Monte Carlo Computation in Finance

Jeremy Staum

Abstract This advanced tutorial aims at an exposition of problems in finance that are worthy of study by the Monte Carlo research community. It describes problems in valuing and hedging securities, risk management, portfolio optimization, and model calibration. It surveys some areas of active research in efficient procedures for simulation in finance and addresses the impact of the business context on the opportunities for efficiency. There is an emphasis on the many challenging problems in which it is necessary to perform several similar simulations.

1 Introduction

This tutorial describes some problems in finance that are of interest to the Monte Carlo research community and surveys some recent progress in financial applications of Monte Carlo. It assumes some familiarity with Monte Carlo and its application to finance: for an introduction, see [24, 46]. For quasi-Monte Carlo methods in finance, see [46, 72]. Section 2 provides an overview of financial simulation problems and establishes notation. Section 3 describes aspects of the business context for simulation in the financial industry and the implications for researchers. The principal theme of this tutorial is the need to solve multiple similar simulation problems and the associated opportunity to design efficient Monte Carlo procedures. The mathematical settings in which multiple similar problems arise, and the tools researchers use to deal with them, occupy Section 4. Section 5, on variance reduction, surveys database Monte Carlo and adaptive Monte Carlo. Section 6 is devoted to simulation for risk management. American options and portfolio optimization are covered in Section 7. Section 8 surveys sensitivity analysis by Monte Carlo. Some

Department of Industrial Engineering and Management Sciences, Robert R. McCormick School of Engineering and Applied Science, Northwestern University, 2145 Sheridan Road, Evanston, Illinois 60208-3119, USA, http://users.iems.northwestern.edu/~staum/

recent progress in simulating solutions of stochastic differential equations appears in Section 9.

2 Overview of Financial Simulation Problems

Financial simulation models involve a vector stochastic process $\mathbf{S}$ of *underlying* financial variables. Let $\mathbf{S}(t)$ be the value of $\mathbf{S}$ at time t and S_j be the jth component of $\mathbf{S}$. The model is expressed as a probability measure P governing $\mathbf{S}$. A characteristic feature of finance is that valuation calls for using another measure Q, derived from P and a choice of *numéraire*, or unit of account. For example, in the Black-Scholes model, one may take the numéraire to be S_0, a money market account earning interest at a constant rate r. Its value $S_0(t) = e^{rt}$. In this model, under the *real-world measure P*, the stock price S_1 is geometric Brownian motion with drift μ and volatility σ. Using the money market account as numéraire leads to the *risk-neutral measure Q*, under which S_1 is geometric Brownian motion with drift r and volatility σ. The real-world expected price of the stock at a future time T is $\mathrm{E}_P[S_1(T)] = S_1(0)e^{\mu T}$. The stock's value now, at time 0, is $S_0(0)\mathrm{E}_Q[S_1(T)/S_0(T)] = S_1(0)$. In general, $S_0(0)\mathrm{E}_Q[H/S_0(T)]$ is a value for a security whose payoff is H at time T. Thus, we use P to simulate the real world, but we simulate under Q to value a security.

Figure 1 shows how P and Q enter into the four interrelated problems of valuing and hedging securities, risk management, portfolio optimization, and model calibration. The model specifies security values as expectations under Q. Sensitivities of these expectations, to the underlying and to the model's parameters, are used in hedging to reduce the risk of loss due to changes in those quantities. In addition to these sensitivities, real-world probabilities of losses are important in risk management. Simulating scenarios under P is one step in sampling from the distribution of profit and loss (P&L). A portfolio's P&L in each scenario involves securities' values in that scenario, and they are conditional expectations under Q. The same structure can arise in portfolio optimization, where the goal is to choose the portfolio strategy that delivers the best P&L distribution. Calibration is a way of choosing a model's parameters. It is very difficult to estimate the parameters of P statistically from the history of the underlying. Instead, one may choose the parameters of Q so that the prices of certain securities observed in the market closely match the values that the model assigns to them.

Before elaborating on these four problems, we establish some notation. We discretize a time interval $[0,T]$ into m steps, considering the times $0 = t_0, t_1, \ldots, t_m = T$, and let $\mathscr{F}_i$ represent the information available at step i after observing $\mathbf{S}(t_0)$, $\mathbf{S}(t_1)$, $\ldots$, $\mathbf{S}(t_i)$. In applying Monte Carlo, we aim to estimate an expectation or integral $\mu = \mathrm{E}[Y] = \int f(\mathbf{u})\,\mathrm{d}\mathbf{u}$. The domain of integration is often omitted; it is understood to be $[0,1)^d$ when the variable of integration is $\mathbf{u}$. We often ignore the details of how to simulate the random variable $Y = f(\mathbf{U})$, where $\mathbf{U}$ is uniformly distributed on $[0,1)^d$. Such details remain hidden in the background: when we generate a point set $\mathbf{u}_1, \ldots, \mathbf{u}_n$ in order to estimate μ by $\sum_{i=1}^{n} f(\mathbf{u}_i)/n$, each vector $\mathbf{u}_i$ results in a simu-

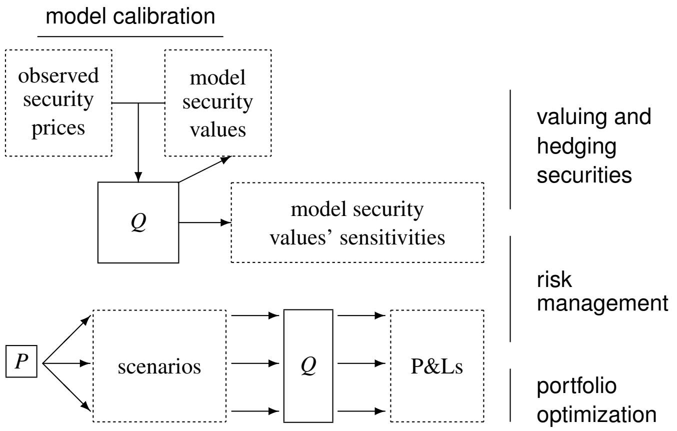

Fig. 1 Ecology of computations in finance

lated *path* $\mathbf{S}^{(i)}(t_1),\ldots,\mathbf{S}^{(i)}(t_m)$, where the superscript (i) indicates that this path is generated by the ith point or *replication*. The mapping ϕ from point to path is such that when $\mathbf{U}$ is uniformly distributed on $[0,1)^d$, $\phi(\mathbf{U})$ has the finite-dimensional distribution specified by P or Q, as appropriate. Sometimes we explicitly consider the intermediate step of generating a random vector $\mathbf{X}$ before computing the random variable $Y=\tilde{f}(\mathbf{X})$. We will often consider the influence of a parameter vector θ, containing initial values of the underlying, parameters of the model, characteristics of a security, or decision variables. In full generality,

$$\mu(\theta)=\mathrm{E}[Y(\theta)]=\int_{[0,1)^d} f(\mathbf{u};\theta)\,\mathrm{d}\mathbf{u}=\int \tilde{f}(\mathbf{x};\theta)g(\mathbf{x};\theta)\,\mathrm{d}\mathbf{x}=\mathrm{E}_\theta[\tilde{f}(\mathbf{X};\theta)].$$

Derivative Securities. *Derivative securities* have payoffs that are functions of the underlying. In many models, the market is *complete*, meaning that the derivative security's payoff can be replicated by trading in the underlying securities. Then, in the absence of arbitrage, the derivative security's value should equal the cost of setting up that replicating strategy, and this is an expectation under Q [46, §1.2]. For a survey of ideas about how to price a derivative security whose payoff can not be replicated, see [92]. According to some of these ideas, price bounds are found by optimizing over hedging strategies or probability measures. Computational methods for these price bounds have received little attention; exceptions are [75, 84].

The *Greeks*, sensitivities of derivative security values to the underlying or to model parameters, are used to measure and to hedge the risk of portfolios. For example, where $\Delta_j=\mu'(S(0))$ is the sensitivity of the jth security's value to small changes in the underlying asset's price, the sensitivity of a portfolio containing w_j shares of each security j is $\Delta=\sum_j w_j\Delta_j$. Selling Δ shares of the underlying asset makes the portfolio value insensitive to small changes in the underlying asset's price. It is portfolios, rather than individual securities, that are hedged. However, it can be helpful to know the Greeks of each security, which are its contribution to the portfolio's Greeks.

The Monte Carlo literature on finance has given a disproportionately great amount of attention to efficient methods for valuing and hedging some particular kind of exotic option in isolation. At this point, it is worth shifting attention to the other three problems or to addressing issues that arise in valuing and hedging derivative securities because of the business context. Also, research on simulating recently developed models can contribute to the solution of all four problems. For example, simulating models with jumps is an important topic of research at present. The following derivative securities are of particular interest:

- Asian options are important in commodities and foreign exchange, because they can help non-financial firms hedge risks arising from their businesses.
- Mortgage-backed securities [32] are in the news.
- So are credit derivatives, from single-name credit default swaps to portfolio credit derivatives such as collateralized debt obligations [13, 40, 41].

All of these lead to a high dimension d for integration, because they involve a large number m of time steps, and can pose challenges for Monte Carlo and quasi-Monte Carlo methods.

Risk Management. As illustrated by Figure 1, risk management is a broad subject that overlaps with the topics of hedging individual securities and of portfolio optimization. Hedging a portfolio's Greeks is one approach in risk management. Another is minimizing a risk measure of the hedged portfolio's P&L [26]. A *risk measure* is a real-valued functional of P&L or the distribution of P&L, such as variance, value at risk (VaR), or conditional value at risk (CVaR). For example, because of a regulatory requirement to report VaR, financial firms compute the 99th percentile of the loss distribution. Because limits on risk constrain activities, and because regulators impose a costly capital requirement on a financial firm proportional to its risk measure, there is also interest in decomposing the risk measure into a sum of *risk contributions* from the firm's positions or activities. Risk contributions as often computed as sensitivities of the risk measure to portfolio positions or the scale of a trading desk's activities. See [80] for an overview of risk management and [40, 48] for credit risk modeling.

Portfolio Optimization. Portfolio optimization features a decision variable that specifies a vector θ of portfolio weights. This may be a static vector or it may be a stochastic process of portfolio weights that would be chosen in every possible scenario at each time step. The objective is often to maximize the expected utility $\mathrm{E}[u(W(T))]$ of future wealth $W(T) = \theta(T)^\top \mathbf{S}(T)$, or to maximize the expected total discounted utility $\mathrm{E}[\sum_{i=0}^{m} e^{-\beta t_i} u(C(t_i))]$ of a consumption process C, which is another decision variable. The investor's initial wealth $W(0)$ imposes the budget constraint $\theta^\top \mathbf{S}(0) = W(0)$. A multi-period formulation requires self-financing constraints like $\theta(t_i)^\top \mathbf{S}(t_i) = \theta(t_{i-1})^\top \mathbf{S}(t_i) - C(t_i)$, which may be more complicated if there are features such as transaction costs and taxes. There may also be constraints such as a prohibition against short-selling, $\theta \geq 0$, or an upper bound on a risk measure of $W(T)$. For background on portfolio optimization, see [14, 28, 33].

Model Calibration. Calibrating the model to observed prices of derivative securities is an inverse problem, usually ill-posed. As shown in the upper left corner of Figure 1, the model maps a parameter vector θ to a vector of security values $\mu(\theta)$, and here the task is to find the θ that yields a given vector $\mathbf{p}$ of the securities' market prices. The difficulty is that the mapping $\mu(\cdot)$ may be non-invertible or the given $\mathbf{p}$ may not be in its range. A standard approach is to put a norm on the space of price vectors and to use $\theta^* = \arg\min_\theta \|\mu(\theta) - \mathbf{p}\|$. If the model has many parameters, it may be necessary to add a penalty term to the objective to prevent over-fitting. For an exposition, see [27, Ch. 13]. A recent innovation employing Monte Carlo methods in the search for good parameters is [10].

3 Financial Simulations in Context

Much research on Monte Carlo in finance focuses on computational efficiency: reducing time required to attain a target precision, or attaining better precision given a fixed computational budget. Efficiency is important: some financial simulations impose such a heavy computational burden that banks have invested in parallel computing platforms with thousands of processors. However, the value of efficiency techniques depends on the context of the business process within which computation takes place. Because computers are cheap and financial engineers are expensive, the benefit of a more efficient simulation must be weighed against the cost of analyst time required to implement it. Efficiency techniques are more valuable the easier they are to implement and the more broadly applicable they are. Efficiency is most important for computationally intensive problems, such as those in Section 4. The software engineering environment may hinder the implementation of efficient simulation procedures. Many firms use modular simulation software engines in which path generation does not depend on the security being considered. They may even generate a fixed set of paths of the underlying, which are then reused for several purposes, such as pricing many derivative securities. This is an obstacle to implementing some efficiency techniques: for example, it prevents the use of importance sampling methods tailored to each derivative security.

The Value of Speed. Faster answers are always welcome, but speed is more valuable in some applications than others. It does not matter whether it takes 0.1 or 0.01 seconds to deliver a precise estimate of one option price to a trader. However, it does matter whether it takes 60 hours or 6 hours to measure firm-wide risk over a one-day horizon: after 60 hours, the answer is useless because that day is over. Faster calibration is beneficial in delivering more frequently updated model parameters.

The Value of Precision. Faster is always better, but precision can be excessive. The reason is that *precision*, related to the reported uncertainty in an estimator, is not the same as *accuracy*, related to how far the estimator is from the truth. In Monte Carlo, precision relates to the statistical properties of an estimator of a quantity that is specified by a model; if the estimator is consistent, it is possible to attain arbitrarily high precision by increasing the computational budget. Accuracy also involves *model error*, the difference between some quantity as specified by the model and the value it really has. Only building a better model, not more computational effort, will reduce model error. It is unhelpful to provide Monte Carlo estimates whose precision greatly exceeds the model's accuracy. Of course, this is true in any scientific computing endeavor, but model error tends to be greater in operations research and finance than in other disciplines such as physics. Therefore the useful degree of precision in financial simulations is less than in some other scientific computations.

In finance, the possibility of loss due to *model error* is known as *model risk*, and it is quite large: we can not be certain that an option's expected payoff is \$10.05 and not \$10.06, nor that value at risk is \$10 million as opposed to \$11 million. Simulation output can be too precise relative to model error. Suppose we run a long simulation and report a 99% confidence interval of $[10.33, 10.34]$ million dollars for value at risk. What this really means is that the Monte Carlo simulation left us with 99% confidence that our model says that value at risk is between \$10.33 and \$10.34 million. However, because of model error, we do not have high confidence that value at risk actually falls in this interval. Reporting excessive precision is a waste of time, and it is also dangerous in possibly misleading decision-makers into thinking that the numbers reported are very accurate, forgetting about model risk.

The utility of precision is also limited by the way in which answers are used. For example, when Monte Carlo is used in pricing derivative securities, the *bid-ask spread* provides a relevant standard: if market-makers charge ("ask") a dollar more when they sell an option than they pay ("bid") when they buy it, they do not need to price the option to the nearest hundredth of a cent.

As a rough general guideline, I suggest that 0.1% relative error for derivative security prices and 1% relative error for risk measures would not be too precise in most applications. Here *relative error* means the ratio of root mean squared error to some quantity. Usually it makes sense to take this quantity to be the price or risk measure being estimated. However, in some applications, the price is zero or nearly zero and it makes sense to take something else as the denominator of relative error. For example, in pricing swaps, one may use the swap rate or the notional principal on which the swap is written (in which case greater precision could be appropriate).

Repeating Similar Simulations. In finance, there are opportunities to improve efficiency because we often perform multiple simulations that are structurally the same and only differ slightly in the values of some parameters. Examples of three kinds of situations in which repeated similar simulations arise are:

- *Fixed set of tasks:* In electronic trading and market-making, we want to value many options which differ only in their strike prices and maturities. The strikes and maturities are known in advance.

- *Multi-step tasks:* Calibration can involve repeated simulations with different model parameters that are not known in advance, but depend on the results of previous steps.
- *Sequential tasks:* We measure a portfolio's risk every day. Tomorrow's portfolio composition and model parameters are currently unknown, but will probably differ only slightly from today's.

Section 4 describes some problems in which multiple simulations arise and methods for handling them efficiently. The variance reduction methods of Section 5 also help in this context. The aim of Database Monte Carlo is to use information generated in one simulation to reduce the variance of similar simulations. Adaptive Monte Carlo and related approaches can be applied to choose good variance reduction parameters to use in one simulation based on the output of a similar simulation.

Thinking about repeated simulations may lead to a paradigm shift in our understanding of how Monte Carlo should support computation in finance. The dominant paradigm is to treat each problem that arises as a surprise, to be dealt with by launching a new simulation and waiting until it delivers a sufficiently precise answer. Instead we might think of a business process as creating an imperative for us to invest computational resources in being able to estimate $\mu(\theta)$ for a range of θ.

4 Multiple Simulation Problems

Many of the computationally intensive problems most worthy of researchers' attention involve multiple simulations. In many cases, these are structurally similar simulations run with different parameters.

The Portfolio Context A large portfolio, containing a large number ℓ of securities, can make risk management and portfolio optimization simulations computationally expensive. The approach to portfolio valuation is often to choose the number m_i of replications in a simulation to value to the ith security large enough to value this security precisely, with the result that the total number of replications $\sum_{i=1}^{\ell} m_i$ is very large. However, [54] point out that if ℓ is large, the portfolio's value can be estimated precisely even if m is very small, as long as each security's value is estimated with independent replications: then the variance in estimating each security's value is large, but the variance in estimating the portfolio value is small.

Nested Simulations Nested simulation arises when, during a simulation, we would like to know a conditional expectation. If it is not known in closed form, we may resort to an *inner-level* simulation to estimate it. That is, within an *outer-level* simulation in which we want to estimate $\int f(\mathbf{u})\,\mathrm{d}\mathbf{u}$ by $\sum_{i=1}^{n} f(\mathbf{u}_i)/n$ but can not evaluate the function f, we may nest an inner level of simulation, in which we estimate $f(\mathbf{u}_1),\ldots,f(\mathbf{u}_n)$. See [69] for a general framework for two-level simulation in which we wish to estimate a functional of the distribution of $f(\mathbf{U})$ and estimate f by Monte Carlo. For examples of nested simulation, see Section 6 on risk management, where inner-level simulation estimates a portfolio's value in each scenario simulated at the outer level, and Section 7 on American option pricing, where inner-level simulation estimates the option's continuation value at every potential exercise date on each path simulated at the outer level. For the sake of computational efficiency, it is desirable to avoid a full-blown nested simulation, which tends to require a very large total number of replications: mn if each of n outer-level scenarios or paths receives m inner-level replications. One way of avoiding nested simulation is metamodeling.

Metamodeling *Metamodeling* of a simulation model is the practice of building an approximation $\hat{\mu}$ to a function μ, using a simulation model that enables estimation of $\mu(\theta)$ for any θ in a domain Θ. One purpose of metamodeling is to be able to compute an approximation $\hat{\mu}(\theta)$ to $\mu(\theta)$ quickly. Many simulation models are slow to evaluate $\mu(\theta)$, but metamodels are constructed so that they are fast to evaluate. This makes them useful in dealing with repeated similar simulations (§3). It can be faster to build a metamodel and evaluate it repeatedly than to run many separate simulations, so metamodeling can reduce the computational burden of a fixed set of tasks or a multi-step task. In dealing with sequential tasks, metamodeling enables an investment of computational effort ahead of time to provide a rapid answer once the next task is revealed. Another benefit of metamodeling is that it supports visualization of the function's behavior over the whole domain Θ, which is more informative than merely estimating local sensitivity.

Metamodeling is better developed for deterministic simulations than for stochastic simulations, but it is becoming more widespread in stochastic simulation: for references, see [3]. In deterministic simulation, the metamodel is built by running the simulation model at some *design points* $\theta_1, \ldots, \theta_k$ and using the observed outputs $\mu(\theta_1), \ldots, \mu(\theta_k)$ to construct $\hat{\mu}$ by regression, interpolation, or both. In stochastic simulation, this is not exactly possible, because $\mu(\theta_i)$ can only be estimated; we explain below how to deal with this conceptual difficulty. The two main approaches to metamodeling are regression and kriging. Regression methods impose on the metamodel $\hat{\mu}$ a particular form, such as quadratic, composed of splines, etc. Then the unknown coefficients are chosen to minimize the distance between the vectors $(\mu(\theta_1), \ldots, \mu(\theta_k))$ and $(\hat{\mu}(\theta_1), \ldots, \hat{\mu}(\theta_k))$. Finance is one of the applications in which it may be hard to find a form for $\hat{\mu}$ that enables it to approximate μ well over a large domain Θ. However, in some applications such as sensitivity estimation and optimization, it may only be necessary to approximate μ well locally. Unlike regression, *kriging* is an interpolation method that forces the metamodel to agree with the simulation outputs observed at all design points. However, it can be combined with estimation of a trend in $\mu(\theta)$ as a function of θ, as in regression. There are two principal difficulties for metamodeling of financial simulations.

One is that metamodeling is hard when θ is high-dimensional and when μ is discontinuous or non-differentiable. One remedy for the latter problem is to construct separate metamodels in different regions, such that μ is differentiable on each region. In some cases, the troublesome points are known a priori. For example, in a typical option pricing example, the option price μ is non-differentiable where the stock price equals the strike price and time to maturity is zero. In other cases, it is not known in advance whether or where μ may be badly behaved. It may help to apply methods that automatically detect the boundaries between regions of Θ in which there should be separate metamodels [55].

The second obstacle is common to all stochastic simulations. It involves the conceptual difficulty that we can not observe the true value $\mu(\theta)$ at any input θ, and a related practical shortcoming. We might deal with the conceptual difficulty in one of two ways. One way is to use quasi-Monte Carlo or fix the seed of a pseudo-random number generator and regard the output v of the stochastic simulation as deterministic, given that these *common random numbers* (CRN) are used to simulate at any input θ. We can build a metamodel $\hat{v}$ of v, but its output $\hat{v}(\theta)$ is an approximation to $v(\theta)$, the output that the simulation would produce if it were run at θ with CRN, not necessarily a good approximation to the expectation $\mu(\theta)$ that we want to know. Then the practical shortcoming is that $v(\theta)$ needs to be a precise estimate of the expectation $\mu(\theta)$, so the number of replications used at each design point must be large. The second way to deal with the conceptual difficulty is to use different pseudo-random numbers at each design point θ_i, but build a metamodel by plugging in a precise simulation estimate for $\mu(\theta_i)$ anyway. This entails the same practical shortcoming, and the Monte Carlo sampling variability makes it harder to fit a good metamodel. It is a practical shortcoming to need many replications at each design point because, given a fixed computational budget, it might be more efficient to have more design points with fewer replications at each. *Stochastic kriging* [3] is one solution to these problems. It shows how uncertainty about the expectation $\mu(\theta)$ arises from the combination of interpolation and the Monte Carlo sampling variability that affects the stochastic simulation output as an estimate of $\mu(\theta_i)$ for each design point. Stochastic kriging makes it possible to get a good approximation to $\mu(\theta)$ even when the number of replications at each design point is small and provides a framework for analyzing the trade-off between having many design points and having more replications at each of them.

Metamodeling is closely related in its aims to database Monte Carlo (§5).

Optimization Another reason that one might need to obtain Monte Carlo estimates of $\mu(\theta)$ for multiple values of θ is when optimizing over θ, if simulation is needed in evaluating the objective or constraints of the optimization problem. This is *optimization via simulation* (OvS): for an overview, see [34, 60]. In the following, we concentrate on the problem $\min_{\theta\in\Theta}\mu(\theta)$ of minimizing an expectation that must be estimated by Monte Carlo over a continuous decision space Θ defined by constraints that can be evaluated without Monte Carlo. The typical pattern is that an optimization procedure visits candidate solutions $\theta_0, \theta_1, \ldots, \theta_K$ sequentially, at each step j using information generated by Monte Carlo to choose θ_j. It is quite useful in choosing θ_j to be able to estimate the gradient $\nabla\mu(\theta_{j-1})$: see Section 8.

- **Sample Average Approximation.** The simplest approach is to approximate the objective value $\mu(\theta)$ by the sample average $\hat{\mu}(\theta) = \sum_{i=1}^{n} f(\mathbf{u}_i;\theta)/n$. That is, the common random numbers $\mathbf{u}_1, \ldots, \mathbf{u}_n$ are used to estimate the objective at any candidate solution θ_j. To minimize $\hat{\mu}$, one can use a gradient-free optimization procedure or use the gradient $\nabla\hat{\mu}(\theta) = \sum_{i=1}^{n} \nabla_\theta f(\mathbf{u}_i;\theta)/n$ if available.

- **Metamodeling and Stochastic Approximation.** Another approach involves running more simulations at each step, using increasing total simulation effort as the optimization procedure converges to the optimal θ. *Sequential metamodeling* [7] considers a neighborhood Θ_j of θ_{j-1} at step j, and builds a metamodel $\hat{\mu}_j$ that approximates μ locally, on Θ_j. The gradient $\nabla\hat{\mu}_j$ helps in choosing θ_j. (Because of the difficulty of building metamodels that fit well globally, it has not been common practice in OvS simply to build one metamodel and minimize over it.) *Stochastic approximation* depends on ways of computing an estimate $\widehat{\nabla}\mu(\theta)$ of the gradient that are described in Section 8 and [35]. At step j, the next candidate solution is $\theta_j = \theta_{j-1} - \gamma_j \widehat{\nabla}\mu(\theta_{j-1})$. It can be troublesome to find a sequence of step sizes $\{\gamma_j\}_{j\in\mathbb{N}}$ that works well for one's particular optimization problem [34]. For recent progress, see [18, 82]. Other questions include whether it is best to estimate the optimal θ by θ_n or a weighted average of $\theta_1,\ldots,\theta_n$, or to constrain θ_j from moving too far from θ_{j-1}; see [60].
- **Metaheuristics.** Various *metaheuristic* methods, such as simulated annealing and genetic algorithms, use Monte Carlo to solve optimization problems heuristically, even if the objective μ can be evaluated without Monte Carlo: they randomly select the next candidate solution θ_j. See [83] for an overview in the simulation context, where it is typical to employ the metaheuristic simply by using a simulation estimate $\hat{\mu}(\theta_j)$ in place of each $\mu(\theta_j)$. Metaheuristics can solve difficult optimization problems, such as model calibration problems that are non-convex, with multiple local minima and regions in which the objective is very flat. However, they are called *meta*heuristics because they require tailoring to the specific problem to produce an algorithm that works well. Randomness over candidate solutions can have more benefits than escaping from local minima: for example, [10] uses a metaheuristic optimization procedure to account for the parameter uncertainty that remains after model calibration.
- **Approximate Dynamic Programming.** This discussion of optimization has not yet explicitly taken into account optimization over policies that include decisions at multiple times, which is important for American options and dynamic portfolio optimization. This is the subject of dynamic programming, in which the optimal decision at each time maximizes a value function, such as the expected utility of terminal wealth as a function of underlying prices and the composition of the portfolio. Approximate dynamic programming (ADP) is a solution method for dynamic programs that are too large to solve exactly. Instead of computing the exact value of each state, ADP constructs an approximate value function. Monte Carlo can help in approximating the value function: then ADP is closely related to simulation metamodeling. For more on ADP, see [11, 88, 89].

5 Variance Reduction

Here we discuss only two active areas of research in variance reduction that have important applications in finance.

Database Monte Carlo. The idea of database Monte Carlo (DBMC) is to invest computational effort in constructing a database which enables efficient estimation for a range of similar problems [16]. In finance, it is often important to solve a whole set of similar problems (§3). The problems are indexed by a parameter θ, and we want to estimate $\mu(\theta) = \int f(\mathbf{u};\theta)\,\mathrm{d}\mathbf{u}$ for several values of θ, for example, to price several options of different strike prices. DBMC involves choosing a base value θ_0 of the parameter and evaluating $f(\cdot;\theta_0)$ at many points $\omega_1,\ldots,\omega_N$ in $[0,1)^d$. The set $\{(\omega_i, f(\omega_i;\theta_0))\}_{i=1,\ldots,N}$ constitutes the database. DBMC provides a generic strategy for employing a variance reduction technique effectively: the purpose of investing computational effort in the database is that it enables powerful variance reduction in estimating $\mu(\theta)$ for values of θ such that $f(\cdot;\theta)$ is similar to $f(\cdot;\theta_0)$. It may be possible to estimate $\mu(\theta)$ well with $f(\cdot,\theta)$ evaluated at only a small number n of points. DBMC has been implemented with stratification and control variates [16, 96, 97, 98]. All but one of the methods in these papers are structured database Monte Carlo (SDMC) methods, in which further effort is expended in structuring the database: the database is sorted so that $f(\omega_i;\theta_0)$ is monotone in i [98].

SDMC with stratification partitions $\{1,\ldots,N\}$ into $n \ll N$ strata $I_1 = \{1,\ldots,i_1\}$, $I_2 = \{i_1+1,\ldots,i_2\}$, …, $I_n = \{i_{n-1}+1,\ldots,N\}$. (How best to partition is a subject of active research.) It then performs stratified resampling of $\mathbf{u}_1,\ldots,\mathbf{u}_n$ from $\{\omega_1,\ldots,\omega_N\}$. That is, $\mathbf{u}_1$ is drawn uniformly from the set $\{\omega_i : i \in I_1\}$, $\mathbf{u}_2$ uniformly from $\{\omega_i : i \in I_2\}$, etc. SDMC then estimates $\mu(\theta)$ by $\sum_{j=1}^{n} p_j f(\mathbf{u}_j;\theta)$ where $p_j = |I_j|/N = (i_j - i_{j-1})/N$. If this stratification provides good variance reduction, then $\sum_{j=1}^{n} p_j f(\mathbf{u}_j;\theta)$ is a good estimator of $\sum_{i=1}^{N} f(\omega_i;\theta)/N$. In turn, $\sum_{i=1}^{N} f(\omega_i;\theta)/N$ is a good estimator of $\mu(\theta)$ because N is large. Then, even though n is small, $\sum_{j=1}^{n} p_j f(\mathbf{u}_j;\theta)$ is a good estimator of $\mu(\theta)$.

The advantage of SDMC can be understood by viewing it as a scheme for automatically creating good strata. Ordinary stratification requires partitioning $[0,1)^d$ into strata, and it is time-consuming and difficult to find a good partition, especially because the partition must be such that we know the probability of each stratum and how to sample uniformly within each stratum. Although SDMC actually stratifies the database, it is similar to partitioning $[0,1)^d$ into strata $\mathscr{X}_1,\ldots,\mathscr{X}_n$ such that $\{\omega_i : i \in I_j\} \subseteq \mathscr{X}_j$ for all $j = 1,\ldots,n$. Typically, this partition is better than one that an analyst could easily create, because SDMC takes advantage of knowledge about $f(\cdot;\theta_0)$ that is encoded in the database. If $f(\omega_i,\theta)$ is close to monotone in the database index i, then SDMC with stratification provides excellent variance reduction [97]. SDMC avoids issues that make it hard for analysts to find good partitions. We need not know the stratum probabilities, because they are estimated by sample proportions from the database. Nor do we need to know how to sample from the conditional distribution of $f(\mathbf{U};\theta)$ given that it falls in a certain stratum, because stratified sampling is performed using the database indices.

DBMC applied to control variates [16] leads to the idea of a quasi-control variate [31], i.e., a random variable used as a control variate even though its mean is unknown and merely estimated by Monte Carlo [85]. In DBMC, one can use $f(\mathbf{u};\theta_0)$ as a quasi-control variate, with estimated mean $\sum_{i=1}^{N} f(\omega_i;\theta_0)/N$. One may resample $\mathbf{u}_1,\ldots,\mathbf{u}_n$ from $\{\omega_1,\ldots,\omega_N\}$ or instead use fresh points $\mathbf{u}_1,\ldots,\mathbf{u}_n$, and then estimate $\mu(\theta)$ by $\sum_{j=1}^{n} f(\mathbf{u}_j;\theta)/n - \beta(\sum_{j=1}^{n} f(\mathbf{u}_j;\theta_0)/n - \sum_{i=1}^{N} f(\omega_i;\theta_0)/N)$. There are also SDMC methods which involve sorting the database and using the database index as a control variate [96].

DBMC is a powerful and exciting new strategy for variance reduction when handling multiple similar problems. DBMC methods are generic and provide automated variance reduction, requiring relatively little analyst effort. Open questions remain, especially in experiment design for DBMC. What is the optimal database size N when one must estimate $\mu(\theta_1),\ldots,\mu(\theta_k)$ given a fixed budget $C = N + kn$ of function evaluations? We may be interested in some values of θ that are near the base value θ_0 and others that are far: when is it worthwhile to restructure the database or create a new database at another base value?

Such questions emphasize differences between DBMC, in its present state of development, and metamodeling. DBMC and metamodeling are two ways of using an investment of computational effort to get fast estimates of $\mu(\theta)$ for many values of θ. However, they work quite differently. Metamodeling provides an estimate of $\mu(\theta)$ without any further simulation, but the estimate is biased, in general; when metamodeling works badly, large errors can result. Metamodeling works by exploiting properties of the function μ, whereas DBMC works by exploiting properties of f. DBMC estimates $\mu(\theta)$ with a small simulation of n replications, and the resulting estimate is unbiased (ignoring bias due to estimating coefficients of control variates). The parallel to metamodeling suggests extending DBMC to incorporate information from multiple simulations, not just one at θ_0.

Adaptive Monte Carlo. The fundamental idea of adaptive Monte Carlo is to improve the deployment of a variance reduction technique during the simulation, using information generated during the simulation. That is, the variance reduction technique is parameterized by ϑ, where the special notation ϑ indicates that parameter does not affect the mean $\mu = \mathrm{E}[f(\mathbf{U};\vartheta)]$. However, it does affect the variance $\mathrm{Var}[f(\mathbf{U};\vartheta)]$. Adaptive Monte Carlo uses simulation output to choose ϑ to reduce the variance $\mathrm{Var}[f(\mathbf{U};\vartheta)]$. A number of Monte Carlo methods can be viewed as adaptive to some extent, even the long-standing practice of using regression to choose the coefficients of control variates based on simulation output.

This standard way of implementing control variates illustrates a recurrent question in adaptive Monte Carlo: should one include the replications used to choose ϑ in the estimator of μ, or should one throw them out and include only fresh replications in the estimator? If separate batches of replications are used to choose the coefficients and to estimate the expectation, the estimator with control variates is unbiased. However, it is preferable to use the same replications for both tasks, despite the resulting bias, which goes to zero as the sample size goes to infinity [46, §4.1.3]. Many adaptive Monte Carlo methods include all the replications in the estimator, which is nonetheless asymptotically unbiased under suitable conditions. In some portfolio risk measurement and American option pricing problems, the bias may be large at the desired sample size. There are methods for these problems, discussed in Sections 6 and 7, that use a fresh batch of replications to reduce bias or to deliver probabilistic bounds for bias.

There are two main approaches to adaptive Monte Carlo. In one approach, the analyst chooses a parameterized variance reduction scheme, and adaptive Monte Carlo tries to choose ϑ to attain variance near $\inf_{\vartheta} \mathrm{Var}[f(\mathbf{U};\vartheta)]$. The other approach is oriented towards learning a value function which, if known, would enable zero-variance simulation. This kind of adaptive Monte Carlo achieves variance reduction by making use of an approximation to the value function. In finance, both approaches employ optimization via simulation (§4), either to minimize variance or to find the approximate value function that best fits the simulation output. Stochastic approximation (SA) and sample average approximation (SAA) have been employed as optimization methods. Importance sampling and control variates are the most common variance reduction methods in this literature.

In the variance-minimization approach, [21] uses SAA while [4, 93] use SA. The procedures using SA can have multiple stages: at stage n, variance reduction is performed using the parameter ϑ_{n-1}, and then the parameter is updated to ϑ_n based on the new simulation output. The estimator is computed by [93] as an average of fresh replications in the last stage, which were never used to choose the variance reduction parameter; it is an average of all replications in [4] and papers that follow it. Under suitable conditions, the variance reduction parameter $\vartheta_{\mathbf{n}}$ converges to an optimal choice, and the average over all replications is a consistent, asymptotically normal estimator. Still, it would also be well to confirm the bias is negligible at the relevant sample sizes. Another issue is how many replications should be in each stage, between updates of ϑ. Although classic SA procedures may update ϑ after each replication, that will usually entail too much computational effort when the goal is variance reduction, or rather, a reduction in work-normalized variance.

The approach that approximates a value function V is surveyed by [73] in a Markov-chain setting. In finance, $V(t,\mathbf{S}(t))$ may be an option's price when the underlying is $\mathbf{S}(t)$ at time t, for example. An approximate value function $\hat{V}$ is built by metamodeling (§4). Adaptive control variates work by using $\hat{V}$ to construct a martingale whose ith increment is $\hat{V}(t_i,\mathbf{S}(t_i)) - \mathrm{E}[\hat{V}(t_i,\mathbf{S}(t_i))|\mathscr{F}_{i-1}]$, and using it as a control variate. Adaptive importance sampling works by setting the likelihood ratio for step i to $\hat{V}(t_i,\mathbf{S}(t_i))/\mathrm{E}[\hat{V}(t_i,\mathbf{S}(t_i))|\mathscr{F}_{i-1}]$. For the sake of computational efficiency, $\hat{V}$ should be such that $\mathrm{E}[\hat{V}(t_i,\mathbf{S}(t_i))|\mathscr{F}_{i-1}]$ can be computed in closed form. If the true value function V could be substituted for the approximation $\hat{V}$, then the control variate or importance sampling would be perfect, resulting in zero variance [63]. A bridge between the two approaches is [66], using SA and SAA methods to construct $\hat{V}$ by minimizing the variance that remains after it is used to provide a control variate. In finance, this approach to adaptive Monte Carlo has been used above all for American options: [30, 63] use SAA and regression metamodeling for this purpose. Because metamodeling is commonly used anyway in American option pricing, to identify a good exercise policy, the marginal computational cost of using the metamodel to find a good control variate or importance sampling distribution can be small, making this adaptive Monte Carlo approach very attractive.

6 Risk Management

Monte Carlo methods in risk management are an active area of research. A straightforward Monte Carlo approach is to sample scenarios $\mathbf{S}^{(1)}(T)$, ..., $\mathbf{S}^{(n)}(T)$ and in each scenario to compute P&L $V(T,\mathbf{S}(T)) - V(0,\mathbf{S}(0))$, the change in the portfolio's value by time T. It is natural to have a high dimensional for $\mathbf{S}$ because a portfolio's value can depend on many factors. There are two main computational challenges in risk measurement.

One challenge is that risk measures such as VaR and CVaR focus on the left tail of the distribution of P&L, containing large losses. It is a moderately rare event for loss to exceed VaR, so straightforward Monte Carlo estimation of a large portfolio's risk can be slow. This makes it worthwhile to pursue variance reduction: see [46, Ch. 9] for general techniques and [8, 25, 48, 49, 51] for techniques specific to credit risk.

The second challenge arises when the portfolio value function $V(T,\cdot)$ is unknown, so P&L in each scenario must be estimated by Monte Carlo. This leads to a computationally expensive nested simulation (§4): simulation of scenarios under P (as in the lower left corner of Figure 1) and a nested simulation under Q conditional on each scenario, to estimate the portfolio value $V(T,\mathbf{S}(T))$ in that scenario. In particular, nested simulation is generally biased, which causes a poor rate of convergence for the Monte Carlo estimate as the computational budget grows. This makes it worthwhile to explore ways to make the simulation more efficient:

- Jackknifing can reduce the bias [54, 74].
- Although variance is not always a good portfolio risk measure, it can be useful in evaluating hedging strategies. Unbiased estimation of the variance of P&L by nested simulation is possible. Indeed, a nested simulation with small computational effort devoted to each scenario, and thus inaccurate estimation of P&L in each scenario, can provide an accurate estimator of the variance of P&L [94].
- It helps to optimize the number n of scenarios to minimize MSE or confidence interval width given a fixed computational budget [54, 68].
- When the risk measure emphasizes the left tail of the distribution, is desirable to allocate more computational effort to simulating the scenarios that seem likely to be near VaR (when estimating VaR) or to belong to the left tail (for CVaR). This suggests adaptive simulation procedures, in which the allocation of replications at one stage depends on information gathered at previous stages. One approach is to eliminate scenarios once they seem unlikely to belong to the left tail [70, 78]. Another is to make the number of replications somehow inversely proportional to the estimated distance from a scenario to the left tail or its boundary [54].
- Metamodeling (§4) and database Monte Carlo (§5) can be useful in portfolio risk measurement because it involves many similar simulation problems: estimating P&L in many scenarios. Metamodeling can be successful because P&L is often a well-behaved function of the scenario. It has been applied in [9] and in an adaptive procedure for estimating CVaR by [79], where more computational effort is allocated to design points near scenarios with large losses.

Estimating sensitivities of risk measures is studied in [47, 61, 62, 76]. They can provide risk components or be useful in optimization.

7 Financial Optimization Problems

The finance problem most clearly linked to optimization is portfolio optimization. Before discussing Monte Carlo methods for portfolio optimization, we turn to American option pricing. It involves a simple optimization problem, and Monte Carlo methods for American option pricing have been more thoroughly studied. See Section 4 for background on optimization via simulation.

American Options. Monte Carlo is best suited for *European options*, which can be exercised only at maturity. *American options* can be exercised at any time until maturity. The owner of an American option faces an optimal stopping problem. Let τ represent the exercise policy: the random variable $\tau = \tau(\mathbf{U})$ is the stopping time at which exercise occurs. The resulting payoff is $f(\mathbf{U};\tau)$. Pricing methods for American options involve computing the optimal exercise policy τ^* that maximizes the value $\mathrm{E}[f(\mathbf{U};\tau)]$ of the option, while computing the price $\mathrm{E}[f(\mathbf{U};\tau^*)]$. It is optimal to exercise at time t if the payoff $f(\mathbf{U};t)$ of doing so exceeds the *continuation value*, the conditional expectation of the payoff earned by exercising at the optimal time after t. Because a continuous-time optimal stopping problem is troublesome for simulation, much research on the topic of American options actually deals with *Bermudan options*, which can be exercised at any one of the times $\{t_1,\ldots,t_m\}$. A Bermudan option with a sufficiently large set of possible exercise times is treated as an approximation of an American option. Even Bermudan options are not straightforward to price by Monte Carlo methods: at every step on every path, one needs to know the continuation value to make the optimal decision about whether to exercise. A naive approach, which is impractical due to excessive computational requirements, is nested simulation (§4): at every step on every path, estimate the continuation value by an inner-level simulation. For overviews of Monte Carlo methods in American option pricing, see [15, 24, 39, 46]. Here we merely emphasize connections to themes of financial simulation.

- The most popular approach to American option pricing, regression-based Monte Carlo, is a form of approximate dynamic programming (ADP). The optimal stopping problem is relatively easy for ADP because there are only two actions, continue or exercise, and they do not affect the dynamics of the underlying.
- After choosing a sub-optimal exercise policy τ and sampling $\mathbf{U}$ independently, $f(\mathbf{U};\tau)$ is an estimator of the American option price with negative bias. Duality yields an estimator with positive bias: see [56] and references therein, particularly [2]. This enables a conservative confidence interval that is asymptotically valid for large simulation sample sizes. A bias reduction method is developed in [65].
- Adaptive Monte Carlo (§5) is very useful in American option pricing. It is connected to duality: according to [63], “the perfect control variate solves the ad-

ditive duality problem and the perfect importance sampling estimator solves the multiplicative duality problem."

American option pricing remains an active research area because there are many rival methods that are amenable to improvement. There is potential to gain efficiency by adaptive simulation that allocates extra simulation effort to design points near the boundary where estimated exercise and continuation values are equal. High-dimensional problems remain challenging. It would also be good to better understand and to reduce the error in approximating an American by a Bermudan option.

Portfolio Optimization. An introduction to this topic, stressing the connection between American option pricing and portfolio optimization, while emphasizing the value of dual methods, is [56]. The purpose of the dual methods is to provide an upper bound on the optimal expected utility: one can use simulation to estimate both the expected utility a candidate portfolio strategy provides and the upper bound on the optimal expected utility, and compare these estimates to see if the candidate is nearly optimal [57]. Other ADP methods in portfolio optimization include [17, 81, 95]. ADP is not the only Monte Carlo approach to portfolio optimization. For an overview, see [14]. Another method uses Monte Carlo to estimate conditional expectations involving Malliavin derivatives, which are proved to be the optimal portfolio weights for a portfolio optimization in a complete market [29].

8 Sensitivity Analysis

Many problems in finance call for estimation of the sensitivity $\mu'(\theta)$ of a mean $\mu(\theta)$ to a parameter θ: the Greeks are of direct interest in hedging, and sensitivities are needed in gradient-based optimization. Approaches to estimating sensitivities via simulation include:

- *Finite differences* (FD). Run the simulation at two values θ_1 and θ_2 in the neighborhood of θ, using common random numbers. The FD estimator is $(f(\mathbf{U};\theta_1) - f(\mathbf{U};\theta_2))/(\theta_1 - \theta_2)$. This approach is biased and computationally inefficient.
- Metamodeling (M, §4) can be viewed as a variant of FD that is helpful when estimating sensitivities with respect to many parameters: where FD would require running many simulations, metamodeling can provide an answer based on simulations at only a few design points. To estimate first-order sensitivities, fit a linear metamodel locally, in a neighborhood of θ. To get second-order sensitivities too, fit a quadratic metamodel locally.
- The *pathwise* (PW) method, known outside finance as *infinitesimal perturbation analysis* (IPA). Under some conditions, $\mu'(\theta) = \mathrm{E}[Y'(\theta)]$, so an unbiased estimator is $Y'(\theta) = (\partial f/\partial\theta)(\mathbf{U};\theta)$. It may be easy to compute this if θ is a parameter, such as a strike price, that has a simple, direct effect on the payoff, but it might be hard if θ is a parameter that governs the distributions of random variables in the simulation. This method can only be applied if Y is suitably differentiable; there are a number of cases in finance in which it does not apply.

- *Smoothed perturbation analysis* (SPA) is an extension of IPA. It works by reformulating the simulation model: if there is a conditional expectation $\tilde{Y}(\theta) = \mathrm{E}[Y(\theta)|\mathscr{F}]$ that can be computed and $\tilde{Y}$ is a smoother function of θ than Y is, then the estimator $\tilde{Y}'(\theta)$ can be used when IPA does not apply. This approach requires the analyst to identify a good set of information $\mathscr{F}$ on which to condition, and to compute the conditional expectation.
- IPA can have problems in first or second derivative estimation because of discontinuity or non-differentiability of the integrand in the commonplace case where $Y(\theta) = f(\mathbf{U};\theta)$ has the form $f_1(\mathbf{U};\theta)1\{f_2(\mathbf{U};\theta) \geq 0\}$. Kernel smoothing leads to the estimator

  $$\frac{\partial f_1}{\partial \theta}(\mathbf{U};\theta)1\left\{f_2(\mathbf{U};\theta) \geq 0\right\} + \frac{1}{\delta} f_1(\mathbf{U};\theta)\frac{\partial f_2}{\partial \theta}(\mathbf{U};\theta)\phi\left(\frac{f_2(\mathbf{U};\theta)}{\delta}\right),$$

  where ϕ is the kernel and δ is the bandwidth [77]. In contrast to SPA, kernel smoothing requires no analyst ingenuity: a Gaussian kernel and automated bandwidth selection perform well. This estimator is biased, although it is consistent under some conditions which may be hard to verify.
- The *likelihood ratio* (LR) method, also known outside finance as the *score function* method, involves differentiating a density $g(\cdot;\theta)$ instead of differentiating a payoff. Here we require a representation $\mu(\theta) = \int f(\mathbf{u};\theta)\,\mathrm{d}\mathbf{u} = \int \tilde{f}(\mathbf{x})g(\mathbf{x};\theta)\,\mathrm{d}\mathbf{x}$, framing the simulation as sampling the random vector $\mathbf{X}(\mathbf{U};\theta)$ which has density $g(\cdot;\theta)$. In the new representation, $Y(\theta) = f(\mathbf{U};\theta) = \tilde{f}(\mathbf{X}(\mathbf{U};\theta))$, so $\tilde{f}$ has no explicit dependence on θ: applying the method requires θ to be a parameter only of the density. Under some conditions,

  $$\mu'(\theta) = \int \tilde{f}(\mathbf{x})\frac{\partial g(\mathbf{x};\theta)/\partial\theta}{g(\mathbf{x};\theta)} g(\mathbf{x};\theta)\,\mathrm{d}\mathbf{x} = \mathrm{E}\left[Y(\theta)\frac{\partial g(\mathbf{X};\theta)/\partial\theta}{g(\mathbf{X};\theta)}\right],$$

  so an unbiased estimator is $Y(\theta)(\partial g(\mathbf{X};\theta)/\partial\theta)/g(\mathbf{X};\theta)$. If the density is not known in closed form, one may apply the LR method instead to a discretized version of the underlying stochastic process.
- Malliavin calculus can provide estimators of sensitivities. Implementing these estimators generally requires that time be discretized. The resulting estimators are asymptotically equivalent, as the number of time steps $m \to \infty$, to combinations of PW and LR estimators for the discretized process [23]. Combinations of PW and LR methods are also used to overcome the limitations of PW and of LR in isolation. For a unified view of the PW and LR methods, see [71].
- The method of *weak derivatives* (WD) can be explained based on LR [37]: suppose $\partial g(\mathbf{x};\theta)/\partial\theta$ can be written in the form $c(\theta)(g_1(\mathbf{x};\theta) - g_2(\mathbf{x};\theta))$, where $g_1(\cdot;\theta)$ and $g_2(\cdot;\theta)$ are densities. If the LR approach is valid, then

  $$\begin{aligned}\mu'(\theta) &= c(\theta)\left(\int \tilde{f}(\mathbf{x})g_1(\mathbf{x};\theta)\,\mathrm{d}\mathbf{x} - \int \tilde{f}(\mathbf{x})g_2(\mathbf{x};\theta)\,\mathrm{d}\mathbf{x}\right)\\ &= c(\theta)\mathrm{E}\left[\tilde{f}(\mathbf{X}_1) - \tilde{f}(\mathbf{X}_2)\right],\end{aligned}$$

where $\mathbf{X}_1$ and $\mathbf{X}_2$ are sampled according to the densities $g_1(\cdot;\theta)$ and $g_2(\cdot;\theta)$ respectively: an unbiased estimator is $c(\theta)(\tilde{f}(\mathbf{X_1}) - \tilde{f}(\mathbf{X_2}))$. (However, the WD approach does not actually require differentiating the density.) Here we did not specify how the original pseudo-random numbers would be used to simulate $\mathbf{X}_1$ and $\mathbf{X}_2$. The whole structure of the simulation is changed, and the dependence or *coupling* of $\mathbf{X}_1$ and $\mathbf{X}_2$ has a major effect on the estimator's variance.

For introductions to these methods, see [24, 37, 43, 46]. Important early references include [19, 38]. The different methods have different realms of applicability and, when two of them apply, they can yield estimators with very different variances.

A recent advance has been in speeding up PW computations of multiple Greeks of the same derivative security price using adjoint methods [43, 45]. Another active area of research is estimation of sensitivities when the underlying stochastic process has jumps: see e.g. [52]. A further topic for future work is the application of WD to estimating sensitivities in financial simulation: although weak derivatives were applied to simulating the sensitivities of option prices in [58], the WD method has not received enough attention in finance. For results on WD when underlying distributions are normal, as happens in many financial models, see [59].

9 Discretization of Stochastic Differential Equations

Many financial simulations involve *stochastic differential equations* (SDEs). The solution $\mathbf{S}$ to an SDE is a continuous-time stochastic process, but it is standard to discretize time and simulate $\mathbf{S}(t_1),\ldots,\mathbf{S}(t_m)$. In some models, it is possible to simulate *exactly*, that is, from the correct distribution for $(\mathbf{S}(t_1),\ldots,\mathbf{S}(t_m))$. However, in many models, it is not known how to do so. *Discretization error* is the difference between the distribution of $(\mathbf{S}(t_1),\ldots,\mathbf{S}(t_m))$ as simulated and the distribution it should have according to the SDE. Discretization error causes *discretization bias* in the Monte Carlo estimator. To reduce the discretization bias, one increases the number m of steps, which increases the computational cost of simulating $\mathbf{S}(t_1),\ldots,\mathbf{S}(t_m)$. On quantifying and reducing this discretization bias, see [46, 67], or [24, 53] for introductions. Some research on SDE discretization is specific to one model, that is, to one SDE, while some is generic.

Model-specific research may consist of showing how to simulate a certain model exactly or how to reduce discretization error. For example, recently there have been major improvements in simulating the Heston model [1, 20, 50]. On simulation of Lévy processes, see [5] and [27, Ch. 6]. Lévy processes used in finance include VG and CGMY: on simulating these, see [6, 36, 64, 87].

The generic research includes the study of different discretization schemes and the rate at which discretization bias decreases as the number m of steps increases. This rate may be unaffected by replacing the normal random variables typically used in SDE discretization by simpler random variables which are faster to simulate, e.g. having discrete distributions with only three values [46, pp. 355-6]. It would be interesting to explore the application of quasi-Monte Carlo to a simulation scheme using these discrete random variables. One active research topic, based on [86], involves new discretization schemes, the quadratic Milstein scheme and a two-stage Runge-Kutta scheme, along with a new criterion, microscopic total variation, for assessing a scheme's quality.

We next consider two important recent developments in simulating SDEs.

Multi-Grid Extrapolation. One method for reducing discretization error is *extrapolation* [46, §6.2.4]. Let $\hat{\mu}(m)$ be a simulation estimator based on discretizing an SDE with m time steps, and $\hat{\mu}(2m)$ be the estimator when $2m$ time steps are used. Because of bias cancelation, the estimator $2\hat{\mu}(2m) - \hat{\mu}(m)$ can have lower bias and a better rate of convergence. This idea is extended by [44] to multiple grids of different fineness, instead of just two. The estimator given L grids, with N_ℓ paths simulated on the ℓth grid which has m_ℓ steps, is $\sum_{\ell=1}^{L} \sum_{i=1}^{N_\ell} (\hat{\mu}^{(i)}(m_\ell) - \hat{\mu}^{(i)}(m_{\ell-1}))/N_\ell$, where $\hat{\mu}^{(i)}(m_\ell)$ involves simulating the same Wiener process sample path $\{W^{(i)}(t)\}_{0 \le t \le T}$ for all grids. It is efficient to simulate fewer paths using the fine grids than with the coarse grids. For one thing, even if N_ℓ is small for a fine grid, including this ℓth grid contributes a correction term $\mathrm{E}[\hat{\mu}^{(i)}(m_\ell) - \hat{\mu}^{(i)}(m_{\ell-1})]$ that reduces bias. Furthermore, simulating paths on a fine grid is computationally expensive, while the variance of $\hat{\mu}^{(i)}(m_\ell) - \hat{\mu}^{(i)}(m_{\ell-1})$ tends to be small for the fine grids. Consequently, computational resources are better spent on coarser grids where it is cheap to attack large components of the variance. The result is reduced bias and better rates of convergence. QMC should be useful particularly when applied to the coarser grids. A related approach involving multiple grids [91] is based on the idea that coarse grids provide biased control variates [90].

Exact Simulation of SDEs. Surprisingly, it is sometimes possible to simulate a scalar diffusion S exactly even when it is not possible to integrate the SDE in closed form to learn the distribution of $(S(t_1), \ldots, S(t_m))$ [12, 22]. The basic idea is to sample according to the law of S by acceptance-rejection sampling of paths of a Wiener process W. If a path $\{W(t)\}_{0 \le t \le T}$ is accepted with probability proportional to the Radon-Nikodym derivative between the law of the S and the law of W, the path is sampled from the law of S. The log of the Radon-Nikodym derivative has the form $A(W(T)) - \int_0^T \phi(t, W(t))\,\mathrm{d}t$ where A and ϕ depend on the coefficients of the SDE. The problem lies in simulating $\int \phi(t, W(t))\,\mathrm{d}t$, which is an awkward functional of the entire continuous-time path $\{W(t)\}_{0 \le t \le T}$. The key insight is that $\exp(-\int_0^T \phi(t, W(t))\,\mathrm{d}t)$ is the conditional probability, given the path of the Wiener process, that no arrivals occur by time T in a doubly stochastic Poisson process whose arrival rate at time t is $\phi(t, W(t))$. This may be simulated by straightforward or sophisticated stochastic thinning procedures, depending on the characteristics of the function ϕ [12, 22, 42]. This approach is a significant development: it is of theoretical interest and, when applicable, it eliminates the need for the analyst to quantify and reduce discretization bias. More work is needed to render this approach widely applicable in finance and to study the efficiency gains it produces. Acceptance-rejection sampling can be very slow, when the acceptance probability is low, so this way of simulating SDEs exactly could be slower to attain a target MSE than existing methods of SDE discretization. The speed of acceptance-rejection sampling can be improved by drawing the original samples from another law. When the Radon-Nikodym derivative between the law of S and the original sampling law is smaller, acceptance occurs faster. In this case, one might think of drawing the original samples from the law of some other integrable Itō process, not a Wiener process. For example, one might sample from the law of geometric Brownian motion or of an Ornstein-Uhlenbeck process, because in many financial models, S is closer to these than to a Wiener process. An interesting question is how best to choose the original sampling law given the SDE one wishes to simulate.

Acknowledgements The author acknowledges the support of the National Science Foundation under Grant No. DMI-0555485. He is very grateful to Mark Broadie, Michael Fu, Kay Giesecke, Paul Glasserman, Michael Gordy, Bernd Heidergott, Shane Henderson, Jeff Hong, Pierre L'Ecuyer, Elaine Spiller, Pirooz Vakili, and an anonymous referee for providing comments, corrections, and references which led to major improvements to this article.

References

1. Leif Andersen. Efficient simulation of the Heston stochastic volatility model. Working paper, Banc of America Securities, January 2007.
2. Leif Andersen and Mark N. Broadie. A primal-dual simulation algorithm for pricing multidimensional American options. *Management Science*, 50(9):1222–1234, 2004.
3. Bruce Ankenman, Barry L. Nelson, and Jeremy Staum. Stochastic kriging for simulation metamodeling. *Operations Research*. Forthcoming.
4. Bouhari Arouna. Adaptative Monte Carlo method, a variance reduction technique. *Monte Carlo Methods and Applications*, 10(1):1–24, 2004.
5. Søren Asmussen and Jan Rosiński. Approximations of small jumps of Lévy processes with a view towards simulation. *Journal of Applied Probability*, 38(2):482–493, 2001.
6. Athanassios N. Avramidis and Pierre L'Ecuyer. Efficient Monte Carlo and quasi-Monte Carlo option pricing under the variance-gamma model. *Management Science*, 52(12):1930–1944, 2006.
7. Russell R. Barton and Martin Meckesheimer. Metamodel-based simulation optimization. In S. G. Henderson and B. L. Nelson, editors, *Simulation*, Handbooks in Operations Research and Management Science, pages 535–574. Elsevier, Amsterdam, 2006.
8. Achal Bassamboo, Sandeep Juneja, and Assaf Zeevi. Portfolio credit risk with extremal dependence. *Operations Research*, 56(3):593–606, 2008.
9. R. Evren Baysal, Barry L. Nelson, and Jeremy Staum. Response surface methodology for hedging and trading strategies. In S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, editors, *Proceedings of the 2008 Winter Simulation Conference*, pages 629–637, Piscataway, N. J., 2008. IEEE Press.
10. Sana Ben Hamida and Rama Cont. Recovering volatility from option prices by evolutionary optimization. *Journal of Computational Finance*, 8(4):43–76, 2005.
11. Dimitri P. Bertsekas and John Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Nashua, N.H., 1996.
12. Alexandros Beskos and Gareth O. Roberts. Exact simulation of diffusions. *Annals of Applied Probability*, 15(4):2422–2444, 2005.
13. Tomasz R. Bielecki, Stéphane Crépey, Monique Jeanblanc, and Marek Rutkowski. Valuation of basket credit derivatives in the credit migrations environment. In J. R. Birge and V. Linetsky, editors, *Financial Engineering*, Handbooks in Operations Research and Management Science, pages 471–507. Elsevier, Amsterdam, 2008.

14. John R. Birge. Optimization methods in dynamic portfolio management. In J. R. Birge and V. Linetsky, editors, *Financial Engineering*, Handbooks in Operations Research and Management Science, pages 845–865. Elsevier, Amsterdam, 2008.
15. Nomesh Bolia and Sandeep Juneja. Monte Carlo methods for pricing financial options. *Sādhanā*, 30(2-3):347–385, 2005.
16. Tarik Borogovac and Pirooz Vakili. Control variate technique: a constructive approach. In S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, editors, *Proceedings of the 2008 Winter Simulation Conference*, pages 320–327, Piscataway, N. J., 2008. IEEE Press.
17. Michael W. Brandt, Amit Goyal, Pedro Santa-Clara, and Jonathan R. Stroud. A simulation approach to dynamic portfolio choice with an application to learning about return predictability. *Review of Financial Studies*, 18(3):831–873, 2005.
18. Mark N. Broadie, Deniz M. Cicek, and Assaf Zeevi. General bounds and finite-time improvement for stochastic approximation algorithms. Working paper, Columbia University, February 2009. Available via `http://www2.gsb.columbia.edu/faculty/azeevi`.
19. Mark N. Broadie and Paul Glasserman. Estimating security price derivatives using simulation. *Management Science*, 42(2):269–285, 1996.
20. Mark N. Broadie and Özgur Kaya. Exact simulation of stochastic volatility and other affine jump diffusion processes. *Operations Research*, 54(2):217–231, 2006.
21. Luca Capriotti. Least squares importance sampling for Monte Carlo security pricing. *Quantitative Finance*, 8(5):485–497, 2008.
22. Nan Chen. Localization and exact simulation of Brownian motion driven stochastic differential equations. Working paper, Chinese University of Hong Kong, May 2009.
23. Nan Chen and Paul Glasserman. Malliavin Greeks without Malliavin calculus. *Stochastic Processes and their Applications*, 117:1689–1723, 2007.
24. Nan Chen and L. Jeff Hong. Monte Carlo simulation in financial engineering. In S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, editors, *Proceedings of the 2007 Winter Simulation Conference*, pages 919–931, Piscataway, N. J., 2007. IEEE Press.
25. Zhiyong Chen and Paul Glasserman. Fast pricing of basket default swaps. *Operations Research*, 56(2):286–303, 2008.
26. Thomas F. Coleman, Yuying Li, and Maria-Cristina Patron. Total risk minimization using Monte Carlo simulations. In J. R. Birge and V. Linetsky, editors, *Financial Engineering*, Handbooks in Operations Research and Management Science, pages 593–635. Elsevier, Amsterdam, 2008.
27. Rama Cont and Peter Tankov. *Financial Modelling with Jump Processes*. Chapman & Hall/CRC, Boca Raton, 2004.
28. Gerard Cornuejols and Reha Tütüncü. *Optimization Methods in Finance*. Cambridge University Press, New York, 2007.
29. Jérôme Detemple, René Garcia, and Marcel Rindisbacher. Intertemporal asset allocation: a comparison of methods. *Journal of Banking and Finance*, 29:2821–2848, 2005.
30. Samuel M. T. Ehrlichman and Shane G. Henderson. Adaptive control variates for pricing multi-dimensional American options. *Journal of Computational Finance*, 11(1), 2007.
31. Markus Emsermann and Burton Simon. Improving simulation efficiency with quasi control variates. *Stochastic Models*, 18(3):425–448, 2002.
32. Frank J. Fabozzi, editor. *The Handbook of Mortgage-Backed Securities*. McGraw-Hill, New York, 5th edition, 2001.
33. Frank J. Fabozzi, Petter N. Kolm, Dessislava Pachamanova, and Sergio M. Focardi. *Robust Portfolio Optimization and Management*. John Wiley & Sons, Hoboken, N. J., 2007.
34. Michael C. Fu. Optimization for simulation: theory vs. practice. *INFORMS Journal on Computing*, 14(3):192–215, 2002.
35. Michael C. Fu. Gradient estimation. In S. G. Henderson and B. L. Nelson, editors, *Simulation*, Handbooks in Operations Research and Management Science, pages 575–616. Elsevier, Amsterdam, 2006.

36. Michael C. Fu. Variance gamma and Monte Carlo. In M. C. Fu, R. A. Jarrow, J.-Y. J. Yen, and R. J. Elliott, editors, *Advances in Mathematical Finance*, pages 21–34. Springer-Verlag, New York, 2008.
37. Michael C. Fu. What you should know about simulation and derivatives. *Naval Research Logistics*, 55(8):723–736, 2008.
38. Michael C. Fu and Jian-Qiang Hu. Sensitivity analysis for Monte Carlo simulation of option pricing. *Probability in the Engineering and Informational Sciences*, 9(3):417–446, 1995.
39. Michael C. Fu, Scott B. Laprise, Dilip B. Madan, Yi Su, and Rongwen Wu. Pricing American options: a comparison of Monte Carlo simulation approaches. *Journal of Computational Finance*, 4(3):39–88, 2001.
40. Kay Giesecke. Portfolio credit risk: top down vs. bottom up approaches. In R. Cont, editor, *Frontiers in Quantitative Finance: Credit Risk and Volatility Modeling*, pages 251–268. John Wiley & Sons, Hoboken, N. J., 2008.
41. Kay Giesecke. An overview of credit derivatives. Working paper, Stanford University, March 2009. Available via `http://www.stanford.edu/dept/MSandE/people/faculty/giesecke/publications.html`.
42. Kay Giesecke, Hossein Kakavand, and Mohammad Mousavi. Simulating point processes by intensity projection. In S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, editors, *Proceedings of the 2008 Winter Simulation Conference*, pages 560–568, Piscataway, N. J., 2008. IEEE Press.
43. Michael B. Giles. Monte Carlo evaluation of sensitivities in computational finance. In E. A. Lipitakis, editor, *HERCMA 2007 Conference Proceedings*, 2007. Available via `http://www.aueb.gr/pympe/hercma/proceedings2007/H07-FULL-PAPERS-1/GILES-INVITED-1.pdf`.
44. Michael B. Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008.
45. Michael B. Giles and Paul Glasserman. Smoking adjoints: fast Monte Carlo Greeks. *Risk*, 19:88–92, 2006.
46. Paul Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer-Verlag, New York, 2004.
47. Paul Glasserman. Measuring marginal risk contributions in credit portfolios. *Journal of Computational Finance*, 9(1):1–41, 2005.
48. Paul Glasserman. Calculating portfolio credit risk. In J. R. Birge and V. Linetsky, editors, *Financial Engineering*, Handbooks in Operations Research and Management Science, pages 437–470. Elsevier, Amsterdam, 2008.
49. Paul Glasserman, Wanmo Kang, and Perwez Shahabuddin. Fast simulation of multifactor portfolio credit risk. *Operations Research*, 56(5):1200–1217, 2008.
50. Paul Glasserman and Kyoung-Kuk Kim. Gamma expansion of the Heston stochastic volatility model. *Finance and Stochastics*. Forthcoming.
51. Paul Glasserman and Jingyi Li. Importance sampling for portfolio credit risk. *Management Science*, 51(11):1643–1656, 2005.
52. Paul Glasserman and Zongjian Liu. Estimating Greeks in simulating Lévy-driven models. Working paper, Columbia University, October 2008. Available via `http://www.paulglasserman.net`.
53. Peter W. Glynn. Monte Carlo simulation of diffusions. In S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, editors, *Proceedings of the 2008 Winter Simulation Conference*, pages 556–559, Piscataway, N. J., 2008. IEEE Press.
54. Michael B. Gordy and Sandeep Juneja. Nested simulation in portfolio risk measurement. Finance and Economics Discussion Series 2008-21, Federal Reserve Board, April 2008. Available via `http://www.federalreserve.gov/Pubs/feds/2008/200821`.
55. Robert B. Gramacy and Herbert K. H. Lee. Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–1130, 2008.

56. Martin B. Haugh and Leonid Kogan. Duality theory and approximate dynamic programming for pricing American options and portfolio optimization. In J. R. Birge and V. Linetsky, editors, *Financial Engineering*, Handbooks in Operations Research and Management Science, pages 925–948. Elsevier, Amsterdam, 2008.
57. Martin B. Haugh, Leonid Kogan, and Jiang Wang. Evaluating portfolio policies: a dual approach. *Operations Research*, 54(3):405–418, 2006.
58. Bernd Heidergott. Option pricing via Monte Carlo simulation: a weak derivative approach. *Probability in the Engineering and Informational Sciences*, 15:335–349, 2001.
59. Bernd Heidergott, Felisa J. Vázquez-Abad, and Warren Volk-Makarewicz. Sensitivity estimation for Gaussian systems. *European Journal of Operational Research*, 187:193–207, 2008.
60. Shane G. Henderson and Sujin Kim. The mathematics of continuous-variable simulation optimization. In S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, editors, *Proceedings of the 2008 Winter Simulation Conference*, pages 122–132, Piscataway, N. J., 2008. IEEE Press.
61. L. Jeff Hong. Estimating quantile sensitivities. *Operations Research*, 57(1):118–130, 2009.
62. L. Jeff Hong and Guangwu Liu. Simulating sensitivities of conditional value at risk. *Management Science*, 55(2):281–293, 2009.
63. Sandeep Juneja and Himanshu Kalra. Variance reduction techniques for pricing American options. *Journal of Computational Finance*, 12(3):79–102, 2009.
64. Vladimir K. Kaishev and Dimitrina S. Dimitrova. Dirichlet bridge sampling for the variance gamma process: pricing path-dependent options. *Management Science*, 55(3):483–496, 2009.
65. K. H. Felix Kan, R. Mark Reesor, Tyson Whitehead, and Matt Davison. Correcting the bias in Monte Carlo estimators of American-style option values. Submitted to *Monte Carlo and Quasi-Monte Carlo Methods 2008.*
66. Sujin Kim and Shane G. Henderson. Adaptive control variates for finite-horizon simulation. *Mathematics of Operations Research*, 32(3):508–527, 2007.
67. Peter E. Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations*. Springer-Verlag, New York, 1992.
68. Hai Lan. Tuning the parameters of a two-level simulation procedure with screening. Working paper, Northwestern University, available via `http://users.iems.northwestern.edu/~staum`, March 2009.
69. Hai Lan, Barry L. Nelson, and Jeremy Staum. Two-level simulations for risk management. In S. Chick, C.-H. Chen, S. G. Henderson, and E. Yücesan, editors, *Proceedings of the 2007 INFORMS Simulation Society Research Workshop*, pages 102–107, Fontainebleau, France, 2007. INSEAD. Available via `http://www.informs-cs.org/2007informs-csworkshop/23.pdf`.
70. Hai Lan, Barry L. Nelson, and Jeremy Staum. Confidence interval procedures for expected shortfall risk measurement via two-level simulation. Working paper 08-02, Department of IEMS, Northwestern University, November 2008. Available via `http://users.iems.northwestern.edu/~staum`.
71. Pierre L'Ecuyer. A unified view of the IPA, SF, and LR gradient estimation techniques. *Management Science*, 36(11):1364–1383, 1990.
72. Pierre L'Ecuyer. Quasi-Monte Carlo methods with applications in finance. Les Cahiers du GERAD G-2008-55, Université de Montréal, August 2008. To appear in *Finance and Stochastics.*
73. Pierre L'Ecuyer and Bruno Tuffin. Approximate zero-variance simulation. In S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, editors, *Proceedings of the 2008 Winter Simulation Conference*, pages 170–181, Piscataway, N. J., 2008. IEEE Press.
74. Shing-Hoi Lee. *Monte Carlo computation of conditional expectation quantiles*. PhD thesis, Stanford University, 1998.
75. Vadim Lesnevski, Barry L. Nelson, and Jeremy Staum. Simulation of coherent risk measures based on generalized scenarios. *Management Science*, 53(11):1756–1769.
76. Guangwu Liu and L. Jeff Hong. Kernel estimation of quantile sensitivities. *Naval Research Logistics*. Forthcoming.

77. Guangwu Liu and L. Jeff Hong. Pathwise estimation of the Greeks of financial options. Working paper, Hong Kong University of Science and Technology, August 2008. Available via `http://ihome.ust.hk/~liugw`.
78. Ming Liu, Barry L. Nelson, and Jeremy Staum. An adaptive procedure for point estimation of expected shortfall. Working paper 08-03, Department of IEMS, Northwestern University, October 2008. Available via `http://users.iems.northwestern.edu/~staum`.
79. Ming Liu and Jeremy Staum. Estimating expected shortfall with stochastic kriging. Working paper, Northwestern University, March 2009. Available via `http://users.iems.northwestern.edu/~staum`.
80. Alexander J. McNeil, Rüdiger Frey, and Paul Embrechts. *Quantitative Risk Management*. Princeton University Press, Princeton, N. J., 2005.
81. Kumar Muthuraman and Haining Zha. Simulation-based portfolio optimization for large portfolios with transactions costs. *Mathematical Finance*, 18(1):115–134, 2008.
82. Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.
83. Sigurdur Ólafsson. Metaheuristics. In S. G. Henderson and B. L. Nelson, editors, *Simulation*, Handbooks in Operations Research and Management Science, pages 633–654. Elsevier, Amsterdam, 2006.
84. Soumik Pal. Computing strategies for achieving acceptability: a Monte Carlo approach. *Stochastic Processes and their Applications*, 117(11):1587–1605, 2007.
85. Raghu Pasupathy, Bruce W. Schmeiser, Michael R. Taafe, and Jin Wang. Control variate estimation using estimated control means. *IIE Transactions*. Forthcoming.
86. Jose Antonio Perez. *Convergence of numerical schemes in the total variation sense*. PhD thesis, Courant Institute of Mathematical Sciences, New York University, 2004.
87. Jérémy Poirot and Peter Tankov. Monte Carlo option pricing for tempered stable (CGMY) processes. *Asia-Pacific Financial Markets*, 13:327–344, 2006.
88. Warren B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley & Sons, Hoboken, N. J., 2007.
89. Warren B. Powell. What you should know about approximate dynamic programming. *Naval Research Logistics*, 56(3):239–249, 2009.
90. Bruce W. Schmeiser, Michael R. Taafe, and Jin Wang. Biased control-variate estimation. *IIE Transactions*, 33(3):219–228, 2001.
91. Adam Speight. A multilevel approach to control variates. *Journal of Computational Finance*. Forthcoming.
92. Jeremy Staum. Incomplete markets. In J. R. Birge and V. Linetsky, editors, *Financial Engineering*, Handbooks in Operations Research and Management Science, pages 511–563. Elsevier, Amsterdam, 2008.
93. Yi Su and Michael C. Fu. Optimal importance sampling in securities pricing. *Journal of Computational Finance*, 5(4):27–50, 2002.
94. Yunpeng Sun, Daniel W. Apley, and Jeremy Staum. $1\frac{1}{2}$-level simulation for estimating the variance of a conditional expectation. Working paper, Northwestern University, 2009.
95. Jules H. van Binsbergen and Michael W. Brandt. Solving dynamic portfolio choice problems by recursing on optimized portfolio weights or on the value function? *Computational Economics*, 29:355–367, 2007.
96. Gang Zhao, Tarik Borogovac, and Pirooz Vakili. Efficient estimation of option price and price sensitivities via structured database Monte Carlo (SDMC). In S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, editors, *Proceedings of the 2007 Winter Simulation Conference*, pages 984–990, Piscataway, N. J., 2007. IEEE Press.
97. Gang Zhao and Pirooz Vakili. Monotonicity and stratification. In S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, editors, *Proceedings of the 2008 Winter Simulation Conference*, pages 313–319, Piscataway, N. J., 2008. IEEE Press.
98. Gang Zhao, Yakun Zhou, and Pirooz Vakili. A new efficient simulation strategy for pricing path-dependent options. In L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, editors, *Proceedings of the 2006 Winter Simulation Conference*, pages 703–710, Piscataway, N. J., 2006. IEEE Press.