THREE IMPROVEMENTS TO MULTI-LEVEL MONTE CARLO SIMULATION OF SDE SYSTEMS

L.F. RICKETSON∗†

Abstract. We introduce three related but distinct improvements to multilevel Monte Carlo (MLMC) methods for the solution of systems of stochastic differential equations (SDEs). Firstly, we show that when the payoff function is twice continuously differentiable, the computational cost of the scheme can be dramatically reduced using a technique we call ‘Ito linearization’. Secondly, by again using Ito’s lemma, we introduce an alternative to the antithetic method of Giles et. al [M.B. Giles, L. Szpruch. arXiv preprint arXiv:1202.6283, 2012] that uses an approximate version of the Milstein discretization requiring no Lévy area simulation to obtain the theoretically optimal cost-to-error scaling. Thirdly, we generalize the antithetic method of Giles to arbitrary refinement factors. We present numerical results and compare the relative strengths of various MLMC-type methods, including each of those presented here.

Key words. stochastic differential equations, multi-level, Monte Carlo, Ito’s lemma

AMS subject classifications. 65C05, 65C30

1. Introduction. Stochastic differential equations (SDEs) have numerous applications: neuroscience [112], chemical kinetics [8], civil engineering [910], biological fluid dynamics [15], physics [1619], and finance [20], to name a few. A prototypical class of problems may be characterized as follows: let \( S(t) \in \mathbb{R}^d \) satisfy the system of SDEs

\[
dS_i = a_i(S, t) \, dt + \sum_{j=1}^{D} b_{ij}(S, t) \, dW_j, \quad S(0) = S_0 \tag{1.1}
\]

for \( t \in [0, T] \) and some given \( S_0 \), where \( S_i \) is the \( i \)th component of \( S \), \( W(t) \in \mathbb{R}^D \) is a \( D \) dimensional Brownian motion, \( a_i : \mathbb{R}^d \to \mathbb{R} \) for each \( i \in \{1, 2, \ldots, d\} \), and similarly for \( b_{ij} \). Then, for some given \( P : \mathbb{R}^d \to \mathbb{R} \), evaluate \( \mathbb{E}[P(S(T))] \). That is, we wish to find the mean value of some functional of the solution of an SDE.

Since exact solutions are available for only the simplest of SDEs, finite difference methods are frequently used to approximate their solutions. The expectation is then evaluated via a Monte Carlo method. The purpose of the present work is to present three improvements to the class of multilevel Monte Carlo (MLMC) methods - introduced in [5] - which are the current state of the art.

The MLMC methods themselves improve upon the most straightforward numerical method for the archetypal SDE problem above. That method is to approximate the SDE’s solution by the well-known Euler-Maruyama discretization with time step \( h \), given by

\[
S_{i,n+1} = S_{i,n} + a_i(S_n, t_n)h + \sum_{j=1}^{D} b_{ij}(S_n, t_n)\Delta W_{j,n}, \tag{1.2}
\]

where \( S_n \) approximates \( S(t_n) \), with \( t_n = nh \), and the \( \Delta W_{j,n} \) are independent normal random variables with mean zero and variance \( h \). We may then generate \( N \) independent samples of \( S_{T/h} \) by generating different \( \Delta W_{j,n} \) for each sample, and estimate the desired expectation by

\[
\mathbb{E}[P(S(T))] \approx \frac{1}{N} \sum_{r=1}^{N} P \left( \frac{S(T)}{h} \right), \tag{1.3}
\]

∗This work was performed under the auspices of the U.S. DOE by the University of California, Los Angeles, under grant DE-FG02-05ER25710.
†Mathematics Department, University of California at Los Angeles, lfr@math.ucla.edu
where $r$ indexes the $N$ samples.

One desires to approximate the true expectation to within an RMS error $\varepsilon$, which will scale as $O(N^{-1/2})$ and $O(h)$. The computational cost of the scheme is proportional to the total number of time steps taken, which scales as $O(N/h)$. Thus, we see that the computational cost of achieving an RMS error $\varepsilon$ - which we henceforth denote by $K$ - is $O(\varepsilon^{-3})$.

In many contexts, such a scaling is prohibitive, so a number of methods which improve upon it have been developed. To understand them, we must define the notions of strong and weak errors for SDE approximations. Let $S_h$ be an approximate solution of (1.1) obtained by some discretization with time-step $h$. We say that discretization has weak error of order $p$ if

$$\|E[g(S_h)] - E[g(S)]\| = O(h^p)$$

for some broad class of functions $g : \mathbb{R}^d \to \mathbb{R}$ (in particular, that class should include $P$). We say that discretization has strong error of order $q$ if

$$\mathbb{E}[|S_h - S|] = O(h^q).$$

We note that the Euler discretization has $p = 1$ and $q = 1/2$ [13].

It is straightforward to see that if we modify the naive scheme presented above to use a discretization of weak order $p$, we have

$$K = O\left(e^{-(2+1/p)}\right),$$

independent of $q$. In contrast, the multilevel Monte Carlo (MLMC) methods introduced in [5] and expanded in [4, 6] achieve

$$K = \begin{cases} O\left(e^{-2\log\varepsilon}\right) & : q = 1/2 \\ O\left(e^{-2}\right) & : q > 1/2 \end{cases}$$

so long as $p > 0$. The proof of this fact may be found in [5], and we sketch the argument in section 2.

We thus see that the multilevel method scales better than the naive method outlined above for any discretization with finite weak order $p$. Moreover, the larger the weak order of a discretization, the more regularity we require of $P$ to achieve that order [13], further limiting the use of high-order weak schemes. Multilevel schemes are thus a great improvement over simple schemes of the type outlined above.

The MLMC schemes achieve their improved cost scaling by approximating the SDE’s solution with many different time-steps (called ‘levels’) and taking advantage of the discretization’s strong convergence to get low variance estimates of the difference in the payoff at adjacent levels. The remaining high variance quantity - the payoff’s expectation at the lowest level - is relatively cheap to compute because of the large time-step. However, the algorithm could be further improved by also applying a variance reduction at this lowest level. The first contribution of the present work is to show that, when the payoff function is twice continuously differentiable, we can reduce the variance at the lowest level to zero by finding the payoff using Ito’s lemma instead of direct evaluation.

Our second contribution is to again make use of Ito’s lemma to derive a variant of the MLMC method that achieves the cost scaling $O(\varepsilon^{-2})$ in spite of having $q = 1/2$. This is a desirable result because discretizations with $q > 1/2$ require the simulation of Lévy areas when $D > 1$, and Lévy areas are notoriously difficult to sample. Indeed, no suitable algorithm has been implemented for $D > 2$. A method achieving $O(\varepsilon^{-2})$ scaling without Lévy area simulation was also derived in
However, the method we propose, while similar in some respects, is simpler to derive and slightly faster for twice differentiable payoffs.

Thirdly, we make use of our analysis to generalize the antithetic method in [6] to arbitrary refinement factor - that is, the ratio between the time-steps at adjacent levels. The method was originally derived for the case of refinement factor $M = 2$, but we show that $M \approx 4$ to $5$ is optimal. Importantly, the generalization to arbitrary $M$ still requires the sampling of only one antithetic path, so the generalization introduces no extra computational complexity. The key lemma in this development - Lemma 5.1 in the present work - was originally proved in [7] toward a different end. Given this lemma, the result is straightforward, but does not appear elsewhere in the literature to the author's knowledge.

The remainder of the paper is structured as follows. Section 2 reviews the details of MLMC methods and the difficulty in implementing SDE solvers with $q > 1/2$, focusing in particular on the Milstein discretization. In section 3, we show how Ito’s lemma can be used to eliminate the lowest level variance in MLMC methods. In section 4, we use the results of the previous section to derive an ‘approximate Milstein’ version of the MLMC method that achieves the $O(\varepsilon^{-2})$ cost scaling. In section 5, we leverage results from the previous section to generalize the antithetic method of [6]. In section 6, we summarize results and present pseudocode for the algorithms proposed in previous sections. In section 7, we present and discuss numerical results. We conclude in section 8.

2. Background. The first portion of this section reviews the derivation and basic properties of MLMC methods, while the second reviews the Milstein discretization, the difficulties inherent in its implementation, and some previous efforts to negotiate those difficulties. For more details on elementary MLMC, see [5]. For more information on Milstein, see [3, 14, 21].

2.1. MLMC Review. The MLMC schemes are constructed in the following way: for some integer $M > 1$, let $h_l = TM^{-l}$ for $l = 0, 1, 2, ..., L$. Setting $P_l = P(S_{h_l}(T))$, the following identity holds:

$$E[P_L] = E[P_0] + \sum_{l=1}^{L} E[P_l - P_{l-1}].$$  \hspace{1cm} (2.1)

The weak convergence of the discretization guarantees that $E[P_L]$ differs from the true expectation by $O(h_L^p)$, and (2.1) shows that it can be estimated by estimating the $L + 1$ expectations on the right side. The first term is relatively cheap to compute, since the time-step $h_0 = T$ is much larger than $h_L$. Meanwhile, the quantities $P_l - P_{l-1}$ have variances controlled by the strong convergence of the discretization, so that their expectations can be estimated accurately with a relatively small number of samples.

We make this more concrete by defining

$$V_l = \text{Var}[P_l - P_{l-1}],$$  \hspace{1cm} (2.2)

for $l > 0$, where $\text{Var}[\cdot]$ denotes the variance of a random variable, and assuming $P$ has a global Lipschitz bound. Then, if $P_l$ and $P_{l-1}$ are sampled using the same Brownian paths, we have

$$V_l = E[(P_l - P_{l-1})^2] - E[P_l - P_{l-1}]^2 \\ \lesssim E|S_{h_l} - S_{h_{l-1}}|^2 + O(h_l^{2p}) \\ = O(h_l^{2p}) + O(h_l^{2p}).$$  \hspace{1cm} (2.3)
It is a general feature of SDE finite difference methods that \( p \geq q \) \[13\], so we will write
\[
V_l = O \left( h_l^{2q} \right)
\] (2.4)
henceforth.

If we estimate \( \mathbb{E} [P_l - P_{l-1}] \) with \( N_l \) samples - that is
\[
\mathbb{E} [P_l - P_{l-1}] \approx \hat{Y}_l \equiv \frac{1}{N_l} \sum_{r=1}^{N_l} \left( P_l^{(r)} - P_{l-1}^{(r)} \right),
\] (2.5)
where \( r \) again indexes the \( N_l \) samples - then the variance in this estimate is \( V_l / N_l \). Similarly, define \( V_0 = \text{Var}[P_0] \) and let
\[
\hat{Y}_0 = \frac{1}{N_0} \sum_{r=1}^{N_0} P_0^{(r)}.
\] (2.6)

Then, let \( \hat{P}_L \) be our estimate of \( \mathbb{E}[P_L] \) defined by
\[
\hat{P}_L = \sum_{l=0}^{L} \hat{Y}_l.
\] (2.7)

This estimate has variance
\[
\text{Var} \left[ \hat{P}_L \right] = \sum_{l=0}^{L} \frac{V_l}{N_l}.
\] (2.8)

The desired RMS error bound of \( \varepsilon \) may thus be written as
\[
(c_1 h_L)^2 + \sum_{l=0}^{L} \frac{V_l}{N_l} \leq \varepsilon^2,
\] (2.9)
where \( c_1 \) is the constant of proportionality in the weak error estimate of the SDE scheme. That is,
\[
\| \mathbb{E}[P(S(T))] - \mathbb{E}[P_L] \| \approx c_1 h_L
\] (2.10)
for sufficiently small \( h_L \). Note that we assume the scheme is first order in the weak sense \( (p = 1) \), a quality shared by all the schemes considered in this paper. We call the first term in (2.9) the \textit{bias error}; it is deterministic and arises from the finite time-step approximation of the SDE’s solution. We call the second term - the sum - the \textit{sampling error}; it arises from the estimation of expectations using a finite number of samples.

In the analysis of Giles, (2.9) is satisfied by setting each of the two mean squared errors to \( \varepsilon^2 / 2 \). The bias error constraint then immediately gives a formula for \( L \), the total number of levels to be used:
\[
L = \left\lceil \frac{\log \left( \frac{\sqrt{2} c_1 T / \varepsilon}{\log M} \right)}{4} \right\rceil.
\] (2.11)
The sampling error constraint gives rise to a constrained optimization problem: one wishes to minimize the computational cost—modeled by the total number of time steps taken—

\[ K \propto \sum_{l=0}^{L} N_l (h_l^{-1} + h_{l-1}^{-1}) = \left(1 + \frac{1}{M}\right) \sum_{l=0}^{L} \frac{N_l}{h_l}, \]  

subject to the constraint \( \sum_{l=0}^{L} (V_l/N_l) \leq \varepsilon^2/2 \). A Lagrange multiplier argument shows that the optimal choice is

\[ N_l = \frac{2}{\varepsilon^2} \sqrt{V_l h_l} \left( \sum_{l=0}^{L} \sqrt{V_l h_l} \right), \]  

which in turn gives the cost

\[ K \propto \frac{2}{\varepsilon^2} \left(1 + \frac{1}{M}\right) \left( \sum_{l=0}^{L} \sqrt{V_l h_l} \right)^2. \]  

When \( q = 1/2 \), we have \( V_l = O(h_l) \), so that each term in the sum is \( O(1) \), making the sum \( O(L) \). Since \( L \) scales like \( \log \varepsilon \), we see that \( K = O(\varepsilon^{-2}(\log \varepsilon)^2) \), as stated in the introduction.

In practice, the constant \( c_1 \) is not known, so \( L \) cannot be specified at the start of the simulation. One typically performs the necessary steps for \( L = 1 \), estimates the bias error by looking at \( \hat{Y}_L \), and increments \( L \) while the bias error is estimated to be more than \( \varepsilon/\sqrt{2} \). More details can be found in [5] and in section 6 of this paper.

### 2.2. Milstein and Lévy Areas

The simplest finite difference scheme for SDEs achieving \( q > 1/2 \) and thus yielding the optimal MLMC scaling is the Milstein scheme, written as

\[ S_{i,n+1} = S_{i,n} + a_{i,n} \Delta t + \sum_{j=1}^{D} b_{ij,n} \Delta W_{j,n} + \sum_{j,k=1}^{D} h_{ijk,n}(\Delta W_{j,n} \Delta W_{k,n} - \Omega_{jk} \Delta t - A_{jk,n}), \]  

where we’ve abbreviated \( a_i(S_n, t_n) = a_{i,n} \) and similarly for \( b_{ij,n} \) and \( h_{ijk,n} \). \( \Omega_{jk} \) is the correlation matrix associated with \( W \), and \( h \) and \( A \) are defined by

\[ h_{ijk} = \frac{1}{2} \sum_{l=1}^{d} b_{lk} \frac{\partial b_{ij}}{\partial x_l}, \]  

\[ A_{jk,n} = \int_{t_n}^{t_{n+1}} \int_{t_n}^{s} [dW_j(u) dW_k(s) - dW_k(u) dW_j(s)]. \]

The \( A_{jk,n} \) are known as Lévy areas. When \( D = 1 \), they vanish, since \( A_{jj,n} = 0 \), and Milstein is straightforward to implement. When \( D = 2 \), there is effectively only one non-zero Lévy area, since \( A_{jk,n} = -A_{kj,n} \). Recently, an efficient method has been developed for sampling a single Lévy area [2], making Milstein implementation feasible when \( D = 2 \). Sampling multiple Lévy areas is a more challenging problem because they are not independent. A method for jointly sampling multiple Lévy areas was also proposed in [2, 3] that builds upon the methods therein and involves sampling a random orthogonal matrix, techniques for which are available in [17].

When \( q = 1/2 \), we have \( V_l = O(h_l) \), so that each term in the sum is \( O(1) \), making the sum \( O(L) \). Since \( L \) scales like \( \log \varepsilon \), we see that \( K = O(\varepsilon^{-2}(\log \varepsilon)^2) \), as stated in the introduction.
However, this method has not been implemented or tested with MLMC. As a result, implementing the Milstein discretization and thus achieving the $O(\varepsilon^{-2})$ scaling for MLMC methods is quite challenging when $D > 2$, except in special cases.

Fortunately, in [6] it was observed that while $q > 1/2$ is sufficient to achieve the optimal scaling, it is not necessary. The necessary condition is

$$\text{Var}[P_l - P_{l-1}] = O(h_l^\beta)$$

for some $\beta > 1$. We can see from (2.23) that if $P$ has a global Lipschitz bound, this necessary condition is achieved if

$$E[|S_{hi} - S_{hi-1}|^2] = O(h_l^\beta).$$

This resembles a strong scaling requirement (1.5), but there is a key difference. Here, we require two approximate solutions to be within $O(h_l^\beta)$ of each other in the mean square sense. It is not necessary that either one of these approximate solutions be within $O(h_l^\beta)$ of the true solution, as would be the case if we were relying on strong convergence.

In [6], the Milstein scheme (2.15) with the Lévy areas set to zero, along with an antithetic path sampling method, is used in order to achieve (2.19) with $\beta > 1$, and thus achieve the $O(\varepsilon^{-2})$ cost scaling for SDE systems with arbitrary $D$. For the moment, we refer the reader to that paper for its detailed derivation and implementation. We will discuss some key aspects of the antithetic method as they become relevant in the course of our discussion here.

In section 4 of this paper, we derive an alternative method to that in [6]. We also achieve the $O(\varepsilon^{-2})$ cost scaling for arbitrary $D$ without simulating Lévy areas. Our method requires more regularity of the payoff function, but is slightly cheaper and simpler to derive. In section 5, we generalize the results of [6] to $M > 2$. Since much of the analysis from [6] carries over directly, we simply cite several results without reprinting proofs.

3. Variance Reduction via Ito’s Lemma. We begin with a simple observation. Suppose that $P(S) = S_m$, for some $1 \leq m \leq d$. That is, $P$ simply picks out one of the components of $S$. Such a payoff function is useful in chemical kinetics, for example, in which each component of the SDE represents the concentration of a particular species and we may desire to compute the mean concentration of some key compound.

Then, we may write a simple analytic expression for $P_0$ - the payoff when the time-step is $T$ - when the Euler discretization is used:

$$P_0 = S_{m,0} + a_m(S_0)T + \sum_{j=1}^{D} b_{mj}(S_0)W_j(T),$$

where $S_0$ is the initial data and $S_{m,0}$ is its $m^{th}$ component. The expectation of this expression is simple to evaluate:

$$E[P_0] = S_{m,0} + a_m(S_0)T.$$

The same result applies to the Milstein scheme, since the additional term has zero expectation. Thus, when $P$ has this simple form (or, indeed, is any linear function of $S$), the base payoff can be evaluated exactly in terms of the initial condition. There is no need to sample any random variables at all. In effect, $V_0 = 0$ and $N_0 = 1$.

This can represent a great computational saving for MLMC because, as already noted, the lowest level is the only level at which no variance reduction is gained. That is, with the standard
The fraction of the computational work exerted at each level in a sample MLMC computation. The Heston model - see (7.1) and proceeding text for specification - is solved with a sinusoidal payoff function, and $M = 2$. 

To illustrate this point, we show in fig. 1 the fraction of the computational work at each level in a sample MLMC computation, using both the Euler and antithetic methods. We see that, for each method, the base level (zero) is the most expensive. The base level represents an even larger fraction of the work in the antithetic method. This is a result of the improved variance scaling, which reduces the cost of the higher levels.

There is thus a motivation to investigate whether the technique of eliminating the cost of computing the base level payoff can be generalized to less trivial payoff functions. Toward that end, assume $P$ is twice continuously differentiable. Then, Ito’s lemma gives an SDE for $P$:

$$dP = \left( \sum_{i=1}^{d} a_i P_{x_i} + \frac{1}{2} \sum_{j=1}^{D} \sum_{i,k=1}^{d} b_{ij} b_{kj} P_{x_i x_k} \right) dt + \sum_{j=1}^{D} \sum_{i=1}^{d} b_{ij} P_{x_i} dW_j,$$

(3.3)

where subscripts on $P$ denote partial derivatives, and all functions are evaluated at $S(t)$.

With this in mind, construct a vector $S \in \mathbb{R}^{d+1}$ as follows: for $1 \leq k \leq d$, set $S_k = S_k$, and set $S_{d+1} = P(S)$. Then, $S$ solves

$$dS_i = \alpha_i dt + \sum_{j=1}^{D} \beta_{ij} dW_j$$

(3.4)

where $\alpha_i(S) = a_i(S)$ and $\beta_{ij}(S) = b_{ij}(S)$ for $i \leq d$, and

$$\alpha_{d+1}(S) = \sum_{i=1}^{d} a_i P_{x_i} + \frac{1}{2} \sum_{j=1}^{D} \sum_{i,k=1}^{d} b_{ij} b_{kj} P_{x_i x_k}, \quad \beta_{(d+1)j}(S) = \sum_{i=1}^{d} b_{ij} P_{x_i}.$$

(3.5)

This is a system of SDEs in the usual sense. Consider further the “payoff” function $\tilde{P}(S) = S_{d+1}$, which is equal to $P(S)$. We now have two distinct formulations of the same problem. The first is
to find \( \mathbb{E}[P(S(T))] \) when \( S \) solves (1.1). The second is to find \( \mathbb{E}[\hat{P}(S(T))]=\mathbb{E}[S_{d+1}(T)] \) when \( S \) solves (3.4).

The second, new formulation has the considerable advantage that its payoff function is linear, and in particular is of the form considered above, so that \( \mathbb{E}[P_0] \) may be immediately evaluated using (3.2) with \( m = d + 1 \). The MLMC method may be applied to (3.4) with \( \hat{P} \) using any discretization available. This approach will not change the resulting cost scaling, but will reduce the cost by a constant factor that may be significant. We demonstrate in section 7 via numerical experiments that these savings are frequently considerable.

This method of using Ito’s lemma to linearize the payoff function - we refer to this henceforth as the Ito linearization technique - does have two drawbacks. The first, and most serious, is that two continuous derivatives are required of \( P \) for Ito’s lemma to apply. In finance the payoff frequently has a discontinuity in the first derivative - e.g. European options - or even in the function itself - e.g. digital options. Ito linearization in its present form is not useful for these problems.

In many applications though, there are many natural payoffs with sufficient regularity. We have already noted that in chemical kinetics a simple linear payoff function is of interest. One may also wish to compute the covariances of the chemical concentrations, which may be computed from the means and the payoffs \( P(S) = S_i S_j \) for each \( i, j \), which of course have the necessary smoothness.

The second drawback is that (3.4) is a \((d + 1)\)-dimensional system, while (1.1) is only \( d \)-dimensional. Each time-step of (3.4) is thus slightly more expensive - by a factor of roughly \((d + 1)/d\) - than a corresponding time-step of (1.1). In numerical tests, we find that the savings at the base level more than compensate for this added expense.

4. Approximate Milstein for MLMC. We now turn to the derivation of an approximate version of the Milstein discretization that achieves \( O(\varepsilon^{-2}) \) cost scaling in arbitrary dimension. There are several observations that make this possible, the first of which is that when estimating \( \mathbb{E}[P_l - P_{l-1}] \), the discretizations used to compute \( P_l \) and \( P_{l-1} \) need not be identical for the reformulated problem (3.4).

To clarify this point, let us assume we have two discretizations. Given the same \( h \) and \( \Delta W \), the ‘fine’ discretization yields the payoff \( P_f \) while the ‘coarse’ one yields \( P_c \). We have the following generalization of (2.1):

\[
\mathbb{E}[P^L_f] = \mathbb{E}[P^f_0] + L \sum_{l=1} \left\{ \mathbb{E}[P^f_l - P^c_l] + \mathbb{E}[P^c_l - P^f_{l-1}] \right\}.
\] (4.1)

In the methods of Giles, it is required that

\[
\mathbb{E}[P_f] = \mathbb{E}[P^f_{l-1}]
\] (4.2)

for some large class of functions \( P \), so that the second term in the sum in (4.1) is identically zero and (4.1) reduces to (2.1). This requires that \( S^f_{h_l}(T) \) and \( S^c_{h_l}(T) \) be identically distributed, which in turn requires that the discretizations used at the fine and coarse levels be at least very nearly identical.

However, when solving the reformulation afforded by Ito’s lemma in the previous section, we may rewrite (4.1) as

\[
\mathbb{E}[S^L_{d+1}] = \mathbb{E}[S^0_{d+1}] + L \sum_{l=1} \left\{ \mathbb{E}[S^f_{d+1} - S^c_{d+1}] + \mathbb{E}[S^c_{d+1} - S^f_{d+1}] \right\},
\] (4.3)
where \( S^{f,l} \) is the result of the ‘fine’ discretization with time-step \( h_1 \), and similarly for \( S^{c,l} \). This now reduces to the analogue of (2.1) if

\[
\mathbb{E}[S^{c,l}] = \mathbb{E}[S^{f,l-1}].
\]  

(4.4)

This condition is actually more than is necessary - we only need the expectations of the last components to match - but there will be no additional difficulty in enforcing this condition. Being constrained by (4.4) instead of (4.2) creates considerable freedom in choosing different fine and coarse discretizations. We leverage this freedom extensively in the remainder of this section.

In what follows, we develop an ‘approximate Milstein’ method, whose intended application is MLMC methods applied to the modified SDE (3.4), as it takes advantage of this system’s linear payoff function. We will, however, denote the solution of the SDE by \( S \) - rather than \( S^{c,l} \) - to emphasize the generality of the specific results. It is only their application to MLMC that requires the modified SDE.

We begin by establishing some notation: define

\[
D^{l}_i(S, t, h, \Delta W_n) \equiv a_i(S, t)h + \sum_{j=1}^{D} b_j(S, t) \Delta W_{j,n} + \sum_{j,k=1}^{D} h_{ijk}(S, t)(\Delta W_{j,n} \Delta W_{k,n} - \Omega_{jk}h),
\]

(4.5)

\[
D^{f}_i(S_1, S_2, t, h, \delta W_n, \delta W_{n+\frac{1}{2}}) \equiv a_i(S_1, t)\Delta t + \sum_{j=1}^{D} b_j(S_2, t) \Delta W_{j,n} + \delta W_{j,n \Delta W_{n+\frac{1}{2}}} + \delta W_{j,n+\frac{1}{2} \delta W_{k,n}}
\]

where \( \Delta W_n \) is a vector in \( \mathbb{R}^D \) whose \( j^{th} \) component is \( \Delta W_{j,n} = \delta W_{j,n} + \delta W_{j,n+\frac{1}{2}} \). The analysis is simpler when \( M = 2 \), so we proceed with that case initially and generalize to arbitrary \( M \) in section 4.3. Fix \( l \) and set \( \delta t = h_1 \), \( \Delta t = 2\delta t = h_{1-1}, t_n = n\Delta t \).

4.1. Review of Antithetic Method. Because the method we develop here is closely related to the antithetic method of [6], we first state and review that algorithm. In our notation, the antithetic scheme may be written as

\[
S^{f,l}_{n+1} = S^{f,l}_{n+\frac{1}{2}} + D^{f}(S^{f,l}_{n+\frac{1}{2}}, t_{n+\frac{1}{2}}, \delta t, \delta W_{n+\frac{1}{2}}), \quad S^{f,l}_{n+\frac{1}{2}} = S^{f,l}_{n} + D^{f}(S^{f,l}_{n}, t_{n}, \delta t, \delta W_{n})
\]

\[
S^{c,l}_{n+1} = S^{c,l}_{n+\frac{1}{2}} + D^{c}(S^{c,l}_{n+\frac{1}{2}}, t_{n+\frac{1}{2}}, \delta t, \delta W_{n}), \quad S^{c,l}_{n+\frac{1}{2}} = S^{c,l}_{n} + D^{c}(S^{c,l}_{n}, t_{n}, \delta t, \delta W_{n+\frac{1}{2}})
\]

(4.7)

where \( D^{f} \) is the vector whose \( i^{th} \) component is \( D^{f}_i \), and the fine payoff is set to

\[
P^{f}_i = \frac{1}{2} \left( P(S^{f}_i) + P(S^{c}_i) \right).
\]

(4.8)

Meanwhile, the coarse evolution is given by

\[
S^{c,l}_{n+1} = S^{c,l}_{n} + D^{c}(S^{c,l}_{n}, t_{n}, \Delta t, \Delta W_{n}),
\]

(4.9)

with the coarse payoff set to \( P^{c}_i = P(S^{c}_i) \).

Notice that the evolution equations for \( S^{f}_i \) and \( S^{c}_i \) are identical except that the Brownian steps \( \delta W_{n} \) and \( \delta W_{n+\frac{1}{2}} \) have been switched. This has the effect of canceling the leading order
contribution of the Lévy areas when the two are averaged, as in (4.8). This cancellation makes the $V_l$ scale like $O(h_l^2)$ for twice differentiable payoffs and like $O(h_l^{2-\delta})$ for any $\delta > 0$ when the payoff is Lipschitz, only non-differentiable on a set of measure zero, and the solution is unlikely to be near this set in a certain sense (see [6] for details). The scheme thus achieves the $O(\varepsilon^{-2})$ cost scaling in both cases.

The scheme has two primary drawbacks: 1) it requires twice as much effort to generate $P^f_l$ due to the need to evolve the antithetic variable $S_{\ast}^n$ as an Euler based multilevel scheme, and 2) its derivation in [6] is restricted to $M = 2$. In our development, we offer a slight improvement to 1) by moving the doubled effort to the coarse level, which is cheaper by a factor of $M$. Moreover, we generalize both our method and the antithetic method to $M > 2$ in sections 4.3 and 5.

4.2. Approximate Milstein for $M = 2$. We consider the following pair of schemes for $S^f$ and $S^c$:

$$S_{n+\frac{1}{2}}^{f,l} = S_n^{f,l} + D^f(S_n^{f,l}, t_n, \delta t, \delta W_n),$$ (4.10)

$$S_{n+1}^{c,l} = S_n^{c,l} + D^c(S_n^{\ast,c,l}, S_n^{c,l}, t_n, \Delta t, \delta W_n, \delta W_{n+\frac{1}{2}}),$$ (4.11)

where $S^{\ast,l}$ is given by

$$S_{n+1}^{\ast,l} = S_n^{\ast,l} + D^f(S_n^{\ast,l}, t_n, \Delta t, \Delta W).$$ (4.12)

We set $P^f_l = P(S^f_l)$ and $P^c_l = P(S^c_l)$.

It is worth clarifying that in this description the number $n$ always indexes the number of level-$l$ coarse time steps taken. This is equal to the number of level-$(l-1)$ fine time steps taken, so that number is also indexed by $n$. By writing (4.10) the way we have, we ensure that $S_{n}^{f,l}$, $S_{n}^{c,l}$, and $S_{n-1}^{f,l}$ are all approximations to $S(n \Delta t)$ for each whole number $n$. In addition to $n = 0, 1, 2, 3, \ldots$, we have definitions of $S_{n}^{f,l}$ at $n = 1/2, 3/2, 5/2, \ldots$, but this fact will not concern us.

In the remainder of this section, we state and prove results that establish first (4.4) and then (2.19) with $\beta = 2$ for this pair of discretizations. The more technical proofs are confined to appendices.

4.2.1. Equal Expectations. Theorem 4.1 (Equal Expectations). For $S^f$ and $S^c$ as defined in (4.10)-(4.12), we have

$$E[S_{n}^{f,l-1}] = E[S_{n}^{c,l}]$$ (4.13)

for each $n = 0, 1, 2, 3, \ldots$

Proof. At the $(l-1)^{st}$ level, we have

$$S_{n+1}^{f,l-1} = S_n^{f,l-1} + D^f(S_n^{f,l-1}, t_n, \Delta t, \Delta W).$$ (4.14)
If we subtract (4.11) from (4.14), we find

\[
S_{n+1}^{f,l} - S_{n+1}^{c,l} = S_n^{f,l} - S_n^{c,l} \\
+ \{a(S_n^{f,l-1}) - a(S_n^{c,l-1})\} \Delta t \\
+ \sum_{j=1}^{D} \{b_j(S_n^{f,l-1}) - b_j(S_n^{c,l-1})\} [\Delta W_{j,n}] \\
+ \sum_{j,k=1}^{D} \{h_{jk}(S_n^{f,l-1}) - h_{jk}(S_n^{c,l-1})\} [\Delta W_{j,n} \Delta W_{k,n} - \Omega_{jk} \Delta t] \\
- \sum_{j,k=1}^{D} \{h_{jk}(S_n^{c,l-1})\} \left[\delta W_{j,n} \delta W_{k,n+\frac{1}{2}} - \delta W_{j,n+\frac{1}{2}} \delta W_{k,n}\right],
\]

(4.15)

where \(a\) is the vector whose \(i\)th component is \(a_i\), and analogously for \(b_j\) and \(h_{jk}\).

We look at (4.15) term by term. In the last three lines, the term in square brackets has zero expectation and is independent of the term in curly braces - this follows from the fact that each Brownian increment is independent of all those before it. Therefore, each of these lines has vanishing expectation. In the second line, \(S_n^{f,l-1} \Delta W_{j,n}\) and \(S_n^{c,l} \Delta W_{k,n}\) are identically distributed for each \(n\) because they are approximated by exactly the same method - compare (4.14) and (4.12) - so the term in curly braces has zero expectation. Therefore, if we take the expectation of (4.15), everything vanishes except the first line. Thus, we have

\[
E[S_{n+1}^{f,l}] = E[S_{n+1}^{c,l}].
\]

(4.16)

Since the coarse and fine approximations start at the same initial condition, the difference in expectation is zero for \(n = 0\), and (4.16) guarantees that this remains the case for all integer \(n > 0\).

**COROLLARY 4.2.** With the same definitions as in Theorem 4.1,

\[
E[S_n^{f,l}] = E[S_n^{c,l}].
\]

(4.17)

**Proof.** Since \(S_n^{f,l} \Delta W_{j,n}\) and \(S_n^{c,l} \Delta W_{k,n}\) are identically distributed, they have the same expectation, so this follows directly from Theorem 4.1.

4.2.2. Variance Scaling. Before establishing the variance scaling (2.19) required for the \(e^{-2}\) cost scaling, we need three lemmas. The first establishes that the weak difference between coarse and starred approximations is \(O(\Delta t)\), while the last two are convenient rewritings of the fine and coarse discretizations.

**LEMMA 4.3.** The weak difference between \(S_n^{c,l}\) and \(S_n^{f,l-1}\) is \(O(\Delta t)\). That is, for sufficiently differentiable \(f : \mathbb{R}^d \to \mathbb{R}\), we have

\[
|E[f(S_n^{c,l})] - E[f(S_n^{f,l-1})]| = O(\Delta t)
\]

(4.18)

for all \(n \leq T/\Delta t\).

**Proof.** See appendix A.

In the proof of lemma 4.3 above, we require that \(f\) have four continuous and bounded derivatives. Elsewhere in our development, the payoff and SDE coefficients are only required to possess two derivatives, and in practice the scaling predicted by lemma 4.3 is observed in these
cases as well. It seems likely that by following [13], lemma 4.3 could be reestablished for \( f \) merely Hölder continuous, but such an exercise is beyond the scope of this paper.

**Lemma 4.4.** The fine discretization (4.10) can be rewritten as

\[
S_{i,n+1}^f = S_{i,n}^f + D^f_i(S_{n}^f, S_{n}^f, t_n, \Delta t, \delta W_n, \delta W_{n+\frac{1}{2}}) + M_{i,n}^f + N_{i,n}^f
\]

(4.19)

where \( \mathbb{E}[M_{i,n}^f] = 0 \) and

\[
\mathbb{E}\left[ \max_{n \leq N} \|M_n^f\|^p \right] = O\left( \Delta t^{3p/2} \right), \quad \mathbb{E}\left[ \max_{n \leq N} \|N_n^f\|^p \right] = O\left( \Delta t^{2p} \right)
\]

(4.20)

for any integer \( p \geq 2 \).

**Proof.** The lemma and proof are identical to Lemma 4.7 and its proof in [6].

**Lemma 4.5.** The coarse discretization (4.11) may be rewritten as

\[
S_{i,n+1}^c = S_{i,n}^c + D^c_i(S_{n}^c, S_{n}^c, t_n, \Delta t, \delta W_n, \delta W_{n+\frac{1}{2}}) + M_{i,n}^c + N_{i,n}^c
\]

(4.21)

where \( \mathbb{E}[M_{i,n}^c] = 0 \) and

\[
\mathbb{E}\left[ \max_{n \leq N} \|M_n^c\|^p \right] = O\left( \Delta t^{3p/2} \right), \quad \mathbb{E}\left[ \max_{n \leq N} \|N_n^c\|^p \right] = O\left( \Delta t^{2p} \right).
\]

(4.22)

**Proof.** Simple algebra shows that

\[
S_{i,n+1}^c = S_{i,n}^c + D^c_i(S_{n}^c, S_{n}^c, t_n, \Delta t, \delta W_n, \delta W_{n+\frac{1}{2}})
\]

\[
+ (a_i(S_{n}^c) - a_i(S_{n}^c)) \Delta t,
\]

(4.23)

so the lemma reduces to analyzing the second line of this expression. Define \( \Delta a_{i,n} = a_i(S_{n}^c) - a_i(S_{n}^c) \), and write the term in question as

\[
\Delta a_{i,n} \Delta t = \mathbb{E}[\Delta a_{i,n}] \Delta t + \{\Delta a_{i,n} - \mathbb{E}[\Delta a_{i,n}]\} \Delta t.
\]

(4.24)

By Lemma 4.3 above (which we again note uses four derivatives), we have \( \mathbb{E}[\Delta a_{i,n}] = O(\Delta t) \), so that the first term is \( O(\Delta t^2) \). Thus, we define \( N_{i,n}^c = \mathbb{E}[\Delta a_{i,n}] \Delta t \).

The second term on the right of (4.24) clearly has zero expectation. The first term in the curly braces is \( O(\sqrt{\Delta t}) \) by strong convergence of both schemes (4.11) and (4.12), and the second term in the curly braces is \( O(\Delta t) \) as before, so their difference is \( O(\sqrt{\Delta t}) \). Thus, the second term on the right of (4.24) is \( O(\Delta t^{3/2}) \), so we define it to be \( M_{i,n}^c \).

Finally, we are ready to prove the desired scaling of the variances:

**Theorem 4.6 (Variance Scaling).** Assume the \( a_i \) have four continuous bounded derivatives, \( b_{ij} \) are twice continuously differentiable with both derivatives uniformly bounded, and that the \( h_{ijk} \) have uniformly bounded first derivatives. Then, for the pair of fine and coarse discretizations (4.10), (4.12), we have (2.19) with \( \beta = 2 \). In fact, we have the stronger statement

\[
\mathbb{E}\left[ \max_{n \leq N} \|S_n^f - S_n^c\|^2 \right] = O(\Delta t^2),
\]

(4.25)

where \( N = T/\Delta t \).

**Proof.** See appendix B.
4.3. Generalization to $M > 2$. Thus far, we’ve developed the approximate Milstein method assuming that the difference in time step at adjacent levels (refinement factor) $M$ is equal to two. This assumption is also made in Giles’ development of the antithetic method. However, in [5] Giles argues that the optimal refinement factor is near seven for an Euler-based multilevel scheme, and a similar argument for Milstein shows the optimal choice to be near four.

Following [5], the latter argument proceeds as follows: Noting that $P_{f}^{l} - P_{c}^{l} = (P_{f}^{l} - P) - (P_{c}^{l} - P)$, where $P$ is the exact mean payoff, we infer that

$$ (M - 1) k h_{l}^{2} \leq V_{l} \leq (M + 1) k h_{l}^{2}, $$

for some constant $k$, where the lower and upper bounds correspond to perfect correlation and anti-correlation between $P_{f}^{l} - P$ and $P_{c}^{l} - P$. Supposing for simplicity that the actual variance is approximately the geometric mean of the two extremes, we have

$$ V_{l} \approx (M^{2} - 1) k h_{l}^{2}. $$

Substituting this expression into the cost formula (2.14), we have (ignoring for clarity the fact that $V_{0}$ need not obey any scaling law)

$$ K \propto \varepsilon^{-2} \frac{(M^{2} - 1)(1 + M^{-1})}{(\sqrt{M} - 1)^{2}}, $$

which for fixed $\varepsilon$ has its minimum for $M$ between 4 and 5. There is thus motivation to study arbitrary $M$.

Notationally, moving to arbitrary $M$ changes (4.10) to read

$$ S_{f,l}^{i,n+1} = S_{f,l}^{i,n} + D_{f}^{l}(S_{f,l}^{i,n}, t_{n}, \Delta t, \delta W_{j,n}), $$

and we set

$$ \Delta t = M \delta t, \quad \Delta W_{j,n} = \sum_{m=0}^{M-1} \delta W_{j,n+m \Delta t}. $$

To see how to change (4.11), we present the following generalization of Lemma 4.4:

**Lemma 4.7.** The fine discretization (4.29) can be rewritten as

$$ S_{f,l}^{i,n+1} = S_{f,l}^{i,n} + D_{f}^{l}(S_{f,l}^{i,n}, t_{n}, \Delta t, \Delta W_{n}), $$

$$ - \sum_{j,k=1}^{D} h_{ijk} (S_{f,l}^{i,n}) (A_{jk,n} - A_{kj,n}) $$

$$ + M_{f}^{i,n} + N_{f}^{i,n}, $$

where $E[M_{f}^{i,n}] = 0$ and

$$ E \left[ \max_{n \leq N} \| M_{f}^{i,n} \|^{p} \right] = O \left( \Delta t^{3p/2} \right), \quad E \left[ \max_{n \leq N} \| N_{f}^{i,n} \|^{p} \right] = O \left( \Delta t^{2p} \right), $$

and $A_{jk,n}$ is defined by

$$ A_{jk,n} \equiv \sum_{m=1}^{M-1} \left( \delta W_{k,n+m \Delta t} \sum_{q=0}^{m-1} \delta W_{j,n+q \Delta t} \right). $$
Proof. See appendix C.

We note that, as described in [2], \((A_{jk,n} - A_{kj,n})\) is a quadrature scheme for the Levy area \(A_{jk,n}\) obtained by dividing up the time step \(\Delta t\) into \(M\) equal parts. As noted in [2], computing the \(A_{jk,n}\) as written can be done in \(O(M)\) operations, even though the double sum contains \(O(M^2)\) terms.

With Lemma 4.7 in hand, we can generalize the coarse discretization to arbitrary \(M\). We define

\[ D_{c,M}^n(S_1, S_2, t, \Delta t, \delta W_{j,n+\frac{m}{M}}) \equiv a_i(S_1, t) \Delta t + \sum_{j=1}^{D} b_{ij}(S_2, t) \Delta W_{j,n} \]

\[ + \sum_{j,k=1}^{D} h_{ijk}(S_2, t)(\Delta W_{j,n} \Delta W_{k,n} - \Omega_{jk} \Delta t - A_{jk,n} + A_{kj,n}). \]

(4.34)

Meanwhile, the starred discretization \((4.12)\) remains unchanged. With these definitions and Lemma 4.7 in place of Lemma 4.4, the proofs of Theorems 4.1 and 4.6 generalize to arbitrary \(M\) with only straightforward notational changes.

5. Generalization of Antithetic Method to \(M > 2\). In this section, we demonstrate that the antithetic method \((4.7)\) may be straightforwardly generalized to arbitrary \(M\). In particular, the same variance scaling can be achieved by using an antithetic variable for which the order of the \(M\) Brownian fine sub-steps of each coarse Brownian step is completely reversed.

More explicitly, set \(\bar{m} = (M - 1) - m\), then we rewrite \((4.7)\) as

\[ S_{f,l}^n + m + 1 = S_{f,l}^n + \sum_{i=1}^{D} D_i^f(S_{f,l}^n, t, \Delta t, \delta W), \]

\[ S_{a,l}^n + m + 1 = S_{a,l}^n + \sum_{i=1}^{D} D_i^f(S_{a,l}^n, t, \Delta t, \delta W), \]

(5.1)

for each \(m = 0, 1, 2, ..., M - 1\). As before, \(S_{f,l}^n\) and \(S_{a,l}^n\) are identical except that the Brownian motions are indexed by \(\bar{m}\) rather than \(m\) for the antithetic variable. By applying Lemma 4.7 to the antithetic variable \(S_{a,l}^n\), we find that its discretization can be rewritten as

\[ S_{a,l}^n + 1 = S_{a,l}^n + \sum_{i=1}^{D} h_{ijk}(S_{a,l}^n) (\bar{A}_{jk,n} - \bar{A}_{kj,n}) \]

\[ + M_{a,l}^n + N_{a,l}^n, \]

(5.2)

where \(M_{a,l}^n\) and \(N_{a,l}^n\) obey the same scalings as \(M_{f,l}^n\) and \(N_{f,l}^n\). The quantity \(\bar{A}_{jk,n}\) is defined by

\[ \bar{A}_{jk,n} \equiv \sum_{m=1}^{M-1} \left( \delta W_{k,n+\bar{m}/M} \sum_{q=0}^{m-1} \delta W_{j,n+q/M} \right), \]

(5.3)

with \(\bar{m}\) as defined before and \(\bar{q} = (M - 1) - q\). The improved variance scaling then results from the following lemma, originally published in [7] with different notation and reprinted here for clarity.

Lemma 5.1.

\[ \bar{A}_{jk,n} = A_{kj,n}. \]

(5.4)
Proof. The proof amounts to a computation:

\[
\bar{A}_{j,k,n} = \sum_{m=1}^{M-1} \sum_{q=0}^{M-1} \delta W_{k,n+m/M} \delta W_{j,n+q/M} \\
= \sum_{m=0}^{M-1} \sum_{q=0}^{M-1} \delta W_{k,n+m/M} \delta W_{j,n+q/M} \\
= \sum_{q=1}^{M-1} \sum_{m=0}^{q-1} \delta W_{k,n+m/M} \delta W_{j,n+q/M} \\
= \bar{A}_{k,j,n}.
\] (5.8)

Notice that (5.6) comes from simply switching the order of summation, while (5.7) results from rewriting the sums in terms of \(\bar{m}\) and \(\bar{q}\).

This lemma implies that, if we define \(\bar{S}_{f,l} = \frac{1}{2}(S_{f,l} + S_{a,l})\), we can write

\[
\bar{S}_{f,l}^{i,n+1} = \bar{S}_{f,l}^{i,n} + D_{f}^{i} (\bar{S}_{f,l}^{n}, t_{n}, \Delta t, \Delta W_{n}) + M_{i,n} + N_{i,n},
\] (5.9)

where \(M_{i,n}\) and \(N_{i,n}\) obey the same scalings as usual. This is in direct analogue to Lemma 4.9 in [6], and its proof is identical, so we omit it. This, in turn, allows one to show the analogue of Theorem 4.10 in [6] and Theorem 4.6 in the present work:

\[
E \left[ \max_{n \leq N} \| \bar{S}_{f,l}^{i,n} - S_{c,l}^{i,n} \|^{p} \right] = O(\Delta t^{p})
\] (5.10)

for each \(p \geq 2\). The desired variance scaling finally follows directly from Lemma 2.2 in [6], which we restate without proof here:

\[
E \left[ \frac{1}{2} \left( P(S_{f}) + P(S_{a}) \right) - P(S_{c}) \right]^{p} \leq \frac{1}{2} \left[ \| S_{f} - S_{c} \|^{p} \right] + \frac{1}{2} \left[ \| S_{a} - S_{c} \|^{2p} \right],
\] (5.11)

where we’ve omitted the \(l\) superscripts and assumed that \(P\) had two continuous and bounded derivatives. The first term on the right is \(O(\Delta t^{p})\) by (5.10), and the second term has the same scaling by strong convergence: indeed, \(S_{f} - S_{a} = (S_{f} - S_{c}) - (S_{a} - S_{c})\), and each of the latter two terms is \(O(\Delta t^{1/2})\) by the strong convergence of the discretization.

Finally, we note that Theorem 5.2 in [6] may be applied - unchanged - to these results, demonstrating that \(V_{l} = O(h_{l}^{3/2-\delta})\) for any \(\delta > 0\) when \(P\) is merely Lipschitz - so long as the set \(A\) on which \(P\) is non-differentiable is measure zero and

\[
P\left( \min_{y \in A} \| S(T) - y \| \leq \varepsilon \right) \leq c \varepsilon
\] (5.12)

for some \(c > 0\) and for all \(\varepsilon > 0\).

This completes the generalization of the antithetic method to arbitrary \(M\).

6. Summary and Implementation. We present an outline of the modified MLMC method including the Ito linearization at the lowest level. This is to be compared to the analogous algorithm in [5]. This may be used with any discretization we choose - including the approximate
Milstein method introduced in section 4 and the generalized antithetic method in section 5 - so long as the discretization is first order in the weak sense. We denote by $\beta$ the expected scaling of the $V_l$. That is, $\beta = 2$ for approximate Milstein or generalized antithetic (assuming $P$ has the necessary regularity), and $\beta = 1$ for Euler.

1. Set

$$E[P_0] = P(S_0) + \alpha_{d+1}(S_0)T,$$

(6.1)

where $\alpha_{d+1}$ is as defined in (3.5).

2. Begin with $L = 1$.

3. Estimate $V_L$ and $\hat{Y}_L$ using an initial $N_i^L$ samples, defined by

$$N_i^L = \begin{cases} 400 & : L = 1 \\ M - (\beta + 1)/2N_{L-1} & : L > 1 \end{cases}$$

(6.2)

4. Set $N_l$ according to

$$N_l = \left\lceil \frac{2}{\epsilon^2} \sqrt{V_l h_l} \left( \sum_{i=1}^{L} \sqrt{V_i h_i} \right) \right\rceil$$

(6.3)

for each $l = 1, 2, ..., L$, as per (2.13).

5. Generate additional samples at each level as needed for new $N_l$, using discretization of your choice to generate approximate solutions of (3.4). Use these samples to update the estimates of the $V_l$ and $\hat{Y}_l$.

6. If $L < 2$ or

$$\max \left\{ \left| \hat{Y}_L \right|, M^{-1} \left| \hat{Y}_{L-1} \right| \right\} > \epsilon \sqrt{2},$$

(6.4)

let $L \rightarrow L + 1$ and go to step 3. Else, end with payoff estimate of

$$\hat{P}_L = E[P_0] + \sum_{l=1}^{L} \hat{Y}_l.$$  

(6.5)

The inequality (6.4) is the convergence criterion used in [5]. It determines the algorithm to be converged if the bias error is estimated to be at most $\epsilon/\sqrt{2}$ when using either of the two finest levels in the estimation.

Equation (6.2) in step 3 is worthy of elaboration. When $L = 1$, we have no information about how many samples we might expect to need, so we pick an arbitrary large number - we find that 400 works well in our test cases, but the number will be problem dependent. However, for $L > 1$, the expected scaling of the $V_l$ allows us to estimate the number of samples needed at the $L^{th}$ level, using

$$N_L \propto \sqrt{V_L h_L} = M^{-(\beta+1)/2} \sqrt{V_{L-1} h_{L-1}} \propto M^{-(\beta+1)/2} N_{L-1}.$$

(6.6)

Particularly at large $L$, when there is relatively little change to the sum in (6.3) as a result of incrementing $L$, (6.2) is thus a good estimate of $N_L$ as it will be set in step 4. This is preferable to the technique used in [5] - where $N_1^L = 10^4$, regardless of $L$ - because we avoid wasteful sampling at the high levels where it may be the case that $N_l \ll 10^4$.  

16
7. Numerical Results. For our numerical tests, we apply our methods to the Heston model - a financial stochastic volatility model [11] - given by

\[
\begin{align*}
\frac{dS_1}{S_1} &= \kappa(\theta - S_1) \, dt + \xi \sqrt{S_1} \, dW_1, \\
\frac{dS_2}{S_2} &= \mu S_2 \, dt + \eta \sqrt{S_1 S_2} \, dW_2,
\end{align*}
\] (7.1)

where \( S_1 \) represents the volatility and \( S_2 \) the asset price. Throughout our tests, we set the constants \( \theta = \mu = \xi = \kappa = 1 \) and \( \eta = 1/4 \). We find that this choice of constants allows us to conduct tests with relatively large \( L \), where the benefits of MLMC are most obvious. We set \( S_0 = (0.5, 1) \), \( \Omega_{jk} = \delta_{jk} \), and \( T = 0.125 \) - a short time simulation allows us to push the limits of the accuracy of the method. Notice that, for this system, \( h_{221} = \eta S_2/4 \), so that the Milstein discretization does in fact feature \( \text{Lévy} \) areas.

Notice also that the coefficients of this SDE system do not have uniformly bounded derivatives - namely, the \( b \)'s have divergent derivative at \( S_1 = 0 \) - so the assumptions for all of the foregoing results do not hold (see e.g. theorem 4.6), nor do those for standard weak convergence results. However, we have constructed the system so that \( S_1 \) is extremely unlikely to approach zero, so that in practice all derivatives are essentially bounded. Indeed, we find excellent agreement between the theory and numerical results.

We test several distinct new MLMC variants:
1. Generalized antithetic method for a payoff with discontinuous derivative
2. Ito linearization technique for smooth payoff using:
   (a) Euler discretization
   (b) Approximate Milstein discretization
   (c) Generalized antithetic discretization
Each of these is compared to the original Euler-based MLMC method introduced in [5] and the original antithetic method proposed in [6].

In all tests, the computational cost is estimated by the total number of time steps taken, weighted by the dimension of the system being solved. That is, for the standard Euler MLMC method, we set

\[
K^e = \sum_{l=0}^{L} N_l (h_l^{-1} + h_{l-1}^{-1}),
\] (7.2)

while for the antithetic and generalized antithetic methods (without Ito linearization), we set

\[
K^a = \sum_{l=0}^{L} N_l (2h_l^{-1} + h_{l-1}^{-1})
\] (7.3)

to account for the added computation of the antithetic variable \( S^{*\cdot l} \). For the approximate Milstein method, we set

\[
K^m = \frac{d + 1}{d} \sum_{l=0}^{L} N_l (h_l^{-1} + 2h_{l-1}^{-1}),
\] (7.4)

accounting both for the added cost of computing \( S^{*\cdot l} \) and the extra dimension. When Ito linearization is applied to Euler and antithetic methods, we simply scale (7.2) and (7.3) by the factor \((d + 1)/d\) and note that \( N_0 = 1 \).

The counting of time steps is the standard method of estimating computational complexity in the MLMC literature - it is used in [5, 6], among others. However, we note in our discussion at the end of this section that there are other relevant considerations as well.
7.1. Generalized antithetic test. We use a standard European option as the payoff function:

$$P(S) = \max\{0, S_2(T) - S_2(0)\}.$$ \hspace{1cm} (7.5)

Figure 2 demonstrates both the improved variance scaling (left) and corresponding reduction in computational cost (right) afforded by antithetic methods. As predicted in section 4.3, \(M\) equal to 4 or 5 minimizes the computational cost. The reduction in cost gained by the generalization to arbitrary \(M\) is comparable to that gained by moving from Euler to the original antithetic method. We note that the cost is reduced for larger \(M\) in spite of the fact that increasing \(M\) actually increases the individual \(V_i\). This because at large \(M\), we need fewer levels, so the sum of the variances is smaller because there are fewer terms in the sum.

7.2. Ito linearization and approximate Milstein test. We again use the Heston model with the same parameters but change the payoff function to

$$P(S) = \sin S_2,$$ \hspace{1cm} (7.6)

which of course has the requisite regularity. Figure 3 shows the results, plotting computational cost against error tolerance \(\varepsilon\). The Euler, antithetic, and approximate Milstein discretizations are all plotted, each with and without Ito linearization.

For each discretization, Ito linearization improves the efficiency of the scheme, by an order of magnitude in some cases. Note that in the case of approximate Milstein, the use of Ito's lemma is required for the expectations at adjacent levels to match - see the beginning of section 4 for details - so that the exclusion of Ito linearization is somewhat artificial. This accounts for the disproportionately large expense when Ito linearization is excluded, since the use of Ito's lemma increases the dimension of the system we solve.

The most efficient scheme of those tested is approximate Milstein with Ito linearization, although the advantage over generalized antithetic with Ito linearization is relatively small. As expected based on fig. 1, the approximate Milstein and antithetic methods benefit more from Ito linearization than does Euler because a larger fraction of the work is concentrated at the base level.
We note that neither the approximate Milstein nor the antithetic method produces a completely flat curve when we plot $\varepsilon^2 K$ against $\varepsilon$, as would be predicted by the asymptotic analysis. However, the observations are perfectly consistent with (2.14): the finite sum is bounded by an infinite sum, but is not itself constant in $\varepsilon$. We in fact only expect a flat curve as $L \to \infty$, and all of the tests conducted here have $L \leq 7$.

Finally, we note that in many cases we get a better-than-expected cost scaling for large $\varepsilon$. This is a result of setting a fixed number of initial samples for $L = 1$ in (6.2) - and for $L = 0$ when Ito linearization is not used - which can turn out to be more than is actually necessary for large $\varepsilon$, making the cost artificially large. We leave this effect in the plots because it is a practical reality of MLMC.

7.3. Discussion. The techniques introduced in the present work have the common aim of optimizing MLMC simulations of SDEs. Since the standard MLMC algorithm with the Euler discretization already achieves a nearly optimal cost-to-error scaling, the savings are relatively modest - we rarely save more than a single order of magnitude. While these improvements are hardly negligible, they are small enough that some care in analyzing all sources of computational cost and coding optimization is justified.

In particular, the estimation of computational cost by the total number of time steps taken ignores the fact that not all time steps have identical complexity. In the antithetic and approximate Milstein discretizations, there is an additional term to be computed at each time step. The dominant cost turns out to be the computation of the rank-3 tensor $h_{ijk}$, which requires $O(d^2 D^2)$ operations - the tensor has $d D^2$ elements, and the computation of each requires a sum of $d$ terms. In contrast, the dominant computation in an Euler time step is the matrix-vector product $\sum_j b_{ij} W_j$, which is $O(d D)$.

With this in mind, a fairer estimate of the computational cost for each method is

\[
K = \begin{cases} 
O \left( \varepsilon^{-2} (\log \varepsilon)^2 dD \right) & \text{: Euler} \\
O \left( \varepsilon^{-2} d^2 D^2 \right) & \text{: Antithetic & Approx. Milst.} 
\end{cases}
\]
Thus, the optimal discretization is problem dependent - for any fixed \( \varepsilon \), Euler with be optimal for some sufficiently large value of \( dD \), while antithetic/approximate Milstein is optimal for smaller values of \( dD \). In financial and chemical kinetic applications, \( d \) and \( D \) are frequently large - very possibly exceeding \( \log \varepsilon \).

This situation is, however, the worst case. Often, \( b_{ij} \) and/or \( h_{ijk} \) exhibit some form of sparsity, which the code may be written to exploit. It may be possible to leverage the relationship between \( b_{ij} \) and \( h_{ijk} \) to further accelerate their computation. This is an interesting area of future research.

8. Conclusions. In this paper we have introduced three related improvements to MLMC methods for SDEs. First, we have introduced the idea of Ito linearization, which makes the computation of the base level payoff essentially free, at the price of increasing the dimension of the SDE by one. Secondly, we have introduced an approximate Milstein discretization which, in conjunction with Ito linearization, achieves an \( O(\varepsilon^{-2}) \) cost scaling with slightly reduced cost compared to the antithetic method. Finally, we demonstrated that the antithetic method can be generalized to arbitrary \( M \) without introducing any additional antithetic paths.

The first two techniques are applicable only to payoff functions with two continuous derivatives. As such, they are of very limited use in financial applications, but are expected to be applicable to other fields - examples from chemical kinetics have been cited in the text. The generalized antithetic method, however, requires only a Lipschitz, piecewise smooth payoff, and may thus find applications in finance as well as other disciplines.

Each new method has been tested on a simple SDE system, and we find excellent agreement between the analysis and the numerical results. In the cases in which they are applicable, our new methods consistently outperform the present state-of-the-art.

Acknowledgements. The author is grateful to Russel Caflisch for numerous discussions that were instrumental in the development of the methods presented here. Also acknowledged are many helpful conversations with Mark Rosin regarding the theory and implementation of MLMC schemes in general. Additional thanks go to Andris Dimits, Bruce Cohen and Michael Giles.

Appendix A. Proof of Lemma 4.3. The argument here follows that in [18]. It proceeds in two stages: first, we express the total error as a sum of various local truncation errors, totaling \( O(\Delta t^{-1}) \) in number. Second, we show that each local truncation error is \( O(\Delta t^2) \).

Toward the first end, we introduce some notation for this proof not used elsewhere: Denote by \( S^{c,l}_n[x] \) that solution of the recursion equation (4.11) at time \( t_n \), which starts at \( x \) at time zero, and similarly for \( S^{*,l}_n[x] \). For this proof we will assume that the system is autonomous, so that

\[
S^{c,l}_{n+1}[x] = S^{c,l}_1 [S^{c,l}_n[x]].
\]  

(A.1)

and similarly for \( S^{*,l}_{n+1}[x] \). All of the arguments presented here generalize to the non-autonomous case, but the notation is much cleaner if autonomy is assumed. Define \( g^{c}_n(x) = E f(S^{c,l}_n[x]) \) and similarly for \( g^{*}_n(x) \). Then, leveraging (A.1), we have

\[
g^{c}_{n+1}(x) - g^{*}_{n+1}(x) = E f\left(S^{c,l}_1 [S^{c,l}_n[x]]\right) - E f\left(S^{*,l}_1 [S^{*,l}_n[x]]\right)
= E g^{c}_1 \left(S^{c,l}_n[x]\right) - E g^{*}_1 \left(S^{*,l}_n[x]\right)
= E \left\{ g^{c}_1 \left(S^{c,l}_n[x]\right) - g^{*}_1 \left(S^{c,l}_n[x]\right) \right\} + E \left\{ g^{*}_1 \left(S^{*,l}_n[x]\right) - g^{*}_1 \left(S^{*,l}_n[x]\right) \right\}.
\]  

(A.2)

The first expectation in the third line is a local truncation error: it’s the difference in \( f \) evaluated at the coarse and starred discretizations after one time step, when both started at the same place; namely, \( S^{c,l}_n[x] \). Due to the nature of the coarse and starred discretizations, the function \( g^{c}_1 \) is as
smooth as \( f \), so the second expectation is of the same sort we’re trying to bound, but one time step earlier than where we started.

If we define \( \varepsilon_n[f] = E[f(S_{n,t}^c|x)] - E[f(S_{n,t}^e|x)] \), then (A.2) reads

\[
\varepsilon_{n+1}[f] = E\{g'_i(S_{n,t}^c|x) - g'_i(S_{n,t}^e|x)\} + \varepsilon_n[g'_i].
\] (A.3)

In the same way, we may go on to derive

\[
\varepsilon_n[g'_i] = E\{h'_i(S_{n,t}^c|x) - h'_i(S_{n,t}^e|x)\} + \varepsilon_{n-1}[h'_i],
\] (A.4)

for appropriate definitions of \( h'_i \) and \( h'_i \). Iterating this process, we find

\[
\varepsilon_n[f] = \sum_{k=1}^{n-1} E\left\{ h_{k,c}^{i} \left( S_{k}^{c,l}[x] \right) - h_{k,s}^{i} \left( S_{k}^{e,l}[x] \right) \right\}
\] (A.5)

for some sequences of functions \( \{ h_{k,c}^{i} \} \) and \( \{ h_{k,s}^{i} \} \), each of which is as smooth as \( f \) and represents the error in the given function after a single time step (we’ve used the fact that \( S_{n,t}^{c,l} = S_{0,t}^{c,l} \)). This completes the first step of expressing the total error in terms of local truncation errors.

It just remains to show that each of these errors is \( O(\Delta t^2) \). We do this by Taylor expansion of \( f \). Suppose \( f \) has four continuous derivatives, so that we can write out its fourth order Taylor series:

\[
f \left( S_{1}^{c,l}[x] \right) = f(x) + \nabla f(x) \cdot D^c(x, x, t_0, \Delta t, \delta W_0, \delta W_2) \\
+ \frac{1}{2!} \nabla^2 f(x) : D^2 D^c + \frac{1}{3!} \nabla^3 f(x)(D^c)^3 + \frac{1}{4!} \nabla^4 f(\xi)(D^c)^4
\] (A.6)

for some \( \xi \) on the line between \( x \) and \( S_{1}^{c,l}[x] \), and \( D^c \) has the same arguments in all instances. A similar expression holds for \( f(S_{1}^{e,l}[x]) \). Subtracting the two expressions and taking expectations, we have

\[
E[f \left( S_{1}^{c,l}[x] \right)] - E[f \left( S_{1}^{e,l}[x] \right)] = \frac{1}{2!} \nabla^2 f(x) : E \left\{ D^c D^c - D^f D^f \right\} \\
+ \frac{1}{3!} \nabla^3 f(x) E \left\{ (D^c)^3 - (D^f)^3 \right\} \\
+ \frac{1}{4!} E \left\{ \nabla^4 f(\xi)(D^c)^4 \right\} - E \left\{ \nabla^4 f(\xi)(D^f)^4 \right\}.
\] (A.7)

Careful but straightforward examination of all the expectations in (A.7) reveals that the lowest order terms that don’t vanish in expectation are all \( O(\Delta t^2) \). This is true for any function as smooth as \( f \), and so is true of each \( h_{k,c}^{i} \) and \( h_{k,s}^{i} \) in (A.5). The number of terms in the sum in (A.5) is \( O(\Delta t^{-1}) \), so we have the desired result.

**Appendix B. Proof of Theorem 4.6** Using Lemmas 4.4 and 4.5 we may write

\[
S_{i,n}^{l} - S_{i,n}^{e} = \left( S_{i,n-1}^{l} - S_{i,n-1}^{e} \right) + \left[ a_{i}(S_{i,n-1}^{l}) - a_{i}(S_{i,n-1}^{e}) \right] \Delta t \\
+ \sum_{j=1}^{D} \left[ b_{ij}(S_{i,n-1}^{l}) - b_{ij}(S_{i,n-1}^{e}) \right] \Delta W_{j,n-1} \\
+ \sum_{j,k=1}^{D} \left[ h_{ijkl}(S_{i,n-1}^{l}) - h_{ijkl}(S_{i,n-1}^{e}) \right] L_{j,k,n-1} \\
+ M_{i,n-1} + N_{i,n-1}.
\] (B.1)
where $M_{i,n} = M_{i,n}^f + M_{i,n}^c$ and similarly for $N_{i,n}$, and
\[
L_{jk,n} = \Delta W_{j,n} \Delta W_{k,n} - \Omega_{jk} \Delta t.
\] (B.2)

If we add up (B.1) all the way back to the initial time and use $S_{0}^{f} = S_{0}^{c}$, we have
\[
S_{n}^{f} - S_{n}^{c} = \sum_{m=0}^{n-1} \left[ a_{i}(S_{m}^{f}) - a_{i}(S_{m}^{c}) \right] \Delta t + \sum_{m=0}^{n-1} \sum_{j=1}^{D} \left[ b_{ij}(S_{m}^{f}) - b_{ij}(S_{m}^{c}) \right] \Delta W_{j,m} + \sum_{m=0}^{n-1} \sum_{j,k=1}^{D} \left[ h_{ijk}(S_{m}^{f}) - h_{ijk}(S_{m}^{c}) \right] L_{jk,m} + \sum_{m=0}^{n-1} (M_{i,m} + N_{i,m}).
\] (B.3)

This is conceptually identical to the second equation in the proof of theorem 4.10 (Appendix 4) in [6], and may be treated with exactly the same methods found therein.

In particular, defining
\[
R_{n} = \mathbb{E} \left[ \max_{m \leq n} \| S_{m}^{f} - S_{m}^{c} \|^2 \right],
\] (B.4)
one can establish the recursive relation
\[
R_{n} \leq C \left( \Delta t^2 + \Delta t \sum_{m=0}^{n-1} R_{m} \right)
\] (B.5)
for some $C > 0$. A discrete version of the Grönwall inequality implies
\[
R_{n} \leq C \left( \Delta t^2 + \sum_{m=0}^{n-1} \Delta t \exp \left\{ \sum_{k=m}^{n-1} \Delta t \right\} \right) \leq C \left( \Delta t^2 + n \Delta t^3 \exp(n \Delta t) \right).
\] (B.6)

Letting $n \to N$, we have
\[
R_{N} \leq C(1 + T \exp T) \Delta t^2,
\] (B.7)
which immediately implies the desired result.

**Appendix C. Proof of Lemma 4.7.** The first step is to reestablish Lemma 4.4 in the case when the fine steps are allowed to have different step sizes. The desired result will follow by induction, in that we will treat the first $r$ time-steps as a single step, and the $(r+1)^{th}$ as the second step. Let $\delta t_1$ be the time step for the first fine step, and $\delta t_2$ the time step for the second, with $\delta W_{j,n}$ and $\delta W_{j,n+\frac{1}{2}}$ the corresponding Brownian increments with variances $\delta t_1$ and $\delta t_2$, respectively. That is,
\[
S_{n+\frac{1}{2}}^{f} = S_{n}^{f} + D^{f}(S_{n}^{f}, t_{n}, \delta t_{1}, \delta W_{j,n}),
\] (C.1)
\[
S_{n+1}^{f} = S_{n+\frac{1}{2}}^{f} + D^{f}(S_{n+\frac{1}{2}}^{f}, t_{n} + \delta t_{1}, \delta t_{2}, \delta W_{j,n+\frac{1}{2}}).
\] (C.2)
where we’ve omitted the $l$ superscripts.

Through diligent algebra, we can show that

$$S_{f,i,n}^{l+1} = S_{f,i,n}^l + D_{t}^f \left( S_{f,n}^l, t_n, \delta t_1, \delta t_2, \delta W_{j,n}^r, \delta W_{j,n+\frac{1}{2}}^r \right) - \sum_{j,k=1}^{D} h_{ijk,n} \left( \delta W_{j,n}^r \delta W_{k,n+\frac{1}{2}}^r - \delta W_{k,n}^r \delta W_{j,n+\frac{1}{2}}^r \right) + R_{i,n} + M_{i,n}^{(1)} + M_{i,n}^{(2)}$$

where

$$R_{i,n} = \left( a_{i,n+\frac{1}{2}} - a_{i,n} \right) \delta t_2$$

$$M_{i,n}^{(1)} = \sum_{j=1}^{D} \left( b_{ij,n+\frac{1}{2}} - b_{ij,n} - 2 \sum_{k=1}^{D} h_{ijk,n} \delta W_{k,n}^r \right) \delta W_{j,n+\frac{1}{2}}^r$$

$$M_{i,n}^{(2)} = \sum_{j,k=1}^{D} \left( h_{ijk,n+\frac{1}{2}} - h_{ijk,n} \right) \left( \delta W_{j,n+\frac{1}{2}}^r \delta W_{k,n+\frac{1}{2}}^r - \delta W_{k,n+\frac{1}{2}}^r \delta W_{j,n+\frac{1}{2}}^r \right).$$

From here, the argument bounding the remainder terms proceeds exactly as in Lemma 4.7 in [6]. In particular, $M_{i,n}^{(1)}$ and $M_{i,n}^{(2)}$ have vanishing expectation, and may be shown to scale like $\Delta t^{3/2}$ by Taylor expanding $b_{ij,n+\frac{1}{2}}$ and $h_{ijk,n+\frac{1}{2}}$ about $t_n$. Similarly, $a_{i,n+\frac{1}{2}}$ is Taylor expanded to separate $R_{i,n}$ into two terms, one of which satisfies the appropriate scaling for $M_{i,n}$ while the other satisfies the scaling for $N_{i,n}$. We refer the interested reader to [6] for a thorough treatment.

We now proceed by induction: Suppose that for some $r < M$, we’ve shown that

$$S_{f,i,n}^{l+r} = S_{f,i,n}^{l} + D_{t}^f \left( S_{f,n}^{l+r}, t_n, r \delta t, \sum_{q=0}^{r-1} \delta W_{j,n+\frac{1}{2}}^q \right) - \sum_{j,k=1}^{D} h_{ijk,n} \left( A_{jk,n}^{(r)} - A_{kj,n}^{(r)} \right) + M_{i,n}^f + N_{i,n}^f$$

where

$$A_{jk,n}^{(r)} = \sum_{m=1}^{r-1} \left( \delta W_{k,n+\frac{1}{2}}^m \sum_{q=0}^{m-1} \delta W_{j,n+\frac{1}{2}}^q \right).$$

and $M_{i,n}^f$ and $N_{i,n}^f$ have the scalings stated in the lemma. Note that the base case $r = 1$ is trivial, and that $r = 2$ is given by Lemma 4.4. Then, applying our modified version of Lemma 4.4 to (C.5) and

$$S_{f,i,n}^{l+r} = S_{f,i,n}^{l+r} + D_{t}^f \left( S_{f,n}^{l+r}, t_n + r \delta t, \delta t, \delta W_{j,n+\frac{1}{2}}^r \right),$$

(23)
we have

\[
S_{i,n+1}^{f,l} = S_{i,n}^{f,l} + D_i \left( S_{n}^{f,l}, t_n, r \delta t, \sum_{q=0}^{r} \delta W_{j,n+1}^{+} \right) - \sum_{j,k=1}^{D} h_{ijk} (S_n^{f,l}) \left( A_{jk,n}^{(r)} - A_{kj,n}^{(r)} \right) + M_{i,n}^{f} + N_{i,n}^{f}
\]

where the new remainder terms have been absorbed into the \( M_{i,n}^{f} \) and \( N_{i,n}^{f} \). The second and third lines can be combined to obtain

\[
S_{i,n+1}^{f,l} = S_{i,n}^{f,l} + D_i \left( S_{n}^{f,l}, t_n, r \delta t, \sum_{q=0}^{r} \delta W_{j,n+1}^{+} \right) - \sum_{j,k=1}^{D} h_{ijk} (S_n^{f,l}) \left( A_{jk,n}^{(r+1)} - A_{kj,n}^{(r+1)} \right) + M_{i,n}^{f} + N_{i,n}^{f}
\]

Letting the induction carry to \( M \), we have the desired result, for \( A_{jk,n}^{(M)} = A_{jk,n} \) by definition.

REFERENCES

[1] A.N. Burkitt. A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input. Biological cybernetics, 95(1):1–19, 2006.
[2] A.M. Dimits, B.I. Cohen, R. Callisch, M.S. Rosin, and L.F. Rickenton. Higher-order time integration of coulomb collisions in a plasma using Langevin equations. Journal of Computational Physics, 2013.
[3] J.G. Gaines and T.J. Lyons. Random generation of stochastic area integrals. SIAM Journal on Applied Mathematics, 54(4):1132–1146, 1994.
[4] M.B. Giles. Improved multilevel Monte Carlo convergence using the Milstein scheme. Monte Carlo and quasi-Monte Carlo methods 2006, pages 343–358, 2008.
[5] M.B. Giles. Multilevel Monte Carlo path simulation. Operations Research, 56(3):607–617, 2008.
[6] M.B. Giles and L. Szpruch. Antithetic multilevel Monte Carlo estimation for multi-dimensional SDEs without Lévy area simulation. arXiv preprint arXiv:1202.6283, 2012.
[7] Michael B Giles and Lukasz Szpruch. Antithetic multilevel monte carlo estimation for multidimensional sdes. Monte Carlo and Quasi-Monte Carlo Methods 2012 (submitted), 2012.
[8] Daniel T. Gillespie. The chemical Langevin equation. The Journal of Chemical Physics, 113:297, 2000.
[9] F. Haghighat, P. Fazio, and T.E. Unny. A predictive stochastic model for indoor air quality. Building and Environment, 23(3):195–201, 1988.
[10] C.J. Harris. Modelling, simulation and control of stochastic systems with applications in wastewater treatment. International Journal of Systems Science, 8(4):393–411, 1977.
[11] Steven L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. Review of financial studies, 6(2):327–343, 1993.
[12] G. Kallianpur and Robert L. Wolpert. Weak convergence of stochastic neuronal models. In Stochastic methods in biology, pages 116–145. Springer, 1987.
[13] Peter E Kloeden and Eckhard Platen. Numerical solution of stochastic differential equations, volume 23. Springer, 2011.
[14] Peter E Kloeden, Eckhard Platen, and IW Wright. The approximation of multiple stochastic integrals. Stochastic analysis and applications, 10(4):431–441, 1992.
[15] Peter R. Kramer, Charles S. Peskin, and Paul J. Atzberger. On the foundations of the stochastic immersed boundary method. Computer Methods in Applied Mechanics and Engineering, 197(25):2232–2249, 2008.
[16] D.S. Lemons, D. Winske, W. Daughton, and B. Albright. Small-angle Coulomb collision model for particle-in-cell simulations. *Journal of Computational Physics*, 228(5):1391–1403, 2009.

[17] Francesco Mezzadri. How to generate random matrices from the classical compact groups. *Notices Amer. Math. Soc.*, 54(5):592–604, 2007.

[18] G.N. Mil'shtein. A method of second-order accuracy integration of stochastic differential equations. *Theory of Probability & Its Applications*, 23(2):396–401, 1979.

[19] Grigori N. Milstein and Michael V. Tretyakov. *Stochastic numerics for mathematical physics*. Springer-Verlag, Berlin, 2004.

[20] Steven E. Shreve. *Stochastic Calculus for Finance II, Continuous Time Models*. Springer, 2004.

[21] Magnus Wiktorsson. Joint characteristic function and simultaneous simulation of iterated Itô integrals for multiple independent brownian motions. *Annals of Applied Probability*, pages 470–487, 2001.