Importance Sampling in the Heath-Jarrow-Morton Framework

PAUL GLASSERMAN, PHILIP HEIDELBERGER, AND PERWEZ SHAHABUDDIN

PAUL GLASSERMAN is a professor at the Graduate School of Business at Columbia University in New York.

PHILIP HEIDELBERGER is a research staff member at the IBM Research Division in Yorktown Heights, New York.

PERWEZ SHAHABUDDIN is an associate professor in the IEOR department at Columbia University.

This article develops a variance-reduction technique for pricing derivatives by simulation in high-dimensional multifactor models. A premise of this work is that the greatest gains in simulation efficiency come from taking advantage of the structure of both the cash flows of a security and the model in which it is priced. For this to be feasible in practice requires automating the identification and use of relevant structure.

We exploit model and payoff structure through a combination of importance sampling and stratified sampling. The importance sampling applies a change of drift to the underlying factors; we select the drift by first solving an optimization problem. We then identify a particularly effective direction for stratified sampling (which may be thought of as an approximate numerical integration) by solving an eigenvector problem.

Examples illustrate that the combination of the methods can produce enormous variance reduction even in high-dimensional multifactor models. The method introduces some computational overhead in solving the optimization and eigenvector problems; to address this, we propose and evaluate approximate solution procedures, which enhance the applicability of the method.

Monte Carlo simulation has become an essential tool for pricing and hedging complex derivative securities and for measuring the risks in derivatives portfolios. The more realistic — and thus more complex — the pricing model used, the more likely that Monte Carlo will be the only viable numerical method for working with the model. The applicability of simulation is relatively insensitive to model details and — in sharp contrast with deterministic numerical methods — to model *dimension*. The relevant notion of dimension can vary from one setting to another, but in general increasing the number of underlying assets, the number of factors, or the number of time steps all increase dimension. Multifactor models of the evolution of the yield curve are a particularly important class of high-dimensional problems often requiring simulation.

The main limitation of Monte Carlo simulation is that it is rather slow; put a different way, the results obtained from Monte Carlo in a short amount of computing time can be very imprecise. Because numerical results obtained through Monte Carlo are statistical estimates, their precision is best measured through their standard error, which is ordinarily the ratio of a standard deviation per observation to the square root of the number of observations.

It follows that there are two ways of increasing precision: reducing the numerator, or increasing the denominator. Given a fixed time allocated for computing a price, increasing the number of observations entails using a faster machine or finding programming speed-ups; such opportunities are fairly quickly exhausted. This leaves the numerator of the standard error as the main opportunity for improvements. Variance-reduction techniques attempt to improve the precision of Monte Carlo estimates by reducing the standard deviation (and thus variance) per observation.

The literature on Monte Carlo simulation offers a broad range of methods for attempting to reduce variance. The effectiveness of these methods varies widely across applications. In practice, the most commonly used methods are the simplest ones, particularly antithetic variates and control variates. These can be very effective in some cases, and provide almost no benefit in others. Some of the most powerful methods — importance sampling is a good example — get much less use, in part because they are more difficult to work with, but also because if used improperly they can give disastrous results. This situation can leave users of Monte Carlo in a quandary, not knowing what methods to use in what settings.

A premise of our work is that the greatest gains from the use of variance-reduction techniques rely on exploiting the special structure of a problem or model. The identification of special structure should be automated to the extent possible so that each application does not require a separate investigation. This perspective is particularly relevant to importance sampling, which seeks to improve precision by focusing simulation effort on the most important regions of the space from which samples are drawn. Which regions are most important depends critically on the underlying model and also on the form of the payoff of the particular security to be priced. The use of importance sampling thus requires adaptation to each payoff and each model; for this to be feasible in practice requires that the adaptation be automated.

In Glasserman, Heidelberger, and Shahabuddin [1999] (henceforth GHS), we propose and analyze a variance-reduction technique that combines importance sampling (based on a change of drift in the underlying stochastic processes) with stratified sampling. As general methods for variance reduction, these are both reasonably standard; the innovation in GHS [1999] lies in the approach used to select the change of drift and the directions along which to stratify.

In the most general version of the method, the new drift is computed by solving an optimization problem, and the stratification direction solves an eigenvector problem. The calculation of these quantities is a "preprocessing" step of the method (executed before any paths are simulated) that systematically identifies structure to be exploited by the variance-reduction methods.

In GHS we give a theoretical analysis of this method based, in part, on scaling the randomness in the underlying model by a parameter ε and investigating asymptotics as $\varepsilon \to 0$. Asymptotic optimality properties of the method are established in GHS. We tested the method on three types of examples: Asian options, a stochastic volatility model, and the Cox-Ingersoll-Ross short rate model. Our numerical results indicate substantial potential for variance reduction using the method.

The purpose of this article is to develop and investigate the use of the method in a much more ambitious class of models — multifactor models of the entire term structure of interest rates of the Heath-Jarrow-Morton [1992] type. Using, for example, quarterly rates with a twenty-year horizon makes the state vector for such a model eighty-dimensional. Simulating this vector for m steps in a d-factor model corresponds to sampling in an md-dimensional space, easily making the dimension very large.

The preprocessing phase of our method selects a drift for the model — tailored to the factor structure and the payoff of the instrument to be priced — that drives the evolution of the forward curve to the most important region. "Importance" is measured by the product of the discounted payoff from a path and the probability density of the path. To further reduce variance, we stratify linear combinations of the input random variables, the choice of linear combination also tailored to the factor structure and the payoff. The complexity of the HJM setting necessitates some simplification in the implementation of the preprocessing calculations, so in addition to investigating the use of the basic method from GHS, we propose and evaluate some approximations.

We test the methods in pricing caps and swaptions (important for fast calibration), yield spread options, and flex caps. Our numerical results support the viability of the method in the HJM setting.

I. OVERVIEW OF THE METHOD

The general simulation method in GHS [1999], based on combining importance sampling and stratified sampling, applies to the estimation of an expression of the form E[G(Z)] where Z has the standard n-dimensional normal distribution, and G takes non-negative values. G is the discounted payoff of a derivative security, and Z the vector of stochastic inputs to the simulation.

The Setting

Although it is not required for the method, it is useful to frame the general setting as one of simulating a vector diffusion process of the form

$$dX_t = \mu(X_t, t)dt + \sigma(X_t, t)dW_t \qquad (1)$$

where X_t and $\mu(X_t, t)$ are k-vectors; $\sigma(X_t, t)$ is a $k \times d$ matrix; and W_t is a d-dimensional standard Brownian motion. Processes of this form are commonly simulated through a discrete-time approximation (an Euler scheme) of the form

$$\hat{X}_{i+1} = \hat{X}_i + \hat{\mu}(\hat{X}_i, i)\Delta t + \hat{\sigma}(\hat{X}_i, i)\sqrt{\Delta t}Z_{i+1} \qquad (2)$$

where $\hat{\mu}$ and $\hat{\sigma}$ are discrete approximations to the continuous coefficients; Δt is the simulation time step; and

$$Z_i = \begin{pmatrix} Z_i(1) \\ Z_i(2) \\ \vdots \\ Z_i(d) \end{pmatrix} \sim N(0, I_d), \qquad i = 1, 2, \ldots,$$

are independent standard multivariate normal random vectors.[1]

Interpret $Z_i(j)$ as the increment in the j-th underlying factor at the i-th step. Simulating such a process for m steps requires a total of md independent normal random variables, and by stacking the vectors $Z_1, \ldots, Z_m$ we may view the simulated path of $\hat{X}$ as a (deterministic) function of a single md-dimensional normal random vector Z.

Letting n = md, the density of this n-dimensional normal vector is given by[2]

$$\phi_n(z) = (2\pi)^{-n/2} \exp(-\frac{1}{2}z'z), \qquad z \in R^n$$

Expectations of functions of $\hat{X}$ may be viewed as integrals with respect to this density.

More specifically, suppose (1) and (2) give the dynamics of all relevant state variables, including the value of whatever asset is chosen as numeraire, under the martingale measure associated with the chosen numeraire. (Later, we specialize to the case where X_t records the forward curve at time t, the numeraire is the money market account, and the martingale measure is the usual risk-neutral measure.)

By including enough information in the state vector X_t, we can ordinarily make the discounted payoff of a derivative security a deterministic function of the path of X_t (or its approximation $\hat{X}$ in a simulation). The price of the derivative security is the expectation of this discounted payoff. But the path of $\hat{X}$ is itself some deterministic function of the n-dimensional random vector Z, so by letting G denote the composition of these two functions we may denote by G(Z) the discounted payoff of the derivative security associated with input Z. The price of the derivative is

$$\alpha = E[G(Z)] = \int_{R^n} G(z)\phi_n(z)dz \qquad (3)$$

Pricing by Monte Carlo may be viewed as a way of estimating such an integral.

Importance Sampling

A standard Monte Carlo estimator of (3) draws independent samples $Z^{(1)}, \ldots, Z^{(k)}$ from $N(0, I_n)$; evaluates the discounted payoff $G(Z^{(i)})$ resulting from each; and averages to arrive at the estimator

$$\frac{1}{k}\sum_{i=1}^{k} G(Z^{(i)})$$

Importance sampling is based on the observation that from (3) we may write

$$\alpha = \int_{R^n} G(z)\frac{\phi_n(z)}{\psi(z)}\psi(z)dz \qquad (4)$$

for any probability density ψ that is positive throughout $\mathbf{R}^n$. This representation suggests an alternative estimator in which $Z^{(1)}, ..., Z^{(k)}$ are independently sampled from ψ and then combined in the estimator:

$$\hat{\alpha}_{\psi}(k) = \frac{1}{k}\sum_{i=1}^{k} G(Z^{(i)})\frac{\phi_n(Z^{(i)})}{\psi(Z^{(i)})}$$

Sampling from ψ rather than ϕ_n results in oversampling some regions and undersampling others. Weighting each $G(Z^{(i)})$ by the *likelihood ratio* $\phi_n(Z^{(i)})/\psi(Z^{(i)})$ ensures that the expected value of the resulting estimator is unchanged — in particular, $\hat{\alpha}_{\psi}(k) \to \alpha$ by virtue of (4) and the law of large numbers.

Different choices of ψ will result in estimators with different variances. In general, there is no guarantee that sampling from ψ rather than ϕ_n will actually reduce variance, but the potential for variance reduction through importance sampling is enormous. Indeed, if G is non-negative, and if we choose ψ proportional to the product of G and ϕ_n, (i.e., $\psi(z) = G(z)\phi_n(z)/c$, for the constant c that makes ψ integrate to 1), then importance sampling yields a zero-variance estimator; each replication $G(Z^{(i)})\phi_n(Z^{(i)})/\psi(Z^{(i)})$ equals the constant c.

The catch, of course, is that the normalization constant c is the unknown quantity α, so this method is not viable in practice. Nevertheless, the observation is useful as a guide to selecting importance sampling densities. An effective choice of ψ should weight each point z roughly in proportion to the product of its discounted payoff G(z) and its original probability $\phi_n(z)$.

The importance sampling method developed in GHS [1999] restricts ψ to be a multivariate normal density $N(\mu, I_n)$ for some $\mu \in \mathbf{R}^n$; i.e., $\psi(z) = \phi_n(z - \mu)$. For any choice of μ, the likelihood ratio under this change of measure is particularly simple, reducing to

$$\frac{\phi_n(z)}{\phi_n(z-\mu)} = \exp(-\mu' z + \frac{1}{2}\mu'\mu) \tag{5}$$

In GHS, μ is chosen to be z^* where z^* solves the optimization problem

$$\max_{z \in R^n}\{G(z)\phi_n(z)\} \tag{6}$$

or, equivalently:

$$\max_{z \in R^n}\{F(z) - \frac{1}{2}z'z\}$$

where F(z) = log G(z) (taking log 0 = –°).

Perhaps the simplest interpretation of this approach is that it approximates the optimal (zero-variance) density $G(z)\phi_n(z)/\alpha$ by a normal density whose mode coincides with that of the optimal density [which occurs at the solution to (6)], and whose covariance matrix coincides with that of the original normal density.[3]

There is further motivation for this importance sampling strategy. For any choice of μ, the resulting estimator is the average of independent copies of

$$G(Z)\exp(-\mu' Z + \frac{1}{2}\mu'\mu), \qquad Z \sim N(\mu, I_n) \tag{7}$$

That is, of

$$\exp(F(\mu + Z) - \mu' Z - \frac{1}{2}\mu'\mu), \qquad Z \sim N(0, I_n) \tag{8}$$

Under conditions in GHS, the μ found by solving (7) will satisfy the first-order conditions $\nabla F(\mu) = \mu'$, with ∇F denoting the gradient of F. If F is approximately linear near μ, then $F(\mu + Z)$ in (8) is approximately $F(\mu) + \nabla F(\mu)Z$. Making this substitution and then using the first-order conditions to replace $\nabla F(\mu)Z$ with $\mu' Z$, (8) reduces to

$$\exp(F(\mu + Z) - \mu' Z - \frac{1}{2}\mu'\mu)$$

$$\approx \exp(F(\mu) + \nabla F(\mu)Z - \mu' Z - \frac{1}{2}\mu'\mu)$$

$$= \exp(F(\mu) - \frac{1}{2}\mu'\mu)$$

which is constant and thus has zero variance.

This indicates that if the log discounted payoff were linear, this choice of importance sampling density would eliminate all variance; more generally, if F is close to linear, this choice of density can be expected to eliminate much of the variance in the original estimator. This gives an alternative interpretation of the method above for choosing μ. (See GHS [1999] for a more extensive analysis and discussion. See Boyle, Broadie, and Glasserman [1997], Newton [1997], and Schoenmakers and Heemink [1997] for other approaches to importance sampling in option pricing.)

Stratified Sampling

Although importance sampling by itself can, in some cases, yield substantial variance reduction (particularly in pricing options for which the probability of a positive payoff is small), the power of the method in GHS comes from the combination of importance sampling with stratified sampling along carefully selected directions. We describe this combination after providing some background on stratification. (For additional background, see, e.g., Fishman [1996] or Hammersley and Handscomb [1964].)

In stratified sampling, one draws samples from a distribution while ensuring that the fraction of samples falling in each of a collection of prespecified sets — the strata — matches the theoretical probability of that set. For example, in sampling from the one-dimensional standard normal distribution, one might choose as strata the positive and negative real half lines. Each of these has probability 1/2 under the standard normal. If we draw one hundred independent samples from the normal distribution, it is unlikely that exactly fifty will be positive and fifty negative. Stratified sampling refers to any mechanism that, in this example, ensures that indeed half of the samples are positive and half are negative.

In general, such stratified sampling ensures a more regular sampling pattern and therefore reduces variance. The amount of variance reduction obtained depends on how the strata are defined; if the variability of the output within each stratum is small, then large variance reductions are obtained. A contribution of this work is to describe stratification schemes that often result in large variance reductions.

In our use of stratified sampling, we partition the real line into M strata, and sample from these in the correct proportions. We choose M intervals, each having probability 1/M, and then draw one value from each interval with the conditional distribution on that interval. We do this by first partitioning the unit interval (0, 1) into M subintervals of length 1/M, and sampling uniformly from each subinterval. We then apply the inverse cumulative normal distribution to each of these M values in (0, 1). The resulting M values in (–° , °) constitute a stratified sample from the normal distribution.

To put this more precisely, let $U^{(1)}$, ..., $U^{(M)}$ be independent and uniformly distributed on (0, 1). If we were to set $X^{(i)} = \boldsymbol{\Phi}^{-1}(U^{(i)})$, i = 1, ..., M, with

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt$$

the cumulative normal, then $X^{(1)}$, ..., $X^{(M)}$ would be distributed as independent draws from the normal distribution.[4] To generate a stratified sample, we first set

$$V^{(i)} = \frac{i-1}{M} + \frac{U^{(i)}}{M}, \qquad i = 1, \ldots, M$$

so that $V^{(i)}$ lies between (i – 1)/M and i/M, and is uniformly distributed over this interval. Thus, $V^{(i)}$ has the distribution of a uniform random variable on (0, 1) conditioned to fall in the stratum [(i – 1)/M, i/M].

Now set $X^{(i)} = \boldsymbol{\Phi}^{-1}(V^{(i)})$, i = 1, ..., M. Each $X^{(i)}$ lies between the (i – 1)/M-th and i/M-th fractiles of the normal distribution (because $V^{(i)}$ lies between the corresponding fractiles of the uniform distribution), and has the distribution of a normal random variable conditioned to lie in this range (because $V^{(i)}$ has the corresponding conditional distribution for a uniform random variable).

In sampling from the multivariate normal distribution $N(0, I_n)$, we apply this technique by stratifying along one direction in n dimensions.[5] This is nearly the same as performing a numerical (rather than Monte Carlo) integration along the stratified dimension and using Monte Carlo for the other dimensions. We are in fact free to choose any direction in $\mathbf{R}^n$ as the direction along which to stratify. A direction is described by a vector $u \in \mathbf{R}^n$ normalized to have unit length (i.e., $u'u = 1$). If $Z \sim N(0, I_n)$, then $u'Z \sim N(0, 1)$ because of the normalization.

Using the one-dimensional algorithm, we may stratify $u'Z$ into M strata, and then — conditional on each of the M outcomes of u´Z — sample the full vector Z. This is facilitated by the fact that the conditional distribution is itself normal:

$$(Z \mid u'Z = a) \sim N(ua, I_n - uu')$$

The precise steps are as follows:

- Generate a stratified sample $X^{(1)}, \ldots, X^{(M)}$ from N(0, 1) as described above; interpret $X^{(i)}$ as the i-th value of u´Z.
- Draw $Y^{(i)}$ from $N(0, I_n)$, i = 1, ..., M, independent of each other and of the $X^{(i)}$.
- Set $Z^{(i)} = uX^{(i)} + C_u Y^{(i)}$, i = 1, ..., M, where C_u is any n × n matrix satisfying $C_u C_u' = I_n - uu'$; the choice $C_u = I_n - uu'$ is particularly convenient because it makes evaluation of $C_u Y^{(i)}$ an O(n) operation rather than $O(n^2)$.

The resulting $Z^{(1)}, \ldots, Z^{(M)}$ constitute a stratified sample from $N(0, I_n)$, stratified along direction u in the sense that the projected values $u'Z^{(1)}, \ldots, u'Z^{(M)}$ form a stratified sample from N(0, 1).

It remains to specify the choice of direction u. In GHS [1999], we propose and analyze two strategies for doing this. The simpler method sets u = μ (= z*), the optimal vector found for importance sampling. The other method finds the best direction for a quadratic approximation to F = log G, the logarithm of the discounted payoff. (Recall that our importance sampling method can be interpreted as eliminating the linear part of F.) In GHS [1999] we prove that if F(z) = z´Az for some (symmetric) matrix A with eigenvectors $v_1, \ldots, v_n$ and associated eigenvalues $\lambda_1, \ldots, \lambda_n$, ordered so that

$$\left(\frac{\lambda_1}{1-\lambda_1}\right)^2 \geq \left(\frac{\lambda_2}{1-\lambda_2}\right)^2 \geq \cdots \geq \left(\frac{\lambda_n}{1-\lambda_n}\right)^2 \qquad (9)$$

then v_1 is an optimal direction for stratification. This suggests that for general (twice-differentiable) F, a good direction may be found by applying this criterion to the Hessian (the matrix of second derivatives) of F. The Hessian should be evaluated at the point z*(= μ) found in the optimization because the importance sampling has recentered the distribution at this point. Interestingly, we find that the optimal eigenvector is itself often similar to the optimal path z*.

Regardless of how the direction u is chosen, this method produces a stratified sample $Z^{(1)}, \ldots, Z^{(M)}$ from $N(0, I_n)$, stratified along direction u. To combine this with importance sampling, we then add the drift vector μ to each $Z^{(i)}$, resulting in a stratified sample from $N(\mu, I_n)$. We evaluate the discounted payoffs $G(Z^{(i)})$; weight each one by the corresponding likelihood ratio from (5); and average to get

$$\hat{G} = \frac{1}{M}\sum_{i=1}^{M} G(Z^{(i)}) \exp(-\mu' Z^{(i)} + \frac{1}{2}\mu'\mu)$$

It should be noted that the M values averaged in this expression are not independent (because of the stratification), so their standard deviation is not a relevant measure of the sampling variability in $\hat{G}$. To supplement the point estimator with a confidence interval, we replicate the procedure above k times to produce the k estimates $\hat{G}_1, \ldots, \hat{G}_k$ (each based on a stratified sample of size M). The final estimator is the average $\bar{G}$ of the $\hat{G}_i$; its standard error is approximately the sample standard deviation S of $\hat{G}_1, \ldots, \hat{G}_k$ divided by $\sqrt{k}$. We thus arrive at, e.g., an approximate 95% confidence interval of the form

$$\bar{G} \pm 1.96 \frac{S}{\sqrt{k}}$$

since 1 – Φ(1.96) = 0.025. This procedure involves simulating a total of kM paths. Given a fixed budget for the total number of paths, increasing M generally increases the precision of the point estimator, while larger k improves the estimate of the standard error and the validity of the normal approximation implicit in the confidence interval.

A concise summary of the algorithm is given in the appendix.

II. THE HEATH-JARROW-MORTON SETTING

Our objective is to investigate the application of the general method in GHS to high-dimensional, multifactor models. Because of its broad applicability and widespread use, the Heath-Jarrow-Morton [1992] framework provides a particularly appropriate setting for this investigation.

Simulation Model

We first review the continuous-time HJM formulation, and then proceed to the discretized version used in a simulation. Let $f(t, \tau)$ be the instantaneous continuously compounded forward rate for time τ as of time t, with $0 \,''\, t \,''\, \tau \,''\, T^*$ for some ultimate maturity T^*. Let $B(t, \tau)$ be the time-t price of a riskless bond paying \$1 at time τ. Then

$$f(t,\tau) = -\frac{\partial}{\partial\tau}\log B(t,\tau)$$

and

$$B(t,\tau) = \exp\left(-\int_t^T f(s,u)du\right)$$

The instantaneous short rate at time t is $r(t) \equiv f(t, t)$.

In its usual formulation, the HJM framework specifies the evolution of the forward curve $[f(t, \tau), t \,''\, \tau \,''\, T^*]$ over the time interval $0 \,''\, t \,''\, T^*$. For a d-factor model, take W_t to be a d-dimensional standard Brownian motion. The arbitrage-free dynamics of the forward curve under the risk-neutral measure have the form

$$df(t, \tau) = (\sigma(t, \tau)' \int_t^\tau \sigma(t, u)du)dt + \sigma(t, \tau)'dW_t \qquad (10)$$

where $\sigma(t, \tau)$ is a d-vector for each t and τ. Under technical conditions detailed in Heath, Jarrow, and Morton [1992], the drift specified in (10) ensures that discounted bond prices

$$\exp\left(-\int_0^t r(u)du\right)B(t,\tau) \qquad (11)$$

(i.e., bond prices divided by the value of the money market account) are martingales (in t), which is the key condition for the absence of arbitrage.

Implementation of a general HJM model requires discretization in both calendar time (the first argument of f) and maturity (the second argument). For simplicity, we assume that both arguments are discretized in multiples of a fixed, common time increment Δt. We write $F(i, j)$ for the continuously compounded forward rate for the interval $[j\Delta t, (j + 1)\Delta t]$ contracted at time $i\Delta t$:

$$F(i,j) = \frac{1}{\Delta t}\int_{j\Delta t}^{(j+1)\Delta t} f(i\Delta t,u)du$$

A simulation algorithm for the forward curve, based on discretizing (10), takes the form

$$F(i+1,j) = F(i,j) + a_{ij}\Delta t + \sqrt{\Delta t}\sum_{k=1}^{d} s_{ij}(k)Z_{i+1}(k),$$

$$j > i,\ i = 0, 1, 2, \ldots, \qquad (12)$$

where the discrete drift a_{ij} is chosen to keep the model arbitrage-free; s_{ij} is a discrete analogue of the volatility $\sigma(t, \tau)$; and $Z_1, Z_2, \ldots$, are independent $N(0, I_d)$ vectors. Specifically, we take

$$a_{ij} = \frac{\Delta t}{2}\sum_{k=1}^{d}\left[\left(\sum_{\ell=i}^{j} s_{i\ell}(k)\right)^2 - \left(\sum_{\ell=i}^{j-1} s_{i\ell}(k)\right)^2\right]$$

This choice ensures that the discrete discounted bond prices $D_iB(i, j)$ [the discrete analogues of (11)] are martingales in i, and thus keeps the model arbitrage-free after discretization.[6] Here, $F(\ell, \ell)$ is the (Δt) short rate at time $\ell\Delta t$; the bond price is given by

$$B(i,j) = \exp\left(-\sum_{\ell=i}^{j-1} F(i,\ell)\Delta t\right) \qquad (13)$$

and the discount factor is

$$D_i = \exp\left(-\sum_{\ell=0}^{i-1} F(\ell,\ell)\Delta t\right) \qquad (14)$$

Observe that simulating a single step in (12) uses d samples from the standard normal distribution, and simulating m steps thus requires n = md standard normals. Pricing a derivative security that requires m simulation steps may thus be viewed as computing an integral with respect to the n-dimensional standard normal distribution.

In our numerical examples we use d = 3 factors. Our choice of factors is fairly ad hoc; the objective is to use an example in which three factors suffice, but fewer than three would not. It is convenient to separate the specification of the discrete volatility s_{ij} in (12) into terms representing the overall level of volatility and terms capturing the correlation structure across points of the forward curve. We use a specification of the form

$$[s_{ij}(1), s_{ij}(2), s_{ij}(3)] = \\ F(i, j)\sigma(j - i)[g_1(j - i), g_2(j - i), g_3(j - i)]$$

The factor loadings g_1, g_2, g_3 are normalized so that

$$g_1(j)^2 + g_2(j)^2 + g_3(j)^2 \equiv 1, \qquad j = 1, 2, \ldots, \qquad (15)$$

With this normalization, $\sigma(j - i)$ becomes the overall level of (proportional) volatility of the j-th forward $F(\bullet, j)$ at the i-th step.[7] The correlation between changes in the j-th and k-th forwards over the i-th step is determined by

$$g_1(j - i)g_1(k - i) + g_2(j - i)g_2(k - i) + g_2(j - i)g_2(k - i)$$

the inner product between the factor loadings at time-to-maturity $j - i$ and time-to-maturity $k - i$.

In the numerical examples, we take $\Delta t = 0.25$ years, a typical discretization used in practice. We consider maturities of up to twenty years, so our initial forward curve is eighty-dimensional. We initialize it by taking

$$F(0, j) = \log(150 + 12j)/100, \qquad j = 0, 2, \ldots, 79$$

producing an upward-sloping curve increasing gradually from 5% to 7%.

For the overall level of volatility as a function of time to maturity, we specify

$$\sigma(j) = 0.12 + (j/81)[1 - (j/81)]^4, \qquad j = 0, 1, \ldots, 79$$

This produces a humped term structure of volatility starting at 13.17%, increasing to a maximum of 20.19% at a maturity of four years, and decreasing gradually to 12% at the end of twenty years.

Finally, for the factors g_1, g_2, g_3 we proceed as follows. We form the 80 × 80 symmetric matrix with entries

$$\exp(-0.0004(i - j)^2), \qquad i, j = 1, \ldots, 80$$

and find orthonormal eigenvectors $x_1, \ldots, x_{80}$ and associated eigenvalues $\beta_1 \geq \beta_2 \ldots \geq \beta_{80}$. With these parameters we have

$$\frac{\beta_1}{\Sigma_i\beta_i} = 72.8\%, \quad \frac{\beta_1 + \beta_2}{\Sigma_i\beta_i} = 95.6\%, \quad \frac{\beta_1 + \beta_2 + \beta_3}{\Sigma_i\beta_i} = 99.5\%$$

suggestive of a model in which three factors suffice but one factor definitely does not.

We complete the specification of the factors by setting

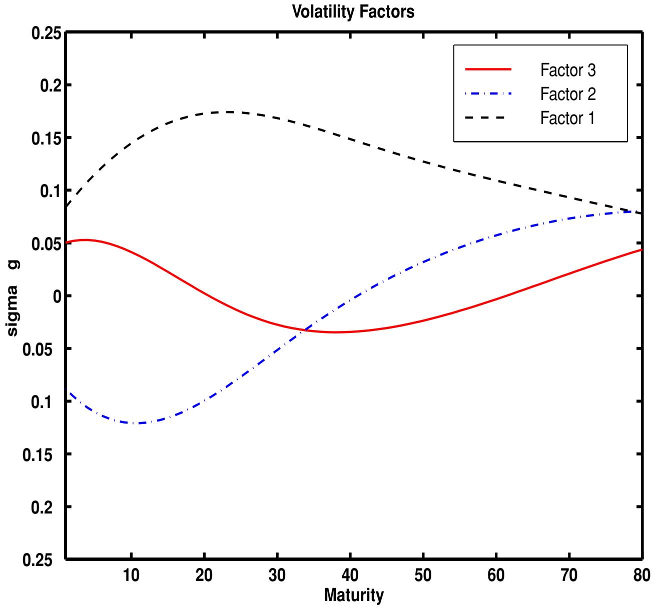
$$g_i(j) = \frac{\sqrt{\beta_i}\, x_i(j)}{\sqrt{\beta_1 x_1^2(j) + \beta_2 x_2^2(j) + \beta_3 x_3^2(j)}},$$

$$j = 1, \ldots, 80, \quad i = 1, 2, 3$$

to enforce the normalization (15).

Exhibit 1 graphs $\sigma(j)g_i(j)$ (the volatility contributed by factor j) as a function of time to maturity j for i = 1, 2, 3. The first factor has constant sign across maturities and thus moves all forward rates in the same direction; the second factor moves the near and the far ends of the forward curves in oppo-

EXHIBIT 1
Factors for Numerical Examples



site directions; the third factor bends the forward curve by moving the ends in the opposite direction of the middle.

Applying the Variance-Reduction Method

An example may help to fix ideas and to motivate the approximations. Consider the pricing of a *caplet* — a single-period interest rate cap. Suppose the caplet applies to the interval from $N\Delta t$ to $(N + 1)\Delta t$. Ordinarily, caplets are written on simple (rather than continuously compounded) rates with settlement at the end of the period. Thus, with a strike of K the caplet pays

$$C_{N+1} = \max(0, (e^{\Delta t\, F(N,N)} - 1) - K\Delta t)100 \qquad (16)$$

at time $(N + 1)\Delta t$ on a notional amount of 100. Its price at time 0 is given by the expected present value

$$E[D_{N+1}C_{N+1}] \qquad (17)$$

where the discount factor D_{N+1} is defined in (14). Through (12) we may view the discounted payoff as a function of Nd standard normal random variables, with d the number of factors — three in our examples. Write G(Z) for this discounted payoff.

It is useful to encode the 3N-dimensional input vector Z as

$$Z_1(1), Z_2(1), ..., Z_N(1), Z_1(2), Z_2(2), ..., Z_N(2), Z_1(3), Z_2(3), ..., Z_N(3)$$

Comparison with (12) indicates that the first N components determine the path of the first factor, the second N components the path of the second factor, and the last N components the path of the third factor. (This convention is important in interpreting some of our graphs.)

The first step in implementing the algorithm solves the optimization problem (6) over all input vectors z. We may interpret this as solving for the paths of the underlying factors that maximize the product of the discounted payoff G(z) and the density element $\exp(-z'z/2)$. In the case of a cap consisting of multiple caplets, the relevant G is the sum of the discounted payoffs of the individual caplets, and thus a sum of terms of the type inside the expectation in (17).

The result of the optimization is displayed in Exhibit 2 for a cap making quarterly payments with a strike of 7% that has an initial payment at time 0.25 years and a final payment at time 5.0 years. The graph may be read as follows. On the optimal path, the increments of the first factor start at a large positive

Exhibit 2
Optimal Paths of Factors for Interest Rate Cap

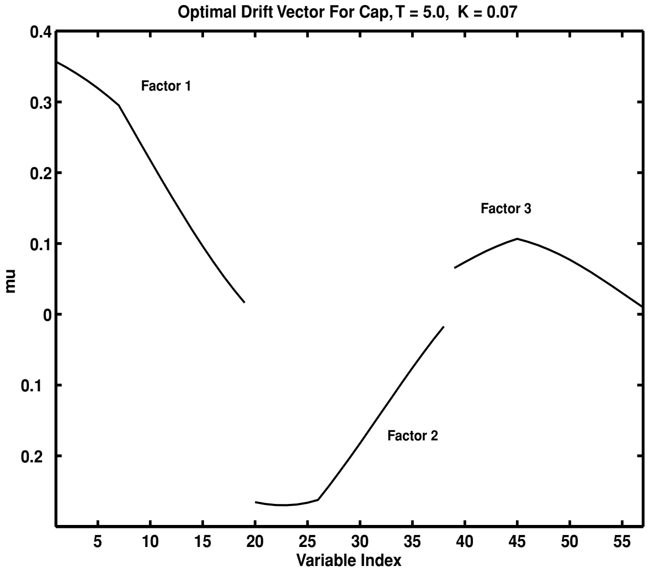
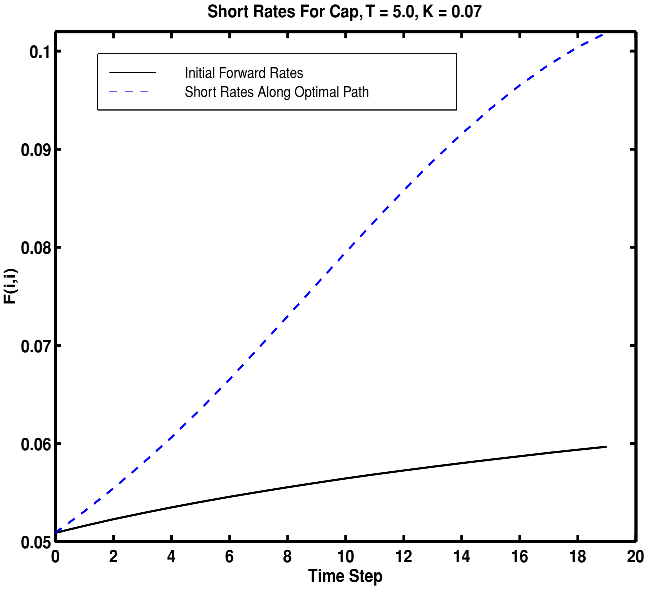


Exhibit 3
Optimal Path of Short Rate



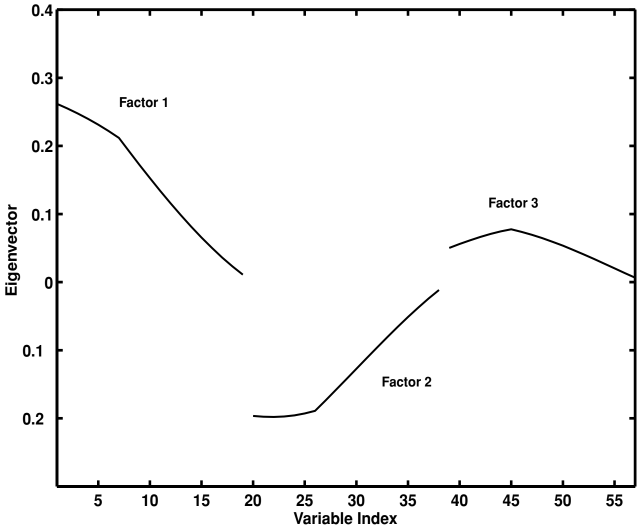
value (driving this factor upward quickly) and then decrease; the increments of the second factor are roughly flat (and negative, thus driving the path of this factor downward) and then increase to zero; and the increments of the third factor increase until just before the midpoint of the cap and then decrease. In our importance sampling method, these optimal paths become the drifts added to the factors — in effect, importance sampling centers the evolution of the underlying Brownian motion around the optimal path rather than around 0.

The combined effect of the optimal factor paths (and thus of the new drift) becomes somewhat clearer in Exhibit 3 showing the evolution of the short rate F(i, i) determined by the optimal paths — i.e., when the driving increments Z_i in (12) are evaluated along the optimal paths rather than sampled randomly. We see that the impact of importance sampling in pricing a cap is to drive the short rate upward much more quickly than would be the case without importance sampling. (Without importance sampling, the evolution of the short rate would be centered roughly around the initial forward curve, also shown in the graph.)

The next step in the procedure finds a good direction for stratified sampling either by using the optimal μ found for importance sampling, or else by choosing one of the eigenvectors of the Hessian of log $G(\mu)$. The best eigenvector according to the criterion in (9), which we denote by v_1, is shown in Exhibit 4. In this particular case, μ and v_1 are very similar in shape; indeed if μ is renormalized to have unit norm like v_1, then $||\mu - v_1||$ is only 0.022. Thus the effectiveness of stratifying on v_1 should be about the same as that of stratifying on μ. Having selected a new drift vector μ for importance sampling and a good direction u for stratification (either μ or v_1), we can proceed with the simulation.

EXHIBIT 4

Best Eigenvector for Stratification for Interest Rate Cap



This example serves to underscore several aspects of the proposed method. First, in the complexity of the HJM setting, one cannot reasonably hope to simply guess at a good change of drift for importance sampling — it is by no means obvious in advance that the optimal change of drift would look anything like Exhibit 2, even if we could guess that the optimal short rate path should look something like Exhibit 3. Nor is it obvious that a good direction for stratification would look like Exhibit 4. Yet we will see in our numerical results that these choices lead to very substantial variance reduction.

This indicates the value of an automated and systematic approach to identifying special structure. At the same time, it must be acknowledged that solving the optimization and eigenvector problems in the preprocessing phase of our method can impose an additional computational burden, particularly in a high-dimensional multifactor model.

III. APPROXIMATIONS

Approximate Optimization

We can reduce the major overheads involved in applying the procedure — the overhead to solve the optimization problem, and the overhead to compute the eigenvectors) for stratification — using an approach outlined in GHS [1999]. In Exhibit 2, if for a moment we forget that the indexes are discrete, it appears that the component values of the optimal drift vector for each factor vary continuously with the index. This suggests approximating the optimal drift vector by a continuous function parameterized by a small number of variables, and then optimizing over those variables to find an approximately optimal drift vector. Since numerical optimization routines often use finite difference approximations for gradients, this approach will reduce the number of paths (i.e., function calls of G) during the optimization.

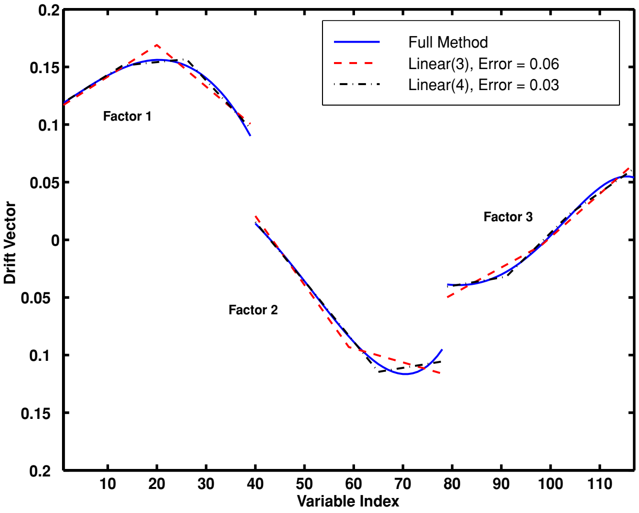
An attractive version of this general approach is to approximate the optimal drift vector by a piecewise linear function. In the piecewise linear approximation, the index points where the slope of the line changes are termed *knot points*. One can approximately represent any drift vector by the component values at the knot points and linearly interpolated values in between. One can then optimize over all the possible component values at the knot points.

In most cases we have tried, this yields a good approximation to the optimal drift vector. We choose equally spaced knot points here, although this is not generally required. If in some problem there are d = 3 factors, each with N = 80 time intervals, then choosing four knot points for each factor reduces the dimension of the optimization problem from 240 to 12.

In general, we define an appropriate mapping h(·) from a lower-dimensional space $\mathbf{R}^k$ (with $k < n$) to $\mathbf{R}^n$ such that most of the variation of $G(z)\phi_n(z)$ with respect to z in $\mathbf{R}^n$ is captured by the variation of $G[h(\bar{z})]\phi_n[h(\bar{z})]$ with respect to $\bar{z}$ in $\mathbf{R}^k$. One then maximizes $G[h(\bar{z})]\phi_n(h(\bar{z})]$ with respect to $\bar{z}$ in $\mathbf{R}^k$. If $\bar{\mu}$ is the optimal point in the new space, then one transforms back to the original space, using the transformation $\mu = h(\bar{\mu})$. The linear interpolation method comes under the class of mappings h(·) that are linear transformations from $\mathbf{R}^k$ to $\mathbf{R}^n$, i.e., $z = \mathbf{M}\bar{z}$ where $\mathbf{M}$ is an $n \times k$ matrix.[8]

EXHIBIT 5

Optimal and Approximate Optimal Drift Vectors for Caplet — T = 10.0 and K = 0.07



The error listed is the norm of the difference between the optimal and approximately optimal drift vectors.

Exhibit 5 gives the optimal drift and the approximate optimal drift using linear interpolation for the caplet example, with T = 10. If each factor uses m knot points, we call the method "linear(m)." For example, linear(3) would use three variables per factor: the first, middle, and last increments. If all the variables are used, we call the method "full." The legend in Exhibit 5 gives the norm of the difference between the optimal drift vector and the approximately optimal drift vector.

An alternative type of a linear transformation not based on linear interpolation of paths uses the subspace defined by the principal components of the Brownian motion associated with each factor, corresponding to a few of the largest eigenvalues. Yet another class of h(·) corresponds to representing components of the drift vector z (corresponding to each factor) by low-order polynomials (with variable coefficients) in the indexes. One then optimizes over the coefficients of the polynomial.

Approximate Eigenvector Calculation

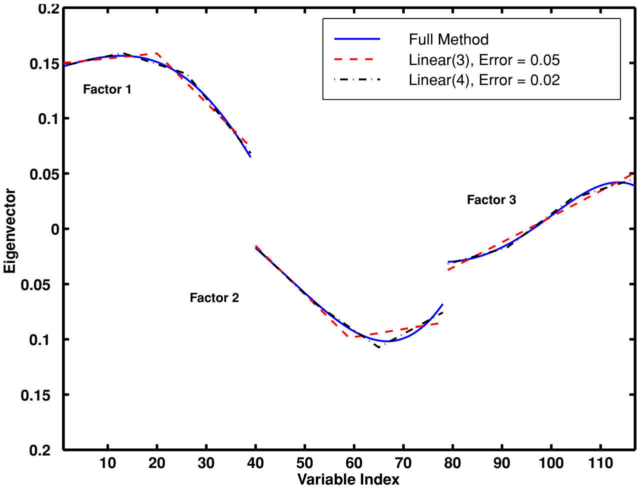
Computing the Hessian H of F(z) = log[G(z)] at any z and the best eigenvector (which we denote by v_1) is usually an $O(n^2)$ operation. There is an approach to approximate v_1 in a k-dimensional subspace, $k < n$ (and then transform back to the n-dimensional subspace). We start by choosing an $n \times m$ matrix $\mathbf{M}$ whose columns seem likely to span a good approximation to v_1. For example, if we believe v_1 should be approximately piecewise linear, we could choose $\mathbf{M}$ to build an n-vector from an m-vector by linear interpolation, as we did for the optimal drift.

Let $\bar{z}$ denote an element of $\mathbf{R}^k$ and z an element of $\mathbf{R}^n$. The function $F_M(\bar{z}) = F(\mathbf{M}\bar{z})$ has Hessian $H_M = \mathbf{M}'H\mathbf{M}$, where H is the Hessian of F at $\mathbf{M}\bar{z}$. Because H_M is $k \times k$, it may be much less costly to evaluate (through finite differences of F_M) than H.

The next step is to find the best eigenvalue γ_1 and the best eigenvector $\bar{v}_1$ of the $k \times k$ matrix $(\mathbf{M}'\mathbf{M})^{-1}\mathbf{M}'H\mathbf{M}$, according to the criterion in (9). Our candidate approximation for v_1 is then $\mathbf{M}\bar{v}_1$. This procedure is exact if F happens to depend on z only through $\mathbf{M}'z$. Moreover, any eigenvector of H that lies

EXHIBIT 6

Eigenvector and Approximate Eigenvector for Caplet n T = 10.0 and K = 0.07



The error listed is the norm of the difference between the eigenvector and approximate eigenvector.

in the range of **M** is recovered by this procedure along with its eigenvalue.[9]

Exhibit 6 displays the best eigenvector and approximate best eigenvector for the caplet problem with T = 10; the legend gives the errors. Again we see that the best eigenvector and the approximate best eigenvector are quite close. Note also how similar the shape of the optimal drift vector is to the best eigenvector.

IV. NUMERICAL RESULTS

Our primary measure of the effectiveness of the method is the estimated variance ratio, defined to be the estimated variance using standard simulation divided by the estimated variance using a variance-reduction technique. This ratio gives an indication of the statistical acceleration of a method, i.e., the (potential) factor by which the number of samples can be reduced by applying the variance-reduction technique.

Suppose the per sample variance using standard simulation is σ^2, while that using a variance-reduction technique is σ_1^2. If standard simulation is run for k replications, the resulting variance is σ^2/k. Since a simulation of k_1 replications using the variance-reduction technique results in a variance of σ_1^2/k_1, to achieve the same variance implies setting $k_1 = (\sigma_1^2/\sigma^2)$ k. Thus, to achieve the same variance, standard simulation requires a factor of $k/k_1 = (\sigma^2/\sigma_1^2)$ times as many replications as the variance reduction technique.

Whether this potential savings in run length can actually be achieved in practice depends on many factors, including the desired accuracy of the resulting estimate. The acceleration factor does not include the setup cost (optimization overhead) of the method or the additional per sample cost of the stratification, which is quite small — typically less than 5%.

To achieve reasonably accurate estimates of the variance ratio, all our results are based on a total of 50,000 replications (paths) per method. We also include results for a straightforward and more commonly applied variance-reduction technique, antithetic sampling (see Hammersley and Handscomb [1964]). In the case of antithetics, results are based on 25,000 independent antithetic pairs, representing a total of 50,000 paths. For methods using stratification, 100 strata are used (M = 100 in step 2 in the appendix and k = 500).

We first consider options in which the payoff function is a *continuous* function of the underlying Gaussian increments Z. Examples of such payoffs include caps, swaptions, and a European yield spread option (specifically, an option on the difference between the yields on zero-coupon bonds of different maturities). We next consider options in which the payoff is not a continuous function of the Gaussian increments. We discuss in detail discontinuities associ-

EXHIBIT 7

Estimated Variance Ratios for Caplets in Three-Factor HJM Model

T	K	Antithetics	IS	IS & Strat. (μ)	IS & Strat. (v_1)
2.5	0.04	8.0	8.1	246	248
	0.07	1.0	16.0	510	444
	0.10	0.8	173.0	3067	2861
5.0	0.04	4.2	8.1	188	211
	0.07	1.3	11.0	241	292
	0.10	1.0	27.0	475	512
10.0	0.04	3.7	6.6	52	141
	0.07	1.4	7.8	70	185
	0.10	1.1	12.0	110	244
15.0	0.04	3.6	5.3	15	67
	0.07	1.6	6.0	22	112
	0.10	1.2	8.0	31	158

ated with flex caps, but discontinuities also arise from trigger or barrier features. Finally, we address the overhead of the method and the performance of techniques to reduce that overhead.

Options with a Continuous Payoff Function

Our first example is a caplet. Note that since the forward rates are continuous functions of the Z, the discounted caplet payoff function $D_{N+1}C_{N+1}$ is also a continuous function of the Z. Furthermore, it is twice-differentiable, except at the points where $\exp[\Delta t\, F(N, N)] - 1 - K\Delta t = 0$ [i.e., except at the kink in the payoff; see (16)].

Exhibit 7 displays the estimated variance ratios for caplets with a range of maturities T and strikes K. The table lists the ratios for antithetics, importance sampling alone (the column labeled IS), importance sampling with stratification upon μ (the IS & Strat. (μ) column), and importance sampling with stratification upon the best eigenvector v_1 (the IS & Strat. (v_1) column).

For each maturity T, there are three different strikes that range from in the money to out of the money. As a point of reference, the at-the-money strikes for T = 2.5, 5.0, 10.0, and 15.0 are 0.0564, 0.0601, 0.065, and 0.0683, respectively.

From Exhibit 7, we observe that:

1. The effectiveness of antithetics decreases as the strike K increases (with maturity T fixed), i.e., as the caplet becomes more out of the money. Similarly, the effectiveness of antithetics decreases as T increases (with K fixed).
2. The effectiveness of importance sampling alone increases as K increases (with T fixed). This is consistent with studies in other application areas in which a properly chosen IS method becomes more effective at estimating a rare event probability as the event becomes rarer (see, e.g., Heidelberger [1995]). In this setting, as K increases, the instrument becomes more out of the money, and the estimation problem takes on more of a rare event simulation flavor. In addition, for a fixed K, IS becomes less effective as T increases.
3. IS with stratification on either μ or v_1 is much more effective than IS alone. The effectiveness increases as K increases, but decreases as T increases. Stratifying on the eigenvector v_1 becomes more effective than stratifying on the optimal drift vector μ as T increases.

A cap is a sum of caplets over a specified interval of time. We let T_0 and T_1 denote the times of the first and last caplet payments in the cap. Then $T_0 = N_0\Delta t$, and $T_1 = N_1\Delta t$ for some integers N_0 and N_1. With discount factor D_i and caplet payoff C_i as defined above, the cap has discounted payoff

$$G(Z) = \sum_{i=N_0}^{N_1} D_i C_i \tag{18}$$

This is also a continuous function of the Gaussian increments.

Estimated variance ratios for caps are listed in Exhibit 8. As with caplets, the effectiveness of IS with stratification typically increases as K increases, but decreases as the time interval $(T_1 - T_0)$ increases. Stratification on the eigenvector provides little or no benefit over stratification on μ.

For a given strike K, the variance ratios are typi-

EXHIBIT 8
Estimated Variance Ratios for Caps in Three-Factor HJM Model

T_0	T_1	K	Antithetics	IS	IS & Strat. (μ)	IS & Strat. (v_1)
0.25	2.5	0.04	2.1	5.3	20	19
		0.07	1.1	23.0	158	161
		0.10	1.1	285.0	1435	1384
0.25	5.0	0.04	8.0	5.2	23	21
		0.07	1.2	13.0	54	48
		0.10	0.9	41.0	176	152
0.25	10.0	0.04	5.3	4.9	15	14
		0.07	1.4	8.4	22	24
		0.10	1.1	16.0	39	40
0.25	15.0	0.04	5.5	4.0	8.9	8.5
		0.07	1.5	5.5	8.4	8.3
		0.10	1.2	8.2	12	12
5.25	10.0	0.04	4.2	6.2	51	43
		0.07	1.4	8.4	44	42
		0.10	1.1	15.0	43	41
10.25	15.0	0.04	4.9	5.2	25	43
		0.07	1.6	6.2	36	38
		0.10	1.2	9.0	46	36

cally not as high as those for caplets. For a caplet, the simulation is optimized for a specific payment, but for a cap, the simulation is optimized for a range of payments. What is best for the caplet near the beginning of the interval may not be good for the caplet near the end of the interval.

The results also suggest that it may be more efficient to decompose a cap over a long interval into a sum of caps over shorter intervals. Rather than running a single simulation to estimate the entire cap, one could perform separately optimized simulations to estimate the caps over each of the subintervals. For example, the cap over the interval [0.25, 15.0] could be decomposed into, say, a cap over the interval [0.25, 10.0] and the cap over the interval [10.25, 15.0]. As the method produces greater variance reductions over the narrower intervals, this decomposition approach may produce lower variance estimates from the same amount of computing time. The success of this decomposition approach of course depends upon many implementation details such as the path generation time.

In the case of European swaptions, suppose the underlying interest rate swap starts at $T_E(= N_E\Delta t)$ and ends $T(= N\Delta t)$ years later. We may represent the value of the swap as the difference between floating- and fixed-rate bonds. With a fixed rate of C% per year, the fixed side makes payments of C/2 every six months starting at $T_E + 0.5$ and in addition pays a principal of 100 at time $T_E + T$. Let P_i be the fixed-rate payment at time $i\Delta t$: $P_i = C/2$ if $N_E < i < N_E + N$ and $(i - N_E)$ is divisible by 2; $P_i = C/2 + 100$ if $i = N_E + N$; and $P_i = 0$ otherwise. The value of the fixed-rate bond at $N_E\Delta t$ is then

$$B_C(N_E, N) = \sum_{i=N_E}^{N_E+N} B(N_E, i)P_i \tag{19}$$

where $B(N_E, i)$ is defined in (13).

Valuing the floating-rate bond at par, the value of the swap at $N_E\Delta t$ becomes $100 - B_C(N_E, N)$. A $T_E \times T$ floating-for-fixed swaption thus has discounted payoff

$$G(Z) - D_{N_E}\max[0,\ 100 - B_C(N_E, N)] \tag{20}$$

This again is a continuous function of the underlying Z.

Estimated variance ratios are given in Exhibit 9. Results are qualitatively similar to those for caplets and caps. We obtain large variance reductions when using IS and stratification. Variance reductions are best for more out-of-the money instruments (lower C) and shorter option expiration times T_E.

Exhibit 9

Estimated Variance Ratios for Swaptions in Three-Factor HJM Model

T_E	T	C (%)	Antithetics	IS	IS & Strat. (μ)	IS & Strat. (v_1)
1	5	5	1.2	12.0	218	231
		6	2.7	6.3	205	207
1	10	5	1.1	18.0	284	311
		6	1.8	7.5	226	242
2	5	5	1.3	9.9	187	172
		6	2.3	6.4	173	146
2	10	5	1.2	13.0	232	222
		6	1.8	7.4	204	179
5	5	5	1.4	8.4	141	163
		6	2.0	6.4	126	152
5	10	5	1.3	11.0	183	205
		6	1.7	7.5	154	182

The final example is a yield spread option. We consider a long-term bond of maturity $T_L(= N_L\Delta t)$; a short-term bond of maturity $T_S(= N_S\Delta t)$; and an exercise time $T_E(= N_E\Delta t)$. Define $Y_S(i) = \Sigma_{j=0}^{NS-1} F(i, i + j)/N_S$ and $Y_L(i) = \Sigma_{j=0}^{N_L-1} F(i, i + j)/N_L$; these are the yields on short-term and long-term zero-coupon bonds as of $i\Delta t$. At time 0, the current spread between these yields is $\delta = [Y_L(0) - Y_S(0)]$, and at time T_E, the spread

Exhibit 10

Estimated Variance Ratios for Option on Zero-Coupon Bond Yield Spread in Three-Factor HJM Model

T_E	K	Antithetics	IS	IS & Strat. (μ)	IS & Strat. (v_1)
1.0	1	1.7	7.8	104	199
	2	1.1	29	355	419
2.5	1	1.7	7.8	50	146
	2	1.2	15	126	165
5.0	1	1.8	7.8	32	129
	2	1.2	12	60	118

Long-term bond maturity T_L = 15 years, short-term bond maturity T_S = 3 years, and strike = K times current spread.

is $Y_L(N_E) - Y_S(N_E)$. If the strike is K times the current spread, then the discounted payoff is

$$G(Z) = 100D_{N_E}(Y_L(N_E) - Y_S(N_E) - K\delta)^+ \quad (21)$$

For this study, we fix $T_S = 3$ years and $T_L = 15$ years. The results, which are shown in Exhibit 10, are similar to the previous cases: large variance reductions that increase as the option becomes more out of the money (K increases) but decrease as the option expiration time T_E increases. Stratification on the best eigenvector v_1 provides significant improvements over stratification on μ.

Options with a Discontinuous Payoff Function

The theory developed in GHS [1999] assumes, among other things, that the payoff function G is a continuous function of the increments Z. There are many types of options, however, for which this is not true. While IS and stratification are still valid techniques in this case, the asymptotic theoretical justification for selecting an IS distribution by maximizing the payoff times the probability density function is not, strictly speaking, valid. Furthermore, when the payoff is discontinuous, optimization using standard non-linear programming packages becomes dicey, at best; such packages often assume the existence of gradients.

Since our approach works so well on continuous payoff functions, we seek to adapt it to options with discontinuous payoff functions. The general approach we take is to approximate the option's true payoff function G(z) by a continuous function $\hat{G}(z)$, and then to pick the IS drift μ by maximizing $\hat{G}(z)\phi_n(z)$. Similarly, a stratification vector can be found from the eigenvectors of the Hessian matrix of $\log[\hat{G}(z)]$ evaluated at μ. (In both cases, the approximation $\hat{G}$ is used solely to design the variance-reduction method; the simulation itself uses the true G.) There is clearly much room for experimentation here, and we report on the results of approaches that work the best among those tested.

Consider a flex cap in which the holder receives the payoffs of the first J caplets to expire in the money over N periods, J < N. We assume that the first potential payment occurs at time 0.25, and we let $T = N\Delta t$ be the time of the last potential payment. As before, let C_i denote the payoff of the i-th caplet. Let N_J denote the (random) index of the J-th caplet to expire in the money — that is, the J-th caplet for which $C_i > 0$. Then the flex cap's discounted payoff function is

$$G(Z) = \sum_{i=1}^{\min(N_J, N)} D_i C_i \quad (22)$$

This is clearly a discontinuous function of Z, since small changes in Z can change the integer-valued variable N_J. To approximate this G, we select a $\hat{G}$ that is the payoff function for an ordinary cap with initial payment at time $N_0\Delta t$ and final payment at time $N_1\Delta t$ for appropriate values of N_0 and N_1. For an in-the-money flex cap, we expect that, with high probability, the first J payments will be positive, suggesting that we should set $N_0 = 1$ and $N_1 = J$. This is the approach we follow,

EXHIBIT 11

Estimated Variance Ratios for Flex Caps in Three-Factor HJM Model

T	K	J	Antithetics	IS	IS & Strat. (μ)	IS & Strat. (v_1)
2.0	0.05	2	2.1	1.5	2.3	2.3
		4	3.2	3.8	4.8	4.5
		6	4.3	7.9	18	16
		8	3.4	6.2	37	34
2.0	0.0532	2	1.9	1.7	2.4	1.6
		4	2.7	4.3	9.3	5.0
		6	2.4	8.3	48	37
		8	1.9	7.7	47	38
2.0	0.07	2	1.1	11.0	28	20
		4	1.0	26.0	150	144
		6	1.0	31.0	199	192
		8	1.0	30.0	170	176
5.0	0.05	5	2.3	2.0	2.2	2.1
		10	3.0	3.8	4.0	3.4
		15	3.5	7.7	17	16
		20	2.7	6.5	33	29
5.0	0.0564	5	2.0	1.8	2.0	1.8
		10	2.4	4.5	6.5	5.6
		15	2.1	8.2	25	35
		20	1.7	8.2	37	34
5.0	0.08	5	1.2	6.0	11	9.6
		10	1.2	15.0	74	66
		15	1.1	20.0	133	128
		20	1.1	19.0	105	88

except we find that setting $N_1 = J + 1$ produces somewhat better results.

For a deep out-of-the-money flex cap, we expect that if any payoffs are positive, then with high probability, they will be from among the last J payments. This suggests that we should set $N_0 = N - J$ and $N_1 = N$ (the approach we follow). If the flex cap becomes at the money at some intermediate time $L\Delta t$, then N_0 should be about L, and N_1 should be about max(N, L + J). In this case, to account for the variation in the time of the first and last payments, the optimizing cap uses $N_0 = L - 1$ and $N_1 = \max(N, L + J + 1)$.

The results are given in Exhibit 11. For each T there are three strikes K: the first in the money, the second the K for which the caplet paying at time T/2 is at the money, and the third a deep out-of-the-money strike. Then, for each pair of T and K, we set the maximum number of payoffs J to be 0.25N, 0.50N, 0.75N, and N. Results are only marginally better than antithetics for in- and at-the-money caplets when J = 0.25N, but improve as J increases. For small values of J, the variability in the time of the first and last payments reduces the amount of variance reduction. Variance reductions for the out of-the-money strikes are much better, and again increase as J increases.

We have undertaken a similar investigation for an option with a trigger event — a simplified trigger swap — and obtain results broadly consistent with those reported for flex caps. In particular, the discontinuity arising from the trigger markedly diminishes the effectiveness of the method. We obtain somewhat better results when the possible times of the trigger event are tightly constrained. (This is consistent with the results for the flex caps, where we observed greater variance reduction for larger J; larger J imposes greater constraints on which caplets exercise.) Details are available from the authors.

Overhead and Approximate Optimization

We use the GRG2 optimization package from Optimal Methods, Inc., which performs satisfactorily for all the continuous payoff functions considered. While convergence to a suboptimal local maximum sometimes occurs when the optimization is initialized at a random starting point, good results are obtained when the optimization is started with all components equal to 0, and the optimization is preceded by a "Phase I" step to find a non-zero payoff.

For example, in a cap, this is accomplished by setting a constraint $C_j \geq 0$ for one of the caplets in the payoff interval. Typically, convergence to an accurate, optimal μ occurs in between 5 to 10 × n paths (evaluations of G[(z)], where n is the number of variables. In other examples with discontinuities, we sometimes find that the optimization fails to converge when initialized to a random starting point.

To compute the Hessian matrix, we use a central differencing scheme as described in Press et al. [1996, p. 187]. To compute the Hessian with this approach requires about $2n^2$ paths. To reduce both the optimization and Hessian overheads, we apply the approaches described in Section III. We obtain the most consistent results using linear interpolation.

Exhibit 12 lists the results when applying this approach to caplets. The columns show the number of paths required to solve the ensuing optimization problem, the number of paths to compute the Hessian matrix, and the resulting variance ratios when combining IS with stratification upon μ or the approximate eigenvector. We observe dramatic savings in overhead with little loss in variance reduction.

For example, with T = 10.0, the full problem has 117 (= 3 × 39) variables. With the linear(3) method, the number of variables is reduced to nine, and the number of optimization paths is reduced by

EXHIBIT 12

Overheads and Estimated Variance Ratios Using Approximate Optimization for Caplets in Three-Factor HJM Model (K = 0.07)

T	Method	Optimization Paths	Hessian Paths	Variance Ratio (μ)	Variance Ratio (v_1)
2.5	Full	183	1459	510	444
	Linear(3)	101	163	415	465
	Linear(4)	123	289	417	467
5.0	Full	480	6499	259	252
	Linear(3)	107	163	282	278
	Linear(4)	136	289	296	292
10.0	Full	718	27,379	70	185
	Linear(3)	84	163	64	111
	Linear(4)	136	289	75	156
15.0	Full	1257	62,659	22	112
	Linear(3)	95	163	16	44
	Linear(4)	118	289	18	74
	Linear(5)	143	451	18	95

Exhibit 13

Overheads and Estimated Variance Ratios Using Approximate Optimization for Yield Spread Option in Three-Factor HJM Model

T	Method	Optimization Paths	Hessian Paths	Variance Ratio (μ)	Variance Ratio (v_1)
2.5	Full	233	1801	126	165
	Linear(3)	109	163	123	187
	Linear(4)	138	289	120	184
5.0	Full	442	7201	60	118
	Linear(3)	98	163	63	105
	Linear(4)	136	289	65	109

about a factor of ten with little loss in variance reduction (when stratifying on μ). The cost to compute the Hessian matrix is reduced by a factor of 171, but we still obtain about 60% of the benefit of stratifying on the best eigenvector. Similar results for yield spreads are shown in Exhibit 13.

V. CONCLUSIONS

We have explored the application of an efficient Monte Carlo algorithm to pricing path-dependent European-style interest rate options in a multifactor HJM setting. The approach is based on importance sampling and stratification. When the option's payoff function is continuous in the underlying increment variables, large variance reductions of one to two orders of magnitude can be obtained using this approach.

Typically, the variance reductions increase as the instrument becomes more out of the money and decrease as the time interval over which the instrument is defined increases. The effectiveness of the method is reduced for options with discontinuous payoff functions, but may still produce quite useful variance reductions.

The method involves overhead: solving a multidimensional optimization problem and, optionally, computing a Hessian matrix and its eigenvectors. In high dimensions, this overhead may become quite significant compared to the number of paths desired for actually pricing the option. By using approximation techniques, almost all the potential variance reduction can be achieved with greatly reduced overheads.

Appendix

Summary of Simulation Algorithm

Purpose: Estimate $E[G(Z)]$, $Z \sim N(0, I_n)$. (Interpret $G(Z)$ as discounted payoff of a derivative security when paths of underlying assets are driven by Z.)

Preprocessing: Find μ as solution to optimization problem:

$$\frac{1}{k}\sum_{i=1}^{k} G(Z^{(i)})$$

or equivalently:

$$\max_{z\in R^n}\{F(z) - \frac{1}{2}z'z\},\ F = \log G$$

Choose stratification direction u:

- Either $u \leftarrow \mu$, with m as above, or
- Find eigenvectors $v_1, \ldots, v_n$ and associated eigenvalues $\lambda_1, \ldots, \lambda_n$ of Hessian for F at μ, ordered so that

$$\left(\frac{\lambda_1}{1-\lambda_1}\right)^2 \geq \left(\frac{\lambda_2}{1-\lambda_2}\right)^2 \geq \cdots \geq \left(\frac{\lambda_n}{1-\lambda_n}\right)^2$$

and choose $u \leftarrow v_1$.

Simulation: Repeat these steps for k replications:

1. Draw $U^{(1)}, \ldots, U^{(M)}$ independently and uniformly over (0, 1). (M is the number of strata.)
2. Set
$$V^{(i)} \leftarrow \frac{i-1}{M} + \frac{U^{(i)}}{M}$$
$i = 1, \ldots, M$.
3. Set $X^{(i)} \leftarrow \Phi^{-1}(V^{(i)})$, $i = 1, \ldots, M$. (Φ is the standard normal cumulative distribution.)
4. Draw $Y^{(1)}, \ldots, Y^{(M)}$ from $N(0, I_n)$.
5. Set $Z^{(i)} \leftarrow uX^{(i)} + (I_n - uu')Y^{(i)}$, $i = 1, \ldots, M$. (This is a stratified sample from $N(0, I_n)$, stratified along direction u.)
6. Add drift vector: $Z^{(i)} \leftarrow Z^{(i)} + \mu$, $i = 1, \ldots, M$.
7. Evaluate likelihood ratios: $L^{(i)} \leftarrow \exp(-\mu' Z^{(i)} + 1/2\mu'\mu)$, $i = 1, \ldots, M$. (If $u = \mu$, this is the same as $L^{(i)} \leftarrow \exp(-X^{(i)} - 1/2\mu'\mu)$.)
8. Evaluate discounted payoffs $G(Z^{(i)})$, and average: $\hat{G} = M^{-1}\Sigma_{i=1}^{M} G(Z^{(i)})L^{(i)}$.

Repeating this procedure k times yields independent replications $\hat{G}_1, \ldots, \hat{G}_k$ of $\hat{G}$. (The total numer of paths generated is kM.)

Output: Sample mean $\bar{G} = k^{-1}\Sigma_{i=1}^{k}\hat{G}_i$ and standard error $S/\sqrt{k}$, with

$$S = \sqrt{\frac{1}{k-1}\sum_{i=1}^{k}(\hat{G}_i - \overline{G})^2}$$

ENDNOTES

This research is supported in part by National Science Foundation grants DMI-9457189 and DMI-9625297.

[1] We use N(a, B) to denote the normal distribution with mean vector a and covariance matrix B; I_d denotes the d × d identity matrix.

[2] Here and throughout, a prime denotes vector or matrix transpose.

[3] In the continuous-time limit represented by (1), we may change the drift of the driving Brownian motion but not its covariance structure if we are to maintain equivalence of the associated probability measures.

[4] Fast approximations to the inverse normal Φ^{-1} are given in Marsaglia, Zaman, and Marsaglia [1994] and Moro [1995].

[5] Alternatively, the stratification method could be applied in multiple dimensions: Partition the unit hypercube $(0, 1)^n$ into M^n subhypercubes by partitioning each coordinate into M strata; sample uniformly from each subhypercube; and map each coordinate to the real line using Φ^{-1}. The result is a stratified sample from $N(0, I_n)$.

The drawback of this generalization is that it requires M^n values to generate a complete stratified sample, so if n is even moderately large, M must be quite small. Rather than apply a coarse stratification to many coordinates, we apply a finer stratification to a small number (typically one) of carefully chosen directions. Yet another variant of the method applies quasi-Monte Carlo to a few important directions and ordinary Monte Carlo to the rest.

[6] See, e.g., Hull [1997, p. 427] for a derivation of the discrete drift.

[7] As noted in Heath, Jarrow, and Morton [1992], a deterministic proportional volatility is incompatible with the continuous-time dynamics in (10); however, since we will keep Δt fixed in (12) and bounded away from zero, we can use this specification in the simulation.

[8] For d = 1, N = 5, and k = 3, the **M** corresponding to the linear interpolation method is given by

$$\begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ 0 & 1 & 0 \\ 0 & 1/2 & 1/2 \\ 0 & 0 & 1 \end{pmatrix}$$

For d = 3, N = 5, and k = 9 (i.e., with three variables per factor), the **M** will be a block-diagonal, 15 × 9, matrix with matrices like the above constituting each block.

[9] For if $v_i = \mathbf{M}x$ [with v_i the i-th eigenvector of H, ranked according to the criterion in (9)] for some x, then $(\mathbf{M'M})^{-1}\mathbf{M'}H\mathbf{M}x = \lambda_i x$, and x will be among the $\bar{v}_i$ [the i-th eigenvector of the reduced matrix ranked according to the criterion in (9)], up to a scalar multiple.

Finding the eigenvalues and eigenvectors of $(\mathbf{M'M})^{-1}H_M$ is a "generalized eigenproblem" as described in Press et al. [1996, p. 462]: find (x, λ) to solve $B^{-1}Ax = \lambda x$ where A and B are symmetric, and B is positive-definite. The eigenvectors are real, and efficient algorithms are built into many scientific libraries.

REFERENCES

Boyle, P., M. Broadie, and P. Glasserman. "Simulation Methods for Security Pricing." *Journal of Economic Dynamics and Control*, 21 (1997), pp. 1267-1321.

Fishman, G. *Monte Carlo: Concepts, Algorithms, and Applications*. New York: Springer-Verlag, 1996.

Glasserman, P., P. Heidelberger, and P. Shahabuddin. "Asymptotically Optimal Importance Sampling and Stratification for Pricing Path-Dependent Options." *Mathematical Finance*, 9 (1999), pp. 117-152.

Hammersley, J., and D. Handscomb. *Monte Carlo Methods*. London: Methuen, 1964.

Heath, D., R. Jarrow, and A. Morton. "Bond Pricing and the Term Structure of Interest Rates: A New Methodology for Contingent Claims Valuation." *Econometrica*, 60 (1992), pp. 77-105.

Heidelberger, P. "Fast Simulation of Rare Events in Queueing and Reliability Models." *ACM Trans. Modeling and Computer Simulation*, 5, No. 1 (1995), pp. 43-85.

Hull, J.C. *Options, Futures, and Other Derivative Securities*, 3rd Edition. Upper Saddle River, NJ: Prentice-Hall, 1997.

Marsaglia, G., A. Zaman, and J. Marsaglia. "Rapid Evaluation of the Inverse of the Normal Distribution Function." *Statistics and Probability Letters*, 19 (1994), pp. 259-266.

Moro, B. “The Full Monte.” *Risk*, 8 (February 1995), pp. 57-58.

Newton, N.J. “Continuous-Time Monte Carlo Methods and Variance Reduction.” In L.C.G. Rogers and D. Talay, eds., *Numerical Methods in Finance*. New York: Cambridge University Press, 1997, pp. 22-42.

Press, W.H., S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C, 2nd Edition*. New York: Cambridge University Press, 1996.

Schoenmakers, J.G., and A.W. Heemink. “Fast Valuation of Financial Derivatives.” *Journal of Computational Finance*, 1 (1997), pp. 47-62.