lambda}}_i - \hat{\mathcal{K}}_\eta \;\middle|\; R_\tau\right] = \mathbb{E}_{\mathbb{Q}}\left[\frac{1}{P}\sum_{i=1}^{P} \Lambda_i - \mathcal{K}_\eta \;\middle|\; R_\tau\right],$$

since

$$\nabla_{R_0} V_{i,0} = \mathbb{E}_{\mathbb{Q}}[\nabla_{R_0} h_i(S^+_{0,R_0})] = \mathbb{E}_{\mathbb{Q}}[\nabla_{R_0} h_i(S^-_{0,R_0})].$$

Recall that $\nabla_{R_0} V_{i,0}$ is independent of the risk scenario R_τ for all i and can be computed once for all risk scenarios as "offline work". In summary, to sample $\hat{\hat{\Lambda}}_i$, we use all the variance reduction techniques discussed above:

(a) the delta control variate,

(b) the antithetic pair $S^+(\tau)$ and $S^-(\tau)$ and

(c) the same Brownian path $\{B(t)\}_{t\geqslant\tau}$ when simulating S_{τ,R_τ}, $S_{\tau,S^+(\tau)}$ and $S_{\tau,S^-(\tau)}$.

Indeed, all three variance reduction techniques ensure that

$$\mathbb{E}_{\mathbb{Q}}[(\hat{\hat{\Lambda}}_i)^2 \mid R_\tau] = \mathcal{O}(\tau^2) \ll \mathbb{E}_{\mathbb{Q}}[\Lambda_i^2 \mid R_\tau] = \mathcal{O}(\tau)$$

for $\tau \ll T$.

3.3 Approximate simulation

More generally, for some derivatives we might only be able to sample Λ_i approximately for a given risk scenario R_τ. This is the case, for example, if (3.1) cannot be solved analytically and we have to use a numerical scheme to approximate samples of process S and then compute the loss to obtain an approximate sample of Λ_i. The

cost for each approximate sample of Λ_i increases as the approximation error, and consequently the bias when estimating $\mathbb{E}_{\mathbb{Q}}[\Lambda_i \mid R_\tau]$, decreases.

Nevertheless, using unbiased MLMC (Rhee *et al* 2015), we can in certain cases obtain an unbiased Monte Carlo estimator of $\mathbb{E}_{\mathbb{Q}}[\Lambda_i \mid R_\tau]$ by using samples whose expected cost is $\mathcal{O}(1)$. To briefly present unbiased MLMC here, we denote by $\Lambda_{i,l}$ the lth approximation level of Λ_i (eg, using 4^l time steps in a Milstein scheme to approximate the samples of the solution of (3.1)).[1] Then define

$$\Delta\Lambda_{i,l} := \Lambda_{i,l} - \Lambda_{i,l-1}, \tag{3.6}$$

with $\Lambda_{i,-1} = 0$. As in standard MLMC (Giles 2015), we assume that the cost of computing $\Delta\Lambda_{i,l}$ grows proportionally to $4^{\gamma l}$, while its first and second moments satisfy

$$|\mathbb{E}_{\mathbb{Q}}[\Delta\Lambda_{i,l} \mid R_\tau]| = \mathcal{O}(4^{-\alpha l}) \quad \text{and} \quad \mathbb{E}_{\mathbb{Q}}[(\Delta\Lambda_{i,l})^2 \mid R_\tau] = \mathcal{O}(4^{-\beta l}),$$

respectively, for $\alpha, \beta, \gamma > 0$. Then, we write

$$\mathbb{E}_{\mathbb{Q}}[\Lambda_i \mid R_\tau] = \sum_{l=0}^{\infty} \mathbb{E}_{\mathbb{Q}}[\Delta\Lambda_{i,l} \mid R_\tau] = \mathbb{E}_{\mathbb{Q}}[C_\zeta 4^{\zeta l}\Delta\Lambda_{i,l} \mid R_\tau], \tag{3.7}$$

where on the right-hand side, with a slight abuse of notation, l is a random integer satisfying $\mathbb{P}[l = j] = 4^{-\zeta j}/C_\zeta$, where $j \in \{0, 1, 2, \dots\}$, $\zeta > 0$ and $C_\zeta := 1/(1-4^{-\zeta})$ is a normalization constant. In other words, just like the random subsampling method introduced in Section 2, unbiased MLMC is based on randomly subsampling the corrections $\Delta\Lambda_{i,l}$ to compute the infinite sum in (3.7). The analysis of unbiased MLMC is also similar to that shown in Section 2. In this setting, the condition $\gamma < \zeta < \beta \leqslant 2\alpha$ is sufficient (Rhee *et al* 2015) to bound the expected cost and variance of $C_\zeta 4^{\zeta l}\Delta\Lambda_{i,l}$, for random l as above, and hence we can estimate $\mathbb{E}_{\mathbb{Q}}[\Lambda_i \mid R_\tau]$ without bias by using standard Monte Carlo simulation to estimate $\mathbb{E}_{\mathbb{Q}}[C_\zeta 4^{\zeta l}\Delta\Lambda_{i,l} \mid R_\tau]$. The optimal value for ζ, obtained by minimizing the RMSE for a given computational budget, is $(\beta+\gamma)/2$. As an example, if $h_i(S) \equiv h_i(S(T))$ for some maturity $T > 0$ (ie, the payoff is a function of the asset value at maturity), then if h_i is Lipschitz and a Milstein scheme is used to approximate samples of the solution of (3.1), we have $\beta = 2\alpha = 2\gamma$ (Giles *et al* 2019). On the other hand, if h_i is discontinuous, then we can show that $\beta = \gamma - \nu$ for any $\nu > 0$ using a similar analysis to Giles *et al* (2009, Section 3). In this case, since $\beta \leqslant \gamma$ we would need

[1] The same discussion applies if m^l time steps are used for the lth approximation level (for any $m > 1$). The choice of $m = 4$ is motivated by the fact that, when the variance of $\Delta\Lambda_{i,l}$ decreases with m^{-2l} while its cost increases with m^l, as we later assume, it minimizes the total cost of an MLMC estimator (see Haji-Ali *et al* 2015).

to truncate the sum of corrections in (3.7) at some maximum level L to ensure that unbiased MLMC has finite work, which introduces a bias of $\mathcal{O}(4^{-\alpha L})$. A modified unbiased MLMC estimator (Rhee *et al* 2015, Section 4) can then be constructed with samples that have bounded variance but with an expected cost that is $\mathcal{O}(4^{(\gamma-\beta)L})$ for $\beta < \gamma$ or $\mathcal{O}(L^2)$ for $\gamma = \beta$. Throughout the current work, we will assume $\beta > \gamma$. In the previous example with a discontinuous h_i, an estimator based on conditional expectation can be used to ensure faster variance convergence (Giles *et al* 2019, Section 3.2.8).

In summary, in the case of approximate simulation we take $f_i \equiv C_\zeta 4^{\zeta l} \Delta\Lambda_{i,l}$, where l is a random index. In this case, for a given risk scenario R_τ, the term f_i has nonzero variance and the expected cost to compute it is $\mathcal{O}(1)$, since we assume $\beta > \gamma$.

REMARK 3.1 (Moments of unbiased estimator) For the $\beta > \gamma$ case, where we do not have to truncate the sum in (3.7) and we have an unbiased estimator of $\mathbb{E}_\mathbb{Q}[\Lambda_i \mid R_\tau]$, assume further that $\mathbb{E}_\mathbb{Q}[|\Delta\Lambda_{i,l}|^q \mid R_\tau] = \mathcal{O}(4^{-q\beta l/2})$ for some $q > 2$. The q-moment of the unbiased estimator is then

$$\begin{aligned}\mathbb{E}_\mathbb{Q}[|C_\zeta 4^{\zeta l}\Delta\Lambda_{i,l}|^q \mid R_\tau] &= C_\zeta^q \sum_{l=0}^{\infty} 4^{\zeta(q-1)l}\mathbb{E}_\mathbb{Q}[|\Delta\Lambda_{i,l}|^q \mid R_\tau] \\ &= \mathcal{O}\left(\sum_{l=0}^{\infty} 4^{-q\beta l/2+\zeta(q-1)l}\right).\end{aligned}$$

Hence, even if the q-moment of $\Delta\Lambda_{i,l}$ is finite for a given level l, the q-moment of $4^{\zeta l}\Delta\Lambda_{i,l}$, where l is a random level, is finite only when $q < (1-\beta/(2\zeta))^{-1}$. For example, when $\zeta = (\beta+\gamma)/2$, the q-moment of the unbiased estimator is finite for $q < 1 + \beta/\gamma$. In other words, if we require certain finite q-moments of the unbiased estimator, for example, when using MLMC with adaptive sampling (see Section 4.1), we might have to use a smaller, suboptimal value of ζ.

3.3.1 Control variates

The discussion on control variates in Section 3.2 carries over to the case of approximate simulation. Seen another way, we assume we can sample $\hat{\hat{\Lambda}}_i$ in (3.5) approximately along with the modified loss threshold $\hat{\mathcal{K}}_\eta$ in (3.3). Then, denoting the lth approximation level by $\hat{\hat{\Lambda}}_{i,l}$ and defining $\Delta\hat{\hat{\Lambda}}_{i,l}$ as in (3.6), we set $f_i \equiv C_\zeta 4^{\zeta l}\Delta\hat{\hat{\Lambda}}_{i,l}$.

One important observation to make here is that, depending on the payoff function h_i, we might have the case where $\mathrm{Var}[\Delta\hat{\hat{\Lambda}}_{i,l} \mid R_\tau] > \mathrm{Var}[\Delta\hat{\Lambda}_{i,l} \mid R_\tau]$ for some l, where $\Delta\hat{\Lambda}_{i,l}$ and $\hat{\Lambda}_{i,l}$ are defined as above for $\hat{\Lambda}$ in (3.4). In other words, using the Delta control variate leads to a larger variance for some approximation levels. As an example, consider $h_i(S) = h_i(S(T))$, where h_i is Lipschitz but $\nabla_{R_0} h_i$ is

discontinuous, and assume that we use the Milstein scheme to approximate (3.1) with 4^l time steps. Then, denote by $D_{i,l}$ the lth approximation level of D_i in (3.5) and define $\Delta D_{i,l}$ as in (3.6) and write

$$\Delta\hat{\hat{\Lambda}}_{i,l} := \Delta\hat{\Lambda}_{i,l} - \tfrac{1}{2}(R_\tau - R_0)\Delta D_{i,l}.$$

We see that, while $\mathrm{Var}[\Delta\hat{\Lambda}_{i,l} \mid R_\tau] = \mathcal{O}(4^{-2l})$, we have

$$\mathrm{Var}[\Delta D_{i,l} \mid R_\tau] = \mathcal{O}(4^{l(\nu-1)}) \quad \text{for any } \nu > 0,$$

again using a similar analysis to that in Giles *et al* (2009, Section 3). Hence, for sufficiently large l we have that $\mathrm{Var}[\Delta\hat{\Lambda}_{i,l} \mid R_\tau] < \mathrm{Var}[\Delta\hat{\hat{\Lambda}}_{i,l} \mid R_\tau]$. In other words, applying the Delta control variate beyond a certain level l might lead to an estimator with a larger variance, unless the payoff h_i is sufficiently smooth (in this example requiring $\nabla_{R_0} h_i$ to be Lipschitz). An alternative is to use a modified Milstein scheme for the Delta control variate (Giles *et al* 2019, Section 3.2.8), so that the variance $\mathrm{Var}[\Delta D_{i,l} \mid R_\tau]$ is sufficiently small compared with, or of the same order as, $\mathrm{Var}[\Delta\hat{\Lambda}_{i,l} \mid R_\tau]$.

If h_i is not sufficiently smooth, then we may apply the Delta control variate only up to some level (eg, $l = 0$). That is, we define

$$\hat{\Delta}\hat{\Lambda}_{i,l} := \begin{cases} \hat{\hat{\Lambda}}_{i,l}, & l = 0, \\ \Delta\hat{\Lambda}_{i,l}, & \text{otherwise}, \end{cases}$$

and set $f_i \equiv \hat{\Delta}\hat{\Lambda}_{i,l}$. In this case, the modification to the threshold value should also be approximated at $l = 0$. That is, we define the new loss threshold by

$$\hat{\hat{\mathcal{K}}}_\eta := \mathcal{K}_\eta + \tfrac{1}{2}(R_\tau - R_0)\mathbb{E}_{\mathbb{Q}}[D_{i,0} \mid R_\tau],$$

so that

$$\mathbb{E}_{\mathbb{Q}}\left[\frac{1}{P}\sum_{i=1}^{P}\hat{\Delta}\hat{\Lambda}_{i,l} - \hat{\hat{\mathcal{K}}}_\eta \;\middle|\; R_\tau\right] = \mathbb{E}_{\mathbb{Q}}\left[\frac{1}{P}\sum_{i=1}^{P}\Lambda_i - \mathcal{K}_\eta \;\middle|\; R_\tau\right].$$

Finally, since the Delta control variate reduces the variance of the first level only, we should ensure that the variance at level $l = 1$ (ie, $\mathrm{Var}[\hat{\Delta}\hat{\Lambda}_{i,1} \mid R_\tau]$) is sufficiently smaller than the variance at level $l = 0$ (ie, $\mathrm{Var}[\hat{\hat{\Lambda}}_{i,0} \mid R_\tau]$), otherwise refining the first level of approximation of (3.1) leads to a smaller overall root mean square value (see the discussion in Giles and Haji-Ali (2019, Section 3) and Section 4.2 for more details).

4 MULTILEVEL MONTE CARLO AND ADAPTIVE SAMPLING

The outcomes of Section 3 are the terms $\{f_i\}_{i=1}^P$ and a new loss threshold $\hat{\hat{\mathcal{K}}}_\eta$ that depends on the risk scenario R_τ, such that we can write

$$\eta = \mathbb{P}\left[\mathbb{E}_{\mathbb{Q}}\left[\frac{1}{P}\sum_{i=1}^{P}\Lambda_i \;\middle|\; R_\tau\right] > \mathcal{K}_\eta\right] = \mathbb{E}_{\mathbb{P}}\left[H\left(\mathbb{E}_{\mathbb{Q}}\left[\frac{f_j}{Pp_j} - \hat{\hat{\mathcal{K}}}_\eta | R_\tau\right]\right)\right],$$

where j is a random integer satisfying $\mathbb{P}[j = i] = p_i$ for $i \in \{1, 2, \ldots, P\}$. In this section, for notational convenience we will drop the measures $\mathbb{P}$ and $\mathbb{Q}$ and define the random variables $Y := R_\tau$ and $X := f_j/(Pp_j) - \hat{\hat{\mathcal{K}}}_\eta$, such that our aim now is simply to compute $\mathbb{E}[H(\mathbb{E}[X \mid Y])]$. Then, we will discuss using MLMC with adaptive inner sampling, as we previously proposed in Giles and Haji-Ali (2019). We start by defining

$$\hat{E}_\ell(y) := \frac{1}{N_\ell}\sum_{n=1}^{N_\ell} X^{(n)}(y), \tag{4.1}$$

which is a Monte Carlo estimator of $\mathbb{E}[X \mid Y]$ using N_ℓ samples. Here, $X^{(n)}(y)$ denotes the nth sample of X conditioned on $Y = y$, and the number of samples N_ℓ may depend on y. Then the MLMC estimator for $\mathbb{E}[H(\mathbb{E}[X \mid Y])]$ is

$$\sum_{\ell=0}^{L}\frac{1}{M_\ell}\sum_{m=1}^{M_\ell}\Delta H_\ell(Y^{(\ell,m)}), \quad \text{where } \Delta H_\ell(y) = H(\hat{E}_\ell(y)) - H(\hat{E}_{\ell-1}(y))$$

and $\{Y^{(\ell,m)}\}_{\ell,m}$ are independent and identically distributed samples of Y. Moreover, we set $H(\hat{E}_{-1}(\cdot)) \equiv 0$. We can choose N_ℓ uniformly for all y (eg, $N_\ell = N_0 2^\ell$ for some $N_0 > 0$). In this case, it can be shown, under certain moment and smoothness conditions (Giles and Haji-Ali 2019; Giorgi *et al* 2017), that

$$|\mathbb{E}[\Delta H_\ell(Y)]| = \mathcal{O}(2^{-\ell}) \quad \text{and} \quad \mathrm{Var}[\Delta H_\ell(Y)] = \mathcal{O}(2^{-\ell/2}).$$

Assuming that the expected cost of evaluating X is $\mathcal{O}(1)$ and is independent of ℓ, the optimal complexity of MLMC needed to achieve an RMSE ε can then be shown to be $\mathcal{O}(\varepsilon^{-5/2-\nu})$ (Giles 2015, Theorem 2.1).

To reduce the computational complexity, we instead select N_ℓ adaptively based on samples of Y. Let

$$\delta := \frac{|\mathbb{E}[X \mid Y]|}{(\mathrm{Var}[X \mid Y])^{1/2}}$$

and let $\hat{\delta} \approx \delta$ be an estimate computed using Monte Carlo estimates of $\mathbb{E}[X \mid Y]$ and $\mathrm{Var}[X \mid Y]$ for a given Y. We then select N_ℓ using Algorithm 4.1, which is an

ALGORITHM 4.1 (Adaptive algorithm to determine N_ℓ)

Input: $\ell, y, N_0 > 1, C > 0, 1 < r < 2$
Output: $N_0 2^\ell \leqslant N_\ell \leqslant N_0 4^\ell$

```
set N_ℓ = N_0 2^ℓ
Set done:=false
repeat
  if 2N_ℓ ⩾ N_0 4^ℓ then
    Set N_ℓ ≡ N_0 4^ℓ
    Set done:=true
  else
    Generate N_ℓ new, and independent, inner samples of X given Y = y
    Estimate δ̂ ≈ δ
    if (4.2) is satisfied then
      Set done:=true
    else
      N_ℓ ≡ 2N_ℓ
    end if
  end if
until done
return N_ℓ
```

iterative algorithm that starts with a minimum number of samples $N_\ell = N_0 2^\ell$ for a given $Y = y$ and then, on every iteration, the number of samples is doubled until the inequality

$$N_\ell \geqslant N_0 4^\ell (C^{-1} N_0^{1/2} 2^\ell \hat{\delta})^{-r}, \tag{4.2}$$

for given constants $C > 0$ and $1 < r < 2$, is satisfied or the maximum number of samples $N_0 4^\ell$ is reached.

Algorithm 4.1 returns the minimum number of samples $N_0 2^\ell$ when $\hat{\delta}$ is sufficiently large, and hence a Monte Carlo estimate of $\mathbb{E}[X \mid Y]$ will likely have the correct sign, leading to an exact evaluation of $H(\cdot)$. When $\hat{\delta}$ is small, estimating the sign of $\mathbb{E}[X \mid Y]$ using a Monte Carlo estimator is more difficult and Algorithm 4.1 returns a larger number of samples, up to the maximum $N_0 4^\ell$, to account for that (see Giles and Haji-Ali (2019) for the motivation for the exact form of (4.2)).

More concretely, the analysis in Giles and Haji-Ali (2019, Theorem 2.7) proves the following two crucial properties:

$$\mathbb{E}[N_\ell] = \mathcal{O}(2^\ell) \quad \text{and} \quad \mathrm{Var}[\Delta H_\ell(Y)] = \mathcal{O}(2^{-\ell}), \tag{4.3}$$

where we assume the following mild conditions:

(1) δ has a probability density function ρ, and there exist positive constants ρ_0 and δ_0 such that $\rho(\delta) \leqslant \rho_0$ for all $\delta \geqslant \delta_0$;

(2) there exists $q > 2$ such that

$$\sup_y \mathbb{E}\left[\left(\frac{|X - \mathbb{E}[X \mid Y]|}{(\mathrm{Var}[X \mid Y])^{1/2}}\right)^q \;\middle|\; Y = y\right] < \infty;$$

and

(3) r is chosen such that

$$1 < r < 2 - \frac{(4q+1)^{1/2} - 1}{q}. \tag{4.4}$$

In addition, assuming that the expected cost of evaluating X is $\mathcal{O}(1)$ and is independent of ℓ guarantees that the optimal complexity of the MLMC method necessary to achieve an RMSE ε is $\mathcal{O}(\varepsilon^{-2}|\log \varepsilon|^2)$ (see Giles 2015; Giles and Haji-Ali 2019).

4.1 Antithetic sampling

Recall that, given a risk scenario Y, we need to sample both $\hat{E}_\ell(Y)$ and $\hat{E}_{\ell-1}(Y)$. Sampling $\hat{E}_\ell$ requires sampling N_ℓ independent and identically distributed samples of X given the risk scenario Y. Similarly, sampling $\hat{E}_{\ell-1}$ requires sampling $N_{\ell-1}$ samples of X given the same risk scenario Y. Here, $\mathrm{Var}[\Delta H_\ell(Y)]$ decreases with increasing ℓ (ie, with an increasing number of internal samples), even if the internal samples used in $\hat{E}_\ell$ and $\hat{E}_{\ell-1}$ are mutually independent. This is because $\hat{E}_\ell(Y)$ converges almost surely to the expectation $\mathbb{E}[X \mid Y]$, due to the Strong Law of Large Numbers. However, by carefully using the same samples of X in both $\hat{E}_\ell$ and $\hat{E}_{\ell-1}$, we can reduce the variance by a constant factor.

In particular, for a given risk scenario, Y, assume $N_\ell \geqslant N_{\ell-1}$ and let $N_\ell = sN_{\ell-1}$ for some integer $s > 0$. Such an integer exists since the adaptive algorithm always returns $N_0 2^{\hat{\ell}}$ for some integer $\hat{\ell}$. Then, let $\{X^{(n)}\}_{n=1}^{N_\ell}$ denote N_ℓ samples of X given Y and define $\hat{E}_\ell(Y)$ as in (4.1). In addition, define s coarse approximations as

$$\hat{E}^{(i)}_{\ell-1}(Y) = \frac{1}{N_{\ell-1}} \sum_{n=1}^{N_{\ell-1}} X^{(n+(i-1)N_{\ell-1})}(Y)$$

for $i = \{1, 2, \ldots, s\}$. The MLMC estimator with antithetic sampling is given by

$$\sum_{\ell=0}^{L} \frac{1}{M_\ell} \sum_{m=1}^{M_\ell} \tilde{\Delta} H_\ell(Y^{(\ell,m)}), \quad \text{where } \tilde{\Delta} H_\ell(y) = H(\hat{E}_\ell(y)) - \frac{1}{s} \sum_{i=1}^{s} H(\hat{E}^{(i)}_{\ell-1}(y)).$$

Note that, since $\mathbb{E}[\tilde{\Delta} H_\ell(Y)] = \mathbb{E}[\Delta H_\ell(Y)]$, the MLMC estimator with antithetic sampling has the same expectation. Moreover, since $\tilde{\Delta} H_\ell = 0$ whenever $\hat{E}_\ell$ and all $\hat{E}^{(i)}_{\ell-1}$ for $i = \{1, 2, \ldots, s\}$ have the same sign, we have that $\mathrm{Var}[\tilde{\Delta} H_\ell(Y)] \leqslant \mathrm{Var}[\Delta H_\ell(Y)]$. When $N_\ell \leqslant N_{\ell-1}$, which may occur due to inaccurate estimates of $\mathbb{E}[X \mid Y]$ and $\mathrm{Var}[X \mid Y]$, the same discussion as above applies, with the fine approximation having the antithetic estimators instead of the coarse one.

4.2 Starting level of the MLMC

An important point to consider when using MLMC is the choice of the starting level. To explain this, let $V_\ell := \mathrm{Var}[\tilde{\Delta} H_\ell(Y)]$ and $V^{\mathrm{f}}_\ell := \mathrm{Var}[H(\hat{E}_\ell(Y))]$ and let W_ℓ denote the expected work of sampling $\tilde{\Delta} H$; in the current setting we have $W_\ell \equiv \mathbb{E}[N_\ell]$. Then, consider the MLMC estimator

$$\frac{1}{M_0} \sum_{m=1}^{M_0} H(\hat{E}_{\ell_0}(Y^{(\ell_0,m)})) + \sum_{\ell=\ell_0+1}^{L} \frac{1}{M_\ell} \sum_{m=1}^{M_\ell} \tilde{\Delta} H_\ell(Y^{(\ell,m)}),$$

which starts at some level $\ell_0 \geqslant 0$ as the coarsest approximation. It can be shown (Giles 2015) that the expected work of MLMC is proportional to

$$\left((V^{\mathrm{f}}_{\ell_0} W_{\ell_0})^{1/2} + \sum_{\ell=\ell_0+1}^{L} (V_\ell W_\ell)^{1/2} \right)^2.$$

Hence, given some level of approximation L, an optimal ℓ_0 satisfies

$$(V^{\mathrm{f}}_{\ell_0} W_{\ell_0})^{1/2} + \sum_{\ell=\ell_0+1}^{\ell'_0} (V_\ell W_\ell)^{1/2} < (V^{\mathrm{f}}_{\ell'_0} W_{\ell'_0})^{1/2} \tag{4.5}$$

for all $\ell_0 < \ell'_0 \leqslant L$. Otherwise, starting at the level ℓ'_0 leads to less computational work overall. Since the quantities V_ℓ and V^{f}_ℓ for $\ell = 0, 1, \ldots, L$ must be approximated using a sample variance estimator, we may relax the previous condition by multiplying the right-hand side by some constant larger than 1 to increase the stability of the MLMC algorithm. We use the constant 1.5 in our numerical examples in Section 5.

Choosing an optimal starting level is especially relevant in nested simulation applications because the variance V^{f}_ℓ may be large for small ℓ but then decreases as more samples are used in the inner estimator, asymptotically converging to $\mathrm{Var}[H(\mathbb{E}[X \mid Y])]$ (see Section 5 and Figure 3 for an illustration of this).

5 NUMERICAL EXPERIMENTS

In this section, using numerical experiments on fictitious portfolios of put and call options, we will illustrate the benefits of using random subsampling (as discussed

in Section 2), the control variates that were discussed in Section 3 and adaptive sampling (as discussed in Section 4).

5.1 Test setup

5.1.1 Underlying assets

We assume we have Q assets, $S \equiv \{S_k\}_{k=1}^{Q}$, modeled by geometric Brownian motions satisfying

$$\mathrm{d}S_k(t) = \mu_k S_k(t)\,\mathrm{d}t + \sigma_k S_k(t)(\rho\,\mathrm{d}B_0(t) + (1-\rho^2)^{1/2}\,\mathrm{d}B_i(t))$$

in the physical measure. Here the Brownian process B_0 is the systematic noise, common to all assets, while $\{B_i\}_{i=1}^{Q}$ are mutually independent Brownian processes and represent the idiosyncratic noise of each asset. We select the following parameters:

number of assets,	$Q \equiv 16$;
initial asset price,	$S_k(0) \in [90, 110]$;
drift rate,	$\mu_k \in [0.05, 0.15]$;
volatility,	$\sigma_k \in [0.01, 0.4]$;
correlation coefficient,	$\rho \equiv 0.2$.

5.1.2 Portfolio construction

The loss in our example portfolio is the average of the losses from P derivatives (see (1.2)) (ie, $\Lambda \equiv P^{-1}\sum_{i=1}^{P}\Lambda_i$) and we consider the market risk. For a short risk horizon $\tau = 0.02$, we set the risk parameter to be the value of the underlying assets at τ (ie, $R_\tau \equiv S(\tau)$), and then set

$$\Lambda_i \equiv w_i(h_i(S_{k_i}(T_i)) - h_i(S_{k_i,\tau,R_\tau}(T_i)))$$

for some weight w_i and with h_i the discounted payoff function for the ith option. Here, S_{k,τ,R_τ} is the kth asset conditioned on $S(\tau) = R_\tau$. We assume that the risk-free interest rate $r = 0.05$ and the discount factor at time t is $\exp(-rt)$. Each option is characterized by its type (put or call), which determines the payoff function h_i, along with the following parameters:

asset,	$k_i \in \{1, 2, \ldots, Q\}$;
maturity,	$T_i \in [0, 5]$;
strike,	$K_i \in [80, 120]$;
weight,	$w_i \equiv \begin{cases} \tilde{w}_i, & \text{put option,} \\ \tilde{w}_i b_{k_i}, & \text{call option.} \end{cases}$

To get concrete values for the parameters above, we generate a random instance of the assets and the portfolio by taking the type to be put or call with equal probability (ensuring at least a single put option and single call option for each underlying asset), and $S_k(0)$, μ_k, σ_k, k_i, T_i and K_i are sampled independently and uniformly in their respective ranges. The parameters b_{k_i} are balancing constants that are determined by the constraint that the portfolio should be delta-neutral with respect to the risk parameter at the initial time $R_0 = \{R_{0,k}\}_{k=1}^{Q} = \{S_k(0)\}_{k=1}^{Q}$; ie,

$$\sum_{i=1}^{P} \frac{\partial V_{i,0}}{\partial R_{0,k}} = 0 \quad \forall k \in \{1, \dots, Q\}.$$

More specifically, for $i = 1, \dots, Q$, we set

$$b_k \equiv -\Bigg(\sum_{\substack{i=1,\\ \text{put option}}}^{P} \frac{\partial}{\partial R_{0,k}} \mathbb{E}[\tilde{w}_i V_{i,0}] \Bigg) \Bigg(\sum_{\substack{i=1,\\ \text{call option}}}^{P} \frac{\partial}{\partial R_{0,k}} \mathbb{E}[\tilde{w}_i V_{i,0}] \Bigg)^{-1}.$$

We will discuss the choice of $\{\tilde{w}_i\}_{i=1}^{P}$ in our fictitious portfolios below. In any case, the last step is to normalize the weights $\{w_i\}_{i=1}^{P}$, so that their average is 1.

5.1.3 Computation methods

We consider the three computational models for computing the value of the options:

(a) an exact, deterministic evaluation of the option value using the analytic solution of the Black–Scholes PDE;

(b) an exact simulation of the asset values by solving the SDE analytically; and

(c) an approximate simulation using the Milstein numerical scheme to estimate the asset values.

5.2 Results

All numerical experiments use MLMC with an initial number of samples $M_0 = 1024$ to estimate the work and variance of the MLMC levels. Moreover, for the inner Monte Carlo estimator, we set $N_0 = 32$ and when using the adaptive algorithm to select the number of inner samples we set $r = 1.5$ and $C = 3$ in (4.2). The code was written in C++ and the experiments were carried out in single-precision format on an NVIDIA Tesla K20m processor with 2496 cores.[2] Note that the embarrassingly

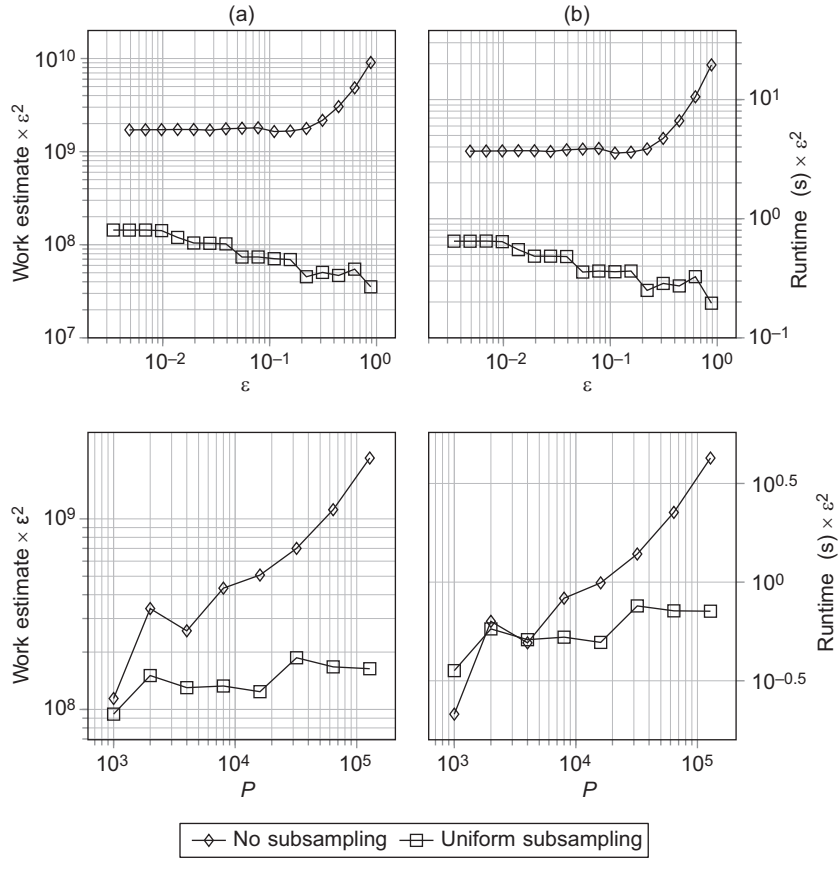
[2] The full C++ code can be found at https://github.com/haji-ali/nested-risk-mlmc.

parallel nature of Monte Carlo simulation makes it possible to fully exploit parallelization in addition to the computational savings provided by the subsampling approach.

To illustrate the benefit of uniform random subsampling we first consider large, delta-hedged portfolios composed of options with similar nominal values (ie, $\tilde{w}_i = 1$ for all i). The computation method to evaluate each option is chosen to be exact evaluation or exact simulation with probabilities 30% and 70%, respectively. We compare two methods. In the first method we use random subsampling with uniform probabilities (ie, setting $\tilde{g}_i = 1$ for all i). In the second method we do not use any subsampling and instead evaluate the full portfolio for every combination of risk scenarios and underlying asset values, making sure that options that can be computed exactly are evaluated only once for every risk scenario. Both methods use MLMC with adaptive sampling, as discussed in Section 4, with an appropriate redefinition of X and Y as well as using all the control variates discussed in Section 3. When estimating the work of these methods, we simply count the number of times the value of an option or a payoff function is evaluated; the work estimates are shown in Figure 1(a). For the tolerances considered, using random subsampling consistently leads to fewer evaluations, and for a fixed tolerance the total number of payoff evaluations does not increase as the number of options increases. Figure 1(b) shows the actual run time for the numerical tests. Uniform, random subsampling has an overhead that makes its advantage slightly less pronounced for small tolerances or small portfolios. To explain these results, recall that evaluating the full portfolio for every combination of risk scenarios and underlying asset values (ie, not using subsampling) imposes a minimum budget that increases the computational complexity for large tolerances. Nevertheless, for sufficiently small tolerances or portfolios and sufficiently large budgets, evaluating the full portfolio for every risk scenario does not add a significant computational overhead. On the other hand, random subsampling has an overhead not accounted for in the work estimate (namely, the cost of sampling the random option index, which entails sampling a uniform random variable and a table lookup operation). While this additional cost is small in typical cases, especially since we use a binary search to perform the table lookup, it is not wholly insignificant compared with the cost of sampling the options in our simple numerical example.

Random subsampling is most useful when the financial derivatives in the portfolio are heterogeneous, even in moderate-sized portfolios. To illustrate this we consider a smaller portfolio of 10^3 options with different nominal values. To model this, we sample the logarithm of the weight parameters, $\log(\tilde{w}_i)$, from a normal distribution with mean 0 and standard deviation 3. Moreover, when using random subsampling we use the estimates $\tilde{g}_i = \tilde{w}_i$. As before, the computation method of each portfolio

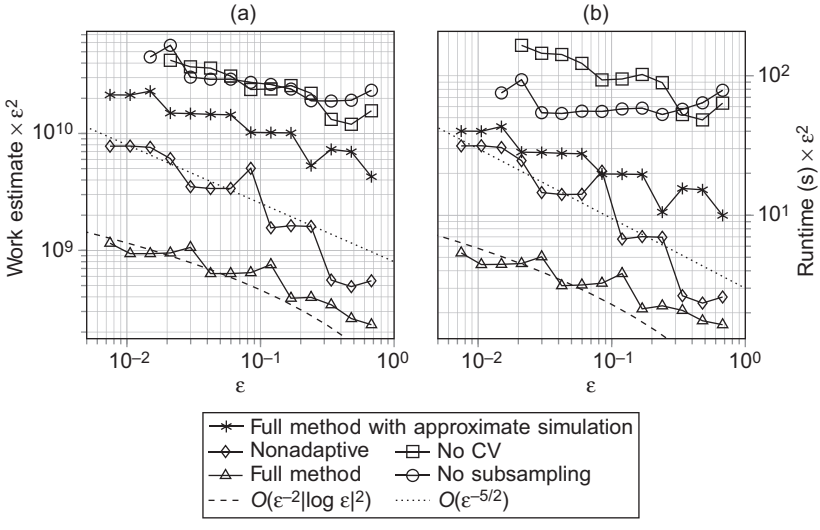
FIGURE 1 (a) The work estimates (measured as the number of evaluations of option values and payoff functions) and (b) runtime (measured in seconds) of MLMC with adaptive sampling when applied to large portfolios of options with similar nominal values (ie, $\tilde{w}_i = 1$ for all i).



Of the options studied, 30% are computed using exact evaluation while 70% are computed using exact simulation. Here ε is the tolerance normalized by the exact value, which was estimated using Monte Carlo to be approximately 3–4% for the portfolios considered. Note that the work estimates and running time are multiplied by ε^2 to normalize the work effort for different portfolios and to emphasize the difference in the computational effort when using – or not using – random, uniform subsampling. In the top two panels we fix the portfolio size to $P = 10^5$ and vary ε, while in the bottom panels we fix $\varepsilon \approx 3 \times 10^{-3}$ and vary P. We see that using random subsampling, even when applied to options with similar nominal value, reduces the computational complexity, particularly for large tolerances. Moreover, the computational complexity is independent of the number of options in the portfolio.

is chosen to be exact evaluation or exact simulation, with probabilities 30% and 70%, respectively. We now test several methods and show their work estimates and run times in Figure 2.

FIGURE 2 (a) The work estimate and (b) runtime of MLMC with adaptive sampling when applied to a portfolio of 10^3 heterogeneous options.



Here ε is the tolerance, normalized by the exact value, which was estimated using Monte Carlo simulation to be approximately 1% for our particular portfolio. Note that the work estimates and running time are multiplied by ε^2 to emphasize the differences between the methods, since $\mathcal{O}(\varepsilon^{-2})$ is the computational complexity in the best case, when the inner expectation can be computed exactly at $\mathcal{O}(1)$ cost. The full method, which uses MLMC with adaptive inner sampling, all control variates (as discussed in Section 3) and random subsampling with nonuniform probabilities, clearly outperforms the other methods.

The first method (labeled "full method") uses MLMC with adaptive sampling (as discussed in Section 4), all the control variates (as discussed in Section 3) and random subsampling (as discussed in Section 2). The second method (labeled "no subsampling") does not use random subsampling and instead evaluates the whole portfolio for every combination of risk scenarios and asset values, again making sure that options that can be computed exactly are evaluated once for every risk scenario. In this case, the work reduction measured by work estimates and total runtime is more than tenfold. The third method we consider (labeled "no CV") is the same as the "full method" except that we do not use the Delta and antithetic control variates that were discussed in Section 3. In this example, by using these control variates, the work estimate and runtime are again reduced around 40-fold. Recall that this reduction is related to the risk horizon, $\tau = 0.02$, and we should expect that longer risk horizons (compared with the maturities of options) would reduce the savings of the antithetic and Delta control variates. The fourth method we consider (labeled "nonadaptive") is again the same as the "full method" except that it instead uses a deterministic,

nonadaptive number of inner samples (ie, $N_\ell = N_0 4^\ell$ for all risk scenarios). Using adaptive sampling is between two and seven times more efficient than nonadaptive sampling. Moreover, recall that to achieve RMSE ε we expect MLMC with adaptive sampling to have a computational complexity of $\mathcal{O}(\varepsilon^{-2}|\log\varepsilon|^2)$, while MLMC with nonadaptive sampling would have a complexity of approximately $\mathcal{O}(\varepsilon^{-5/2})$. The observed complexities in Figure 2 are consistent with the expected complexities and with the variance and work estimates in Figure 3.

To show that using the framework outlined above accommodates approximate simulation, we also include in these plots the runtime of the "full method" when applied to a similar portfolio with the same number of options and the same weights but with the computational method being exact evaluation, exact simulation or approximate simulation with probabilities 30%, 50% and 20%, respectively. Recalling the discussion and the notation in Remark 3.1, we note that setting $r = 1.5$ in the adaptive algorithm to select the number of inner samples would not work in this setting. This is because we use the Milstein scheme to approximate samples of the underlying assets for 20% of the options, which yields $\beta = 2\gamma$, and we use unbiased MLMC with $\zeta = (\beta + \gamma)/2$ to approximate the expectation of the loss, as discussed in Section 3.3. Hence, the q-moments of the unbiased estimator are finite only for $q < 3$, while $r = 1.5$ requires finite q-moments for $q \geqslant 15$ to satisfy the condition (4.4). Instead, we set $r = 1.1$ in this case, which requires a finite q-moment for $q \approx 2.72$.

The starting levels, ℓ_0, of MLMC for each of the methods in this section were selected based on the criteria in (4.5). As discussed above, the correct choice of starting level is crucial in nested simulation because the variance,

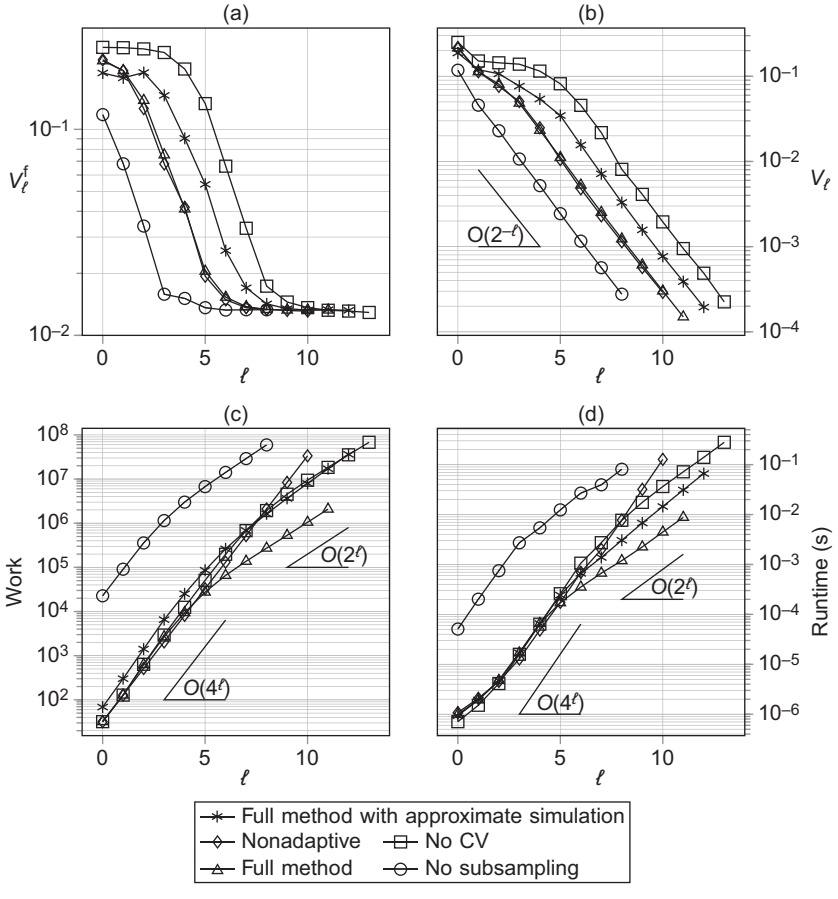
$$V_\ell^{\mathrm{f}} = \mathrm{Var}[H(\hat{E}_\ell(X \mid Y))],$$

may exhibit a pre-asymptotic behavior with respect to ℓ. This is illustrated in parts (a) and (b) of Figure 3.

6 CONCLUSIONS

This paper has shown the application of MLMC with adaptive sampling to estimating the probability of a large loss in a large financial portfolio of heterogeneous derivatives. The key elements in reducing computational complexity are using MLMC with adaptive sampling, applying several control variates that exploit the short risk horizon and using subsampling strategies to obtain a computational complexity that does not depend on the number of derivatives in the portfolio. By using the methods above to efficiently compute probabilities of loss in a portfolio, other risk measures such as VaR or CVaR can also be computed efficiently, as discussed in detail in Giles

FIGURE 3 The variance estimates of the MLMC, work estimate and runtime of levels.



Parts (a) and (b) show the variance estimates $V_\ell := \mathrm{Var}[\tilde{\Delta} H(Y)]$ and $V_\ell^{\mathrm{f}} := \mathrm{Var}[H(\hat{E}_\ell(Y))]$, respectively. Note that V_ℓ^{f} has a pre-asymptotic behavior where it asymptotically approaches $\mathrm{Var}[H(E[X \mid Y])]$ from above. Because of this, the starting level should be chosen carefully, as discussed in Section 4. Note also that V_ℓ decreases with $\mathcal{O}(2^{-\ell})$ for all methods. Part (c) shows the work estimate, and part (d) the runtime of the MLMC levels. Note that the work increases with $\mathcal{O}(2^\ell)$ for methods that use adaptive inner sampling for sufficiently large ℓ, unlike the nonadaptive method, where the work increases with 4^ℓ for all ℓ. In addition, when not using the control variates and because of the increase in the variance per level, the region of pre-asymptotic behavior where the work increases with 4^ℓ is extended.

and Haji-Ali (2019). VaR can be computed by finding the root $\mathcal{K}_\eta$ of the equation $\mathbb{P}[\mathbb{E}[\Lambda \mid R_\tau] > \mathcal{K}_\eta] = \eta$ for a given risk level η. Given an efficient method to solve the forward problem (ie, computing η given an estimate of $\mathcal{K}_\eta$), the root can be approximated efficiently using a stochastic root-finding algorithm (see Giles and Haji-Ali 2019). Since CVaR can be written as a minimization problem whose

solution is VaR (Giles and Haji-Ali 2019; Rockafellar and Uryasev 2002), setting $X = \mathbb{E}[\Lambda \mid R_\tau]$ and given an estimate of VaR, $\tilde{\mathcal{K}}_\eta$, we can write

$$\begin{aligned}\mathbb{E}[X \mid X > \mathcal{K}_\eta] &= \mathcal{K}_\eta + \eta^{-1}\mathbb{E}[\max(0, X - \mathcal{K}_\eta)] \\ &= \min_x\{x + \eta^{-1}\mathbb{E}[\max(0, X - x)]\} \\ &= \tilde{\mathcal{K}}_\eta + \eta^{-1}\mathbb{E}[\max(0, X - \tilde{\mathcal{K}}_\eta)] + \mathcal{O}(\tilde{\mathcal{K}}_\eta - \mathcal{K}_\eta)^2.\end{aligned}$$

Hence, to approximate the CVaR, we first approximate $\tilde{\mathcal{K}}_\eta$ up to an RMSE of $\varepsilon^{1/2}$ with work $\mathcal{O}(\varepsilon^{-2})$. Then, $\mathbb{E}[\max(0, \mathbb{E}[\Lambda \mid R_\tau] - \tilde{\mathcal{K}}_\eta)]$, involving a nested expectation, can be estimated with total work $\mathcal{O}(\varepsilon^{-2})$ to achieve an RMSE ε using MLMC with antithetic sampling for nested expectations (Bujok *et al* 2013; Giles 2018; Giles and Haji-Ali 2019) combined with random subsampling of the financial derivatives in the portfolio and the control variates that were discussed in Sections 2 and 3, respectively.

DECLARATION OF INTEREST

The authors report no conflicts of interest. The authors alone are responsible for the content and writing of the paper.

ACKNOWLEDGEMENTS

The numerical experiments in this work were performed on an NVIDIA Tesla K20m processor that was provided by the Edinburgh Centre for Robotics' Robotarium Cluster located at Heriot-Watt University, funded by the Engineering and Physical Sciences Research Council Centre for Doctoral Training in Robotics and Autonomous Systems through grant EP/L016834/1.

REFERENCES

Broadie, M., Glasserman, P., and Jain, G. (1997). Enhanced Monte Carlo estimates for American option prices. *Journal of Derivatives* **5**(1), 25–44 (https://doi.org/10.3905/jod.1997.407983).

Broadie, M., Du, Y., and Moallemi, C. C. (2011). Efficient risk estimation via nested sequential simulation. *Management Science* **57**(6), 1172–1194 (https://doi.org/10.1287/mnsc.1110.1330).

Bujok, K., Hambly, B. M., and Reisinger, C. (2013). Multilevel simulation of functionals of Bernoulli random variables with application to basket credit derivatives. *Methodology and Computing in Applied Probability* **17**, 579–604 (https://doi.org/10.1007/s11009-013-9380-5).

Giles, M. B. (2008). Multilevel Monte Carlo path simulation. *Operations Research* **56**(3), 607–617 (https://doi.org/10.1287/opre.1070.0496).

Giles, M. B. (2015). Multilevel Monte Carlo methods. *Acta Numerica* **24**, 259–328 (https://doi.org/10.1017/S096249291500001X).

Giles, M. B. (2018). MLMC for nested expectations. In *Contemporary Computational Mathematics: A Celebration of the 80th Birthday of Ian Sloan*, Dick, J., Kuo, F. Y., and Woźniakowski, H. (eds), pp. 425–442. Springer (https://doi.org/10.1007/978-3-319-72456-0_20).

Giles, M. B., and Haji-Ali, A. L. (2019). Multilevel nested simulation for efficient risk estimation. *SIAM/ASA Journal on Uncertainty Quantification* **7**(2), 497–525 (https://doi.org/10.1137/18M1173186).

Giles, M. B., Higham, D. J., and Mao, X. (2009). Analysing multi-level Monte Carlo for options with non-globally Lipschitz payoff. *Finance and Stochastics* **13**, 403–413 (https://doi.org/10.1007/s00780-009-0092-1).

Giles, M. B., Debrabant, K., and Rössler, A. (2019). Analysis of multilevel Monte Carlo path simulation using the Milstein discretisation. *Discrete and Continuous Dynamical Systems* B **24**, 3881–3903 (https://doi.org/10.3934/dcdsb.2018335).

Giorgi, D., Lemaire, V., and Pagès, G. (2017). Limit theorems for weighted and regular multilevel estimators. *Monte Carlo Methods and Applications* **23**(1), 43–70 (https://doi.org/10.1515/mcma-2017-0102).

Glasserman, P. (2003). *Monte Carlo Methods in Financial Engineering*. Stochastic Modelling and Applied Probability, Volume 53. Springer (https://doi.org/10.1007/978-0-387-21617-1).

Gordy, M. B., and Juneja, S. (2010). Nested simulation in portfolio risk measurement. *Management Science* **56**(10), 1833–1848 (https://doi.org/10.1287/mnsc.1100.1213).

Gou, W. (2016). Estimating value-at-risk using multilevel Monte Carlo maximum entropy method. Master's Thesis, University of Oxford.

Haji-Ali, A. L., Nobile, F., von Schwerin, E., and Tempone, R. (2015). Optimization of mesh hierarchies in multilevel Monte Carlo samplers. *Stochastics and Partial Differential Equations: Analysis and Computations* **4**(1), 76–112 (https://doi.org/10.1007/s40072-015-0049-7).

Rhee, C. H., and Glynn, P. W. (2015). Unbiased estimation with square root convergence for SDE models. *Operations Research* **63**(5), 1026–1043 (https://doi.org/10.1287/opre.2015.1404).

Rockafellar, R. T., and Uryasev, S. (2002). Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance* **26**(7), 1443–1471 (https://doi.org/10.1016/S0378-4266(02)00271-6).