RC 21367 (96265) January 11, 1999 Computer Science/Mathematics

Research Report

Importance Sampling in the Heath-Jarrow-Morton Framework

Paul Glasserman
Columbia Business School
Columbia University
New York, New York 10027

Philip Heidelberger
IBM T. J. Watson Research Center
P. O. Box 218
Yorktown Heights, NY 10598

Perwez Shahabuddin
IEOR Department
Columbia University
New York, New York 10027

IBM Research Division
Almaden - Austin - Beijing - Haifa - T. J. Watson - Tokyo - Zurich

Importance Sampling in the Heath-Jarrow-Morton Framework

Paul Glasserman
Graduate School of Business
Columbia University
New York, NY 10027

Philip Heidelberger
IBM Research Division
T.J. Watson Research Center
Yorktown Heights, NY 10598

Perwez Shahabuddin
IEOR Department
Columbia University
New York, NY 10027

Abstract

This paper develops a variance reduction technique for pricing derivatives in high-dimensional multifactor models, with particular emphasis on term structure models formulated in the Heath-Jarrow-Morton framework. A premise of this work is that the largest gains in simulation efficiency come from taking advantage of the structure of both the cashflows of a security and the model in which it is priced; for this to be feasible in practice requires that the identification and use of relevant structure be automated. 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. These further enhance the applicability of the method.

1 Introduction

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: decreasing 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 yet 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 this work is that the largest gains from the use of variance reduction techniques rely on exploiting 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 a previous paper (Glasserman, Heidelberger, and Shahabuddin [4], henceforth GHS), we proposed and analyzed 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 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 may be viewed as 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 gave 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 $\epsilon \to 0$. Asymptotic optimality properties of the method are established in GHS. We also tested the method on three types of examples: Asian options, a stochastic volatility model, and the Cox-Ingersoll-Ross short rate model. Numerical results reported there indicate very substantial potential for variance reduction using the method.

The purpose of this paper 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 [5] type. Using, for example, quarterly rates with a twenty-year horizon makes the state vector for such a model 80-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, flex caps, and trigger swaps. Our numerical results support the viability of the method in the HJM setting.

The rest of this paper is organized as follows. Section 2 describes the general procedure in GHS, giving relevant background on importance sampling and stratification along the way. Section 3 specializes to the HJM setting. Approximations are developed in Section 4 and numerical results are reported in Section 5.

2 Overview of the Method

Figure 1 gives a self-contained summary of the simulation algorithm combining importance sampling and stratified sampling as in GHS. The algorithm 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 nonnegative values. (Interpret G as the discounted payoff of a derivative security and Z as the stochastic inputs to the simulation.) A reader interested only in the final algorithm may proceed directly to Figure 1; the rest of this section provides background for the algorithm and details of some of the steps.

2.1 The Setting

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

$$dX_t = \mu(X_t, t)\,dt + \sigma(X_t, t)\,dW_t, \tag{1}$$

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

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

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

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

are independent standard multivariate normal random vectors. (We use $N(a, B)$ to denote the normal distribution with mean vector a and covariance matrix B; I_d denotes the $d \times d$ identity matrix.) 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

$$\phi_n(z) = (2\pi)^{-n/2} \exp(-\tfrac{1}{2} z' z), \quad z \in \mathbf{R}^n.$$

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

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

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

2.2 Importance Sampling

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

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

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

$$\alpha = \int_{R^n} G(z)\frac{\phi_n(z)}{g(z)} g(z)\, dz \tag{4}$$

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

$$\hat{\alpha}_g(M) = \frac{1}{M}\sum_{i=1}^{M} G(Z^{(i)})\frac{\phi_n(Z^{(i)})}{g(Z^{(i)})}.$$

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

Different choices of g will result in estimators with different variances. In general, there is no guarantee that sampling from g rather than ϕ_n will actually reduce variance, but the potential for variance reduction through importance sampling is enormous. Indeed, if G is nonnegative and if we choose g proportional to the product of G and ϕ_n, it is not hard to see that importance sampling yields a zero-variance estimator. The catch, of course, is that the normalization constant required to make $G(z)\phi_n(z)$ a probability density is precisely 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 g 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 restricts g to be a multivariate normal density $N(\mu, I_n)$ for some $\mu \in \mathbf{R}^n$; i.e., $g(z) = \phi_n(z - \mu)$. For any choice of μ, the likelihood ratio under this change of measure is particularly simple, reducing to

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

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

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

or, equivalently,

$$\max_{z\in\mathbf{R}^n}\{F(z)-\tfrac{1}{2}z'z\}, \tag{7}$$

where $F(z)=\log G(z)$ (taking $\log 0=-\infty$). 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 and whose covariance matrix coincides with that of the original measure. (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.) For any choice of μ, the resulting estimator is the average of independent copies of

$$G(Z)\exp(-\mu'Z+\tfrac{1}{2}\mu'\mu), \quad Z\sim N(\mu,I_n);$$

i.e., of

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

Under conditions in GHS, the μ found by solving (7) will satisfy the first-order conditions $\nabla F(\mu)=\mu'$. 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, (8) reduces to

$$\begin{aligned}\exp(F(\mu+Z)-\mu'Z-\tfrac{1}{2}\mu'\mu) &\approx \exp(F(\mu)+\nabla F(\mu)Z-\mu'Z-\tfrac{1}{2}\mu'\mu)\\ &= \exp(F(\mu)-\tfrac{1}{2}\mu'\mu),\end{aligned}$$

which is constant. 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 for a more extensive analysis and discussion. See Boyle et al. [1], Newton [9], and Schoenmakers and Heemink [11] for other approaches to importance sampling in option pricing.

2.3 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 [2] or Hammersley and Handscomb [3].)

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 100 independent samples from the normal distribution, it is unlikely that exactly 50 will be positive and 50 negative. Stratified sampling refers to any mechanism that, in this example, ensures that indeed half of the samples are positive and half are negative.

More generally, we can partition the real line into M strata and sample from these in the correct proportions. In our method, 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 of these subintervals. We then apply the inverse cumulative normal distribution to each of these M values in $(0,1)$. The resulting M values in $(-\infty,\infty)$ constitute a stratified sample from the normal distribution.

To put this more precisely, let $U^{(1)},\ldots,U^{(M)}$ be independent and uniformly distributed on $(0,1)$. Set

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

to make $V^{(i)}$ uniformly distributed over $((i-1)/M, i/M)$. Then set $X^{(i)} = \Phi^{-1}(V^{(i)})$, $i = 1,\ldots,M$, where

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

(Fast approximations to Φ^{-1} are given in Marsaglia, Zaman, and Marsaglia [7] and Moro [8].) Each $X^{(i)}$ lies between the $(i-1)/M$th and i/Mth fractiles of the normal distribution and has the distribution of a normal random variable conditioned to lie in this range.

This method generalizes to multiple dimensions: partition the unit hypercube $(0,1)^n$ into M^n sub-hypercubes by partitioning each coordinate into M strata; sample uniformly from each sub-hypercube; 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 method 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 try to stratify many coordinates of $N(0,I_n)$, we instead focus on stratifying along a small number (typically 1) of carefully chosen directions. (Another variant of the method applies quasi-Monte Carlo to a few important directions and ordinary Monte Carlo to the rest.)

In sampling from $N(0, I_n)$ we could, for example, stratify any one of the n coordinates and sample the others independently. 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 $u \in \mathbf{R}^n$ with $u'u = 1$. If $Z \sim N(0, I_n)$, then $u'Z \sim N(0,1)$. Using the one-dimensional algorithm described above, 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:

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

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 ith value of $u'Z$;
- draw $Y^{(i)}$ from $N(0, I_n)$, $i = 1, \ldots, M$, independent of each other and of the $X^{(i)}$;
- set $Z^{(i)} = uX^{(i)} + C_u Y^{(i)}$, $i = 1, \ldots, M$, where C_u is any $n \times n$ 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 we proposed and analyzed two strategies for doing this. The simpler method sets $u = \mu(= z^*)$, the optimal vector found for importance sampling. The other method finds the best direction for a quadratic approximation to $F = \log G$. (Recall that our importance sampling method can be interpreted as eliminating the linear part of F.) In GHS we proved 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

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

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 of F. The Hessian should be evaluated at the point $z^*(= \mu)$ 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 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

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

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 $\sqrt{k}$ (as in the last step in Figure 1). We thus arrive at, e.g., an approximate 95% confidence interval of the form

$$\bar{G} \pm 1.96 \frac{s}{\sqrt{k}},$$

since $1 - \Phi(1.96) = 0.025$. 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.

3 The Heath-Jarrow-Morton Setting

As explained in Section 1, the objective of this paper 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 [5] framework provides a particularly appropriate setting for this investigation.

3.1 Simulation Model

We start by reviewing the continuous-time HJM formulation and then proceed to the discretized version used in a simulation. Let $f(s,t)$ be the instantaneous continuously compounded forward rate for time t as of time s, with $0 \le s \le t \le T^*$ for some ultimate maturity T^*. Let $B(s,t)$ be the time-s price of riskless bond paying \$1 at time t. Then

$$f(s,t) = -\frac{\partial}{\partial t} \log B(s,t) \quad \text{and} \quad B(s,t) = \exp\left(-\int_s^t f(s,u)\, du\right).$$

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 $\{f(s,t), s \le t \le T^*\}$ over the time interval $0 \le s \le T^*$. For a d-factor model take W_t to be

Algorithm Summary

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

$$\max_{z \in \mathbf{R}^n} \{G(z) \exp(-z'z/2)\} \quad \left(\text{or equivalently } \max_{z \in \mathbf{R}^n} \{F(z) - \tfrac{1}{2} z'z\},\ F = \log G\right)$$

Choose stratification direction u:

- either $u \leftarrow \mu$, with μ as above, or
- find eigenvectors $v_1, \ldots, v_n$ and associated eigenvalues $\lambda_1, \ldots, \lambda_n$ of Hessian of 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 the following 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 $V^{(i)} \leftarrow \frac{i-1}{M} + \frac{U^{(i)}}{M}, \quad i = 1, \ldots, M$.
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: $L^{(i)} \leftarrow \exp(-\mu' Z^{(i)} + \frac{1}{2}\mu'\mu)$, $i = 1, \ldots, M$. (If $u = \mu$, this is the same as $L^{(i)} \leftarrow \exp(-X^{(i)} - \frac{1}{2}\mu'\mu)$.)
8. Evaluate discounted payoffs $G(Z^{(i)})$ and average: $\hat{G} = M^{-1} \sum_{i=1}^{M} G(Z^{(i)}) L^{(i)}$

Repeating this procedure k times yields independent replications $\hat{G}_1, \ldots, \hat{G}_k$ of $\hat{G}$.
Output: Sample mean $\bar{G} = k^{-1} \sum_{i=1}^{k} \hat{G}_i$ and standard error $s/\sqrt{k}$, with

$$s = \sqrt{\frac{1}{k-1} \sum_{i=1}^{k} (\hat{G}_i - \bar{G})^2}.$$

Figure 1: Summary of Simulation Algorithm

a d-dimensional standard Brownian motion. The arbitrage-free dynamics of the forward curve under the risk-neutral measure have the form

$$df(s,t) = \left(\sigma(s,t)' \int_s^t \sigma(s,u)\,du\right) dt + \sigma(s,t)'\,dW_t, \tag{10}$$

where $\sigma(s,t)$ is a d-vector for each s and t. Under technical conditions detailed in [5], the drift specified in (10) ensures that discounted bond prices

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

(i.e., bond prices divided by the value of the money market account) are martingales (in s).

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 $i\Delta t$; i.e.,

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

A simulation algorithm for the forward curve takes the form

$$F(i+1,j) = F(i,j) + m_{ij}\Delta t + \sqrt{\Delta t}\sum_{k=1}^{d} s_{ij}(k) Z_{i+1}(k), \quad j > i,\ i = 0,1,2,\ldots, \tag{11}$$

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

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

This choice ensures that the discrete discounted bond prices $D_i B(i,j)$ are martingales (in i) and thus keeps the model arbitrage-free. Here, $F(\ell,\ell)$ is the (Δt) short rate at time $\ell\Delta t$, the bond price is given by

$$B(i,j) = \exp\left(-\sum_{\ell=i}^{j-1} F(i,\ell)\Delta t\right) \tag{12}$$

and the discount factor is

$$D_i = \exp\left(-\sum_{\ell=0}^{i-1} F(\ell,\ell)\Delta t\right). \tag{13}$$

Observe that simulating a single step in (11) 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 and we now detail the volatility structure used. Our choice of factors is fairly *ad hoc*; our 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 s_{ij} in (11) 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 factors g_1, g_2, g_3 are normalized so that

$$g_1(j)^2 + g_2(j)^2 + g_3(j)^2 \equiv 1, \quad j = 1, 2, \ldots. \tag{14}$$

With this normalization, $\sigma(j-i)$ becomes the overall level of (proportional) volatility of the jth forward $F(\cdot, j)$ at the ith step. As noted in the original work of Heath, Jarrow, and Morton [5], a deterministic proportional volatility is incompatible with the continuous-time dynamics in (10); however, since we will keep Δt fixed in (11) and bounded away from zero, we are free to use this specification in the simulation.

In our numerical examples, we take $\Delta t = 0.25$ years, a typical discretization used in practice. We consider maturities of up to 20 years, so our initial forward curve is 80-dimensional. We initialize it by taking

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

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

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

This produces a humped term structure of volatility starting at 13.17%, increasing to a maximum of 20.19% at a maturity of 4 years and decreasing gradually to 12% at the end of 20 years. Finally, for the factors g_1, g_2, g_3 we proceed as follows. We form the symmetric matrix

$$C_{ij} = \exp(-0.0004(i-j)^2), \quad i, j = 1, \ldots, 80,$$

and find orthonormal eigenvectors $x_1, \ldots, x_{80}$ and associated eigenvalues $\alpha_1 \geq \alpha_2 \cdots \geq \alpha_{80}$. With these parameters we have

$$\frac{\alpha_1}{\sum_i \alpha_i} = 72.8\%, \quad \frac{\alpha_1 + \alpha_2}{\sum_i \alpha_i} = 95.6\%, \quad \frac{\alpha_1 + \alpha_2 + \alpha_3}{\sum_i \alpha_i} = 99.5\%,$$

suggestive of a model in which three factors suffice but one factor definitely does not. We complete the specification of the factors by setting

$$g_i(j) = \frac{\sqrt{\alpha_i} x_i(j)}{\sqrt{\alpha_1 x_1^2(j) + \alpha_2 x_2^2(j) + \alpha_3 x_3^2(j)}}, \quad j = 1, \ldots, 80, \; i = 1, 2, 3,$$

to enforce the normalization (14).

Figure 2 graphs $\sigma(j)g_i(j)$ 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 far ends of the forward curves in opposite directions; the third factor bends the forward curve by moving the ends in the the opposite direction of the middle.

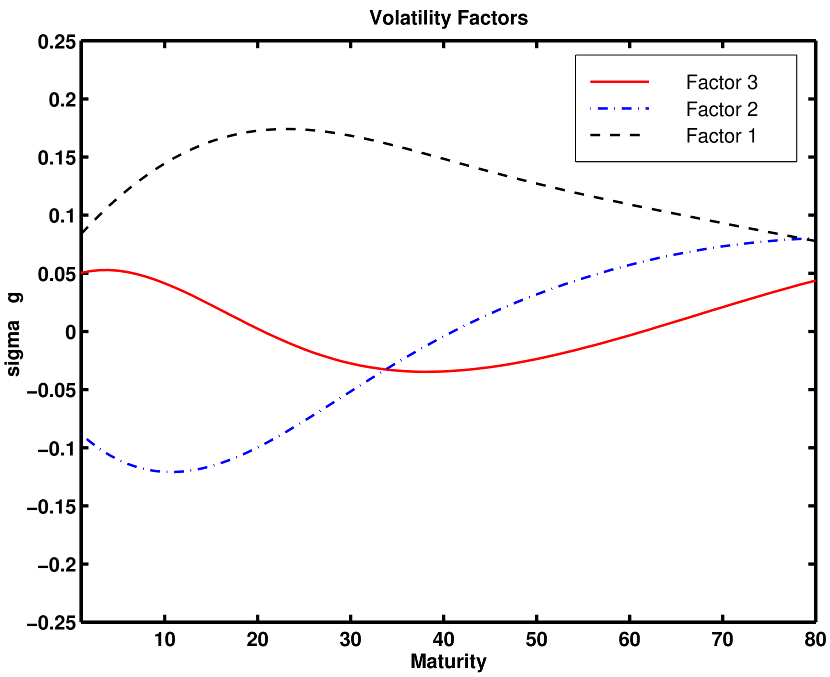

Figure 2: Factors for Numerical Examples

3.2 Applying the Variance Reduction Method

In Section 5, we undertake a numerical investigation of the application of the method of Section 2 in the HJM framework, but at this point an example may help to fix ideas and to motivate the approximations developed in the next section.

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, in the notation of Section 3.1, with a strike of K the caplet pays

$$C_{N+1} = \max\left(0, \left(e^{\Delta t F(N,N)} - 1\right) - K\Delta t\right) \times 100 \tag{15}$$

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

$$E\left[D_{N+1}C_{N+1}\right] \tag{16}$$

where D_{N+1} is defined in (13). Through (11) 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 will be useful to encode the $3N$-dimensional input vector Z as

$$Z_1(1), Z_2(1), \ldots, Z_N(1),\ Z_1(2), Z_2(2), \ldots, Z_N(2),\ Z_1(3), Z_2(3), \ldots, Z_N(3);$$

comparison with (11) 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 figures.)

The first step in implementing the procedure in Figure 1 solves the optimization problem (6) over all input vectors z. In the present context, 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 over terms of the type inside the expectation in (16). The result of the optimization is displayed in Figure 3 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 figure may be read as follows: on the optimal path, the increments of the first factor start at a large positive 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 downwards) 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 Figure 4 showing the evolution of the short rate $F(i,i)$ determined by the optimal paths — i.e., when the increments Z_i in (11) are evaluated along the optimal paths. We see that the impact of importance sampling in pricing a cap is to drive the short rate upwards 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 figure.)

The next step in the procedure of Section 2 finds a good direction for stratified sampling by either 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 Figure 5. 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, as will be seen in Section 5. Having selected a new drift vector μ for importance sampling and a good direction u for stratification

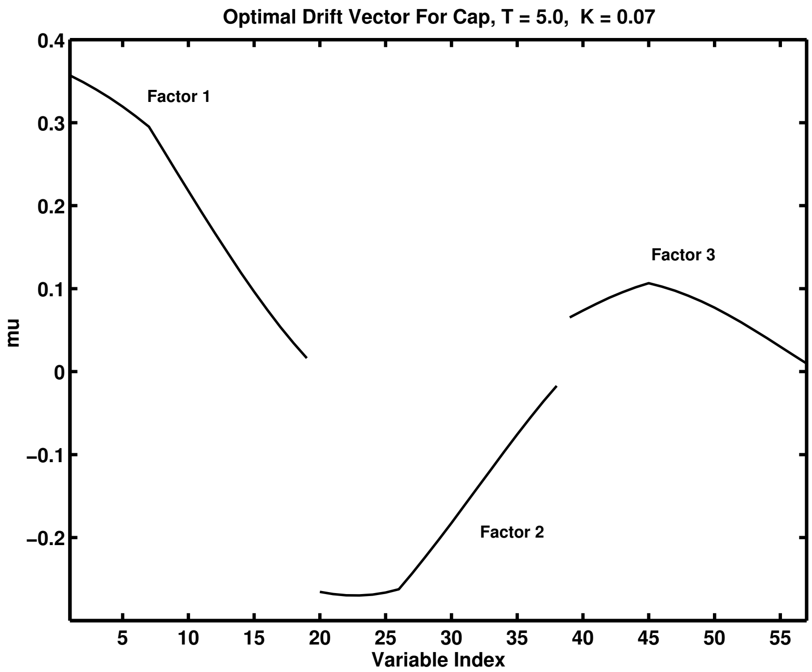

Figure 3: Optimal paths of factors for interest rate cap

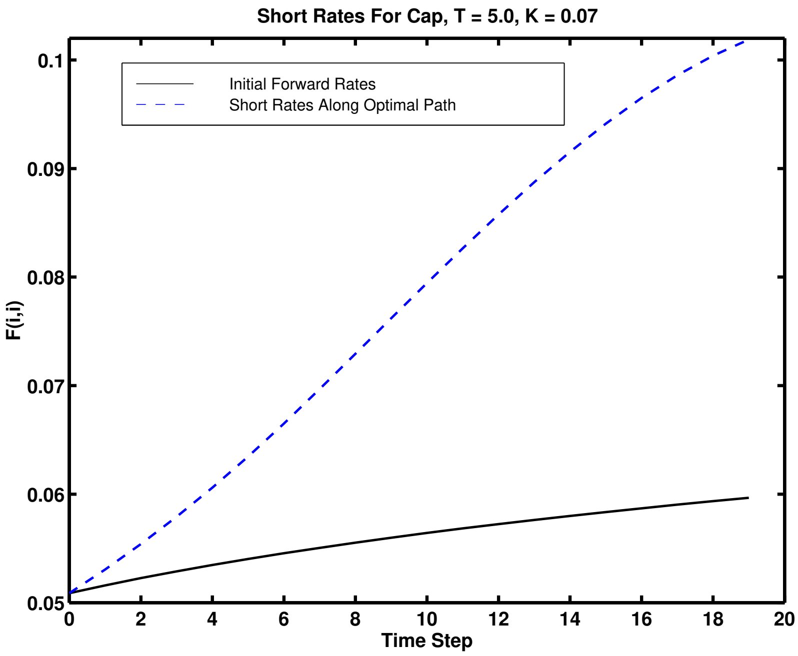

Figure 4: Optimal path of short rate

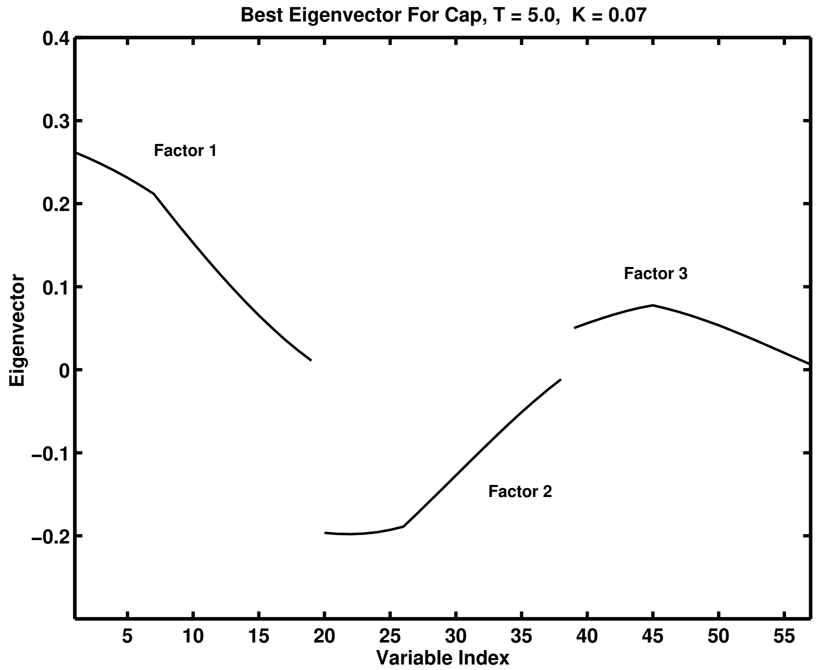

Figure 5: The best eigenvector for stratification for interest rate cap

(either μ or v_1), we can proceed with the simulation as described in Figure 1.

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 was by no means obvious in advance that the optimal change of drift would look anything like Figure 3, even if we had guessed that the optimal short rate path should look something like Figure 4. Nor was it obvious that a good direction for stratification would look like Figure 5. Yet we will see in our numerical results that these choices lead to very substantial variance reduction; an automated and systematic approach is needed to identify this 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. To balance the benefits of the method with the additional effort required, we investigate approximations in the next section.

4 Approximations

4.1 Approximate Optimization

In this section, we describe an approach for reducing the major overheads involved in applying the procedure: the overhead to solve the optimization problem and the overhead to compute the eigenvector(s) for stratification. The general approach was outlined in GHS. In Figure 3, if for a moment we forget the fact that the indices are discrete, then 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. In this study, we chose the knot points so that the time spacing between adjacent ones is the same, although that 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 $h(\cdot)$ 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 $G(h(\tilde{z}))\phi_n(h(\tilde{z}))$ with respect to $\tilde{z}$ in $\mathbf{R}^k$. One then maximizes $G(h(\tilde{z}))\phi_n(h(\tilde{z}))$ with respect to $\tilde{z}$ in $\mathbf{R}^k$. If $\tilde{\mu}$ is the optimal point in the new space, then one transforms back onto the original space, using the transformation $\mu = h(\tilde{\mu})$.

The linear interpolation method mentioned above, comes under the class of mappings $h(\cdot)$ which correspond to linear transformations from $\mathbf{R}^k$ to $\mathbf{R}^n$, i.e., $z = M\tilde{z}$ where M is an $n \times k$ matrix. 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 of the above type consitituting each of the blocks.

Figure 6 gives the optimal drift and the approximate optimal drift using linear interpolation for a caplet as described in Section 3.2, with $T = 10$. If each factor uses m knot points, we call the method "Linear(m)." For example, Linear(3) would use 3 variables per factor: the first, middle, and last increments. If all the variables are used, we call the method "Full." The legend in this figure also gives the norm of the difference between the optimal drift vector and the approximately optimal drift vector. Reduction of the computational overhead by using approximate optimization instead of exact optimization is discussed later in Section 5.

Another example of a linear transformation is to use the subspace defined by the principal components of the Brownian motion associated with each factor, corresponding to a few of the largest eigenvalues. For illustration, take the case of one factor, i.e., $d = 1$. The increments corresponding to this factor are $Z_1(1), Z_2(1), \ldots, Z_N(1)$. Define $W_i(1) = \sum_{j=1}^{i} Z_j(1)$. $W_i(1)$ is the value of the standard Brownian motion corresponding to that factor at the time points $i\Delta t$. If $Z(1)$ is the vector $(Z_1(1), \ldots, Z_N(1))'$ then $W(1) = (W_1(1), \ldots W_N(1))'$ is given by $W(1) = CZ(1)$ where C is the lower-triangular matrix

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}.$$

From this one can easily deduce that C is invertible and that the covariance of $W(1)$ is given by $\Sigma_W = CC'$. Since Σ_W is a symmetric, positive definite matrix, one can express Σ_W as $U_W \Lambda_W^{1/2} \Lambda_W^{1/2} U_W'$, where Λ_W is the diagonal matrix of eigenvalues of Σ_W and the columns of the matrix U_W contain the orthogonal eigenvectors of Σ_W. The eigenvalues $\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_N$ of Σ_W decrease quickly so we may choose a $k \ll N$ such that $\sum_{i=1}^{k} \gamma_i \gg \sum_{i=k+1}^{N} \gamma_i$ and hence $U_W \Lambda_W^{1/2} z \approx \tilde{U}_W \tilde{\Lambda}_W \tilde{z}$ for most z's not far from zero. Here $\tilde{U}_W$ is the matrix containing only the first k columns of U_W, $\tilde{\Lambda}_W$ is the matrix containing the top k rows and columns of Λ_W, $z = (z_1, \ldots, z_N)$, and $\tilde{z} = (z_1, \ldots, z_k)$. Since $Z(1) = C^{-1}W(1)$, this motivates using the linear transformation with $M = C^{-1}\tilde{U}_W \tilde{\Lambda}_W$.

Another class of $h(\cdot)$'s corresponds to representing components of the drift vector z (corresponding to each factor) by a low order polynomials (with variable coefficients) in the indices. One then optimizes over the coefficients of the polynomial.

4.2 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. We now describe an approach to approximate v_1 in a k-dimensional subspace, $k \ll n$ (and then transform back to the n-dimensional subspace). We start by choosing an $n \times m$ matrix 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 M to build an n-vector from an m-vector by linear interpolation, as was done in Section 4.1 for the optimal drift. As in Section 4.1, let $\bar{z}$ denote an element of $\mathbf{R}^k$ and z an element of $\mathbf{R}^n$. The function $F_M(\bar{z}) \triangleq F(M\bar{z})$ has Hessian $H_M \triangleq M'HM$, where H the Hessian of F at $M\bar{z}$. 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 $(M'M)^{-1}M'HM$, according to the criterion in (9). Our candidate approximation for v_1 is then $M\bar{v}_1$. This procedure is exact if F happens to depend on z only through $M'z$. Moreover, any eigenvector of H that lies in the range of M is recovered by this procedure along with its eigenvalue. For if $v_i = Mx$ (v_i is now the ith eigenvector of H, ranked according to the criterion in (9)) for some x, then $(M'M)^{-1}M'HMx = \lambda_i x$, and x will be among the $\bar{v}_i$ (the ith eigenvector of the reduced matrix ranked according to the criterion in (9)), up to a scalar multiple. Finding the eigenvalues and eigenvectors of $(M'M)^{-1}H_M$ is a "generalized eigenproblem" as described on p.462 of [10]: 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.

Figure 7 displays the best eigenvector and approximate best eigenvector for the caplet problem with $T = 10$; the legend again 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.

5 Numerical Results

In this section we report on the results of numerical experiments testing the effectiveness of the method. Our primary measure of effectiveness 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 speedup 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 M replications, the resulting variance is σ^2/M. Since a simulation of M_1 replications using the variance reduction technique results in a variance of σ_1^2/M_1, to achieve the same variance implies setting $M_1 = (\sigma_1^2/\sigma^2)M$. Thus, to achieve the same variance, standard simulation requires a factor of $M/M_1 = (\sigma^2/\sigma_1^2)$ times as many replications as the variance reduction technique. Whether or not this potential savings in run length can actually be achieved in practice depends on many factors including, e.g., the desired accuracy of resulting estimate. The speedup factor does not include the setup cost (optimization overhead) of the method, which will be discussed in Section 5.3, nor 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 results in this paper are based on a total of 50,000 replications (paths) per method. We also include results for a straightforward and and more commonly applied variance reduction technique, antithetic sampling [3]. 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 were used ($M = 100$ in step 2 of Figure 1).

The organization of the rest of this section is as follows. 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. Examples of such payoffs are flex caps and options with certain trigger events, such as a down and in put option on a bond. Finally, we address the overhead of the method and the performance of techniques to reduce that overhead.

5.1 Options with a Continuous Payoff Function

The first payoff function we consider is a caplet, as defined in Section 3.2. Note that since the forward rates are continuous functions of the Z's, the discounted caplet payoff function $D_{N+1}C_{N+1}$ is also a continuous function of the Z's. Furthermore it is twice differentiable, except at the points where $\exp\{\Delta t F(N,N)\} - 1 - K\Delta t = 0$ (see (15)). Table 1 displays the estimated variance ratios for caplets with a range of payoff times T ($T = (N+1)\Delta t$ in (15)) 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 this table, we observe that:

1. The effectiveness of antithetics decreases as K increases (with 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., [6]). 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.

T	K	Anti-thetics	IS	IS & Strat. (μ)	IS & Strat. (v_1)
2.5	0.04	8.0	8.1	246	248
	0.07	1.0	16	510	444
	0.10	0.8	173	3067	2861
5.0	0.04	4.2	8.1	188	211
	0.07	1.3	11	241	292
	0.10	1.0	27	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	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

Table 1: Estimated variance ratios for caplets in 3 factor HJM model.

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, respectively. Then $T_0 = N_0\Delta t$ and $T_1 = N_1\Delta t$ for some integers N_0 and N_1. With D_i and C_i as defined above, the cap has payoff function

$$G(Z) = \sum_{i=N_0}^{N_1} D_i C_i. \tag{17}$$

This is also a continuous function of the Gaussian increments. Estimated variance ratios are listed in Table 2. 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 typically not as large 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.

T_0	T_1	K	Anti-thetics	IS	IS & Strat. (μ)	IS & Strat. (v_1)
0.25	2.5	0.04	21	5.3	20	19
		0.07	1.1	23	158	161
		0.10	1.1	285	1435	1384
0.25	5.0	0.04	8.0	5.2	23	21
		0.07	1.2	13	54	48
		0.10	0.9	41	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	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	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

Table 2: Estimated variance ratios for caps in 3 factor HJM model.

We next consider 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 the principal (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

$$B_C(N_E,N)=\sum_{i=N_E}^{N_E+N}B(N_E,i)P_i \tag{18}$$

where $B(N_E,i)$ is defined in (12). 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

$$G(Z)=D_{N_E}\max(0,100-B_C(N_E,N)). \tag{19}$$

This again is a continuous function of the underlying Z's. Estimated variance ratios are given in Table 3. 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.

The final example we consider in this subsection 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

T_E	T	C (%)	Anti-thetics	IS	IS & Strat. (μ)	IS & Strat. (v_1)
1	5	5	1.2	12	218	231
		6	2.7	6.3	205	207
1	10	5	1.1	18	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	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	183	205
		6	1.7	7.5	154	182

Table 3: Estimated variance ratios for swaptions in 3 factor HJM model.

time $T_E(= N_E \Delta t)$. Define $Y_S(i) = \sum_{j=0}^{N_S-1} F(i, i+j)/N_S$ and $Y_L(i) = \sum_{j=0}^{N_L-1} F(i,i+j)/N_L$: these are the yields on a short term (respectively, long term) zero coupon bond 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 is $Y_L(N_E) - Y_S(N_E)$. If the strike is K times the current spread, then the discounted payoff is

$$G(Z) = D_{N_E}(Y_L(N_E) - Y_S(N_E) - K\delta)^+ \times 100. \tag{20}$$

For this study we fix $T_S = 3$ years and $T_L = 15$ years. The results, which are shown in Table 4, are similar to the previous cases: large variance reductions which 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.

T_E	K	Anti-thetics	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

Table 4: Estimated variance ratios for option on zero coupon bond yield spread in 3 factor HJM model. The long term bond maturity is $T_L = 15$ years, the short term bond maturity is $T_S = 3$ years, and the strike is K times the current spread.

5.2 Options with a Discontinuous Payoff Function

The methodology outlined in this paper is based upon theory developed in GHS which assumes, among other things, that the payoff function G is a continuous function of the increments Z. However, many types of options arise in practice for which this is not true. While IS and stratification are still valid techniques in this situation, the asymptotic large deviations theoretical justification for the 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 nonlinear programming packages becomes dicey, at best; such packages often assume the existence of gradients. Since the approach above 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. There is clearly much room for experimentation here, and we report on the results of approaches that worked best from among those tested.

We first consider a flex cap in which the holder receives the payoffs of the first M caplets to expire in-the-money over N periods, $M < N$. 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 ith caplet. Let N_M denote the (random) index of the Mth caplet to expire in-the-money — that is, the Mth caplet for which $C_i > 0$. Then the flex cap's discounted payoff function is

$$G(Z) = \sum_{i=1}^{\min(N_M, N)} D_i C_i. \tag{21}$$

This is clearly a discontinuous function of Z, since small changes in Z can change the integer-valued variable N_M. To approximate this G, we select a $\hat{G}$ which 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 M payments will be positive, suggesting that we should set $N_0 = 1$ and $N_1 = M$. This is the approach we followed, except we found that setting $N_1 = M + 1$ produces somewhat better results. For a deep out-the-money flex cap, we expect that if any payoffs are positive, then with high probability, they will be from among the last M payments. This suggests that we should set $N_0 = N - M$ and $N_1 = N$ (the approach we followed). 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 $\max(N, L + M)$. In this case, to account for the variation in the time of the first and last payments, the optimizing cap used $N_0 = L - 1$ and $N_1 = \max(N, L + M + 1)$.

The results are given in Table 5. For each T there are three strikes K, the first being in-the-money, the second being the K for which the caplet paying at time $T/2$ is at-the-money, and the third being a deep out-of-the-money strike. Then, for each pair of T and K, we set the maximum number of payoffs M 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 $M = 0.25N$, but improve as M increases. For small values of M, 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 M increases.

T	K	M	Anti-thetics	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	28	20
		4	1.0	26	150	144
		6	1.0	31	199	192
		8	1.0	30	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	74	66
		15	1.1	20	133	128
		20	1.1	19	105	88

Table 5: Estimated variance ratios for flex caps in 3 factor HJM model. The maximum number of payouts is M.

We now turn to an option with a trigger event — a simplified trigger swap. Specifically, we consider an option to swap a single fixed payment for a floating payment provided the short rate drops below a trigger level before the option excercise date. (This can also be viewed as a down-and-in put on a zero coupon bond.) We denote by T_E the option exercise time and by T the date at which the payments are to be swapped. Given a trigger value F and a set of trigger dates S, the option knocks in if $F(i,i) \leq F$ for some $i \in S$. If S contains more than one value of i, the payoff function is discontinuous.

Suppose $S = \{1, \ldots, m\}$ for some $m > 1$. To approach such a problem, we solve a constrained optimization optimization problem where the constraint set takes the form $E_i = \{F(i,i) \leq F\}$ for each $i \in S$. We then pick the i with largest optimal objective function value and use its corresponding drift vector for estimating the option with trigger set S (or constraint set $A_m = \{F(j,j) \leq F \text{ for some } j, 1 \leq j \leq m\}$).

The results of this approach are given in Table 6. The table lists the optimal value of $p(z)G(z)$ (with $p(z) = \exp(-\frac{1}{2}z'z)$) over the constraint set. For the rows labeled E_i, the variance ratios are for the option with the same constraint set, i.e., the option for which the trigger event is constrained to occur exactly at time $i\Delta t$. For the row labeled A_4, we pick the μ with the best objective function value from E_2, E_3, E_4 — e.g., for $F = 0.05, K = 50$, we pick the μ corresponding to E_2. The variance ratios are then for the option with the constraint set A_4. The variance ratios for antithetics range between 1.0 and 1.3, and so are not listed separately. Stratification on v_1 produces no additional benefit and so corresponding results are also omitted. The instrument becomes less likely to pay a positive value as either F decreases or K increases. The effectiveness of the method is greatly reduced, with at best an improvement factor of 6.6 when the trigger event can occur over a range of times.

F	K	Constraint Set	$p(\mu)G(\mu)$	IS	IS & Strat.(μ)
0.05	50	E_2	6.62	3.8	4.9
		E_3	6.35	3.7	5.0
		E_4	6.04	3.8	4.4
		A_4		2.3	2.2
0.05	45	E_2	3.99	5.4	7.1
		E_3	3.74	5.5	6.7
		E_4	3.46	5.8	7.5
		A_4		3.7	4.1
0.045	50	E_2	0.84	31	41
		E_3	1.71	14	19
		E_4	2.16	11	14
		A_4		4.2	5.3
0.045	45	E_2	0.45	48	60
		E_3	0.91	24	30
		E_4	1.10	19	21
		A_4		5.8	6.6

Table 6: Estimated variance ratios for down and in put option on bond in 3 factor HJM model. $T = 15.0, T_E = 2.5$,

5.3 Overhead and Approximate Optimization

We used the GRG2 optimization package from Optimal Methods Inc., which performed satisfactorily for all of the continuous payoff functions considered in this section. While convergence to a suboptimal local maximum sometimes occurred when initialized at a random starting point, good results were obtained when the optimization was started with all components equal to 0 and the optimization was preceded by a "Phase I" step to find a non-zero payoff. For example, in a cap, this was 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 occurred in between 5 to 10 times n paths (evaluations of $G(z)$), where n is the number of variables. The constrained optimization implicit in the down and in bond put took more paths, and often would not converge when initialized to a random starting point.

To compute the Hessian matrix, we used a central differencing scheme as described on page 187 of [10]. To compute the Hessian with this approach requires about $2n^2$ paths. To reduce both the optimization and Hessian overheads, we applied the approaches described in Section 4. We experimented with the variety of approaches mentioned in that section, but we obtained the most consistent results using the method of linear interpolation.

Table 7 lists the results when applying this approach to caplets. The table lists 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 (the column labeled Variance Ratio (v_1)). We observe dramatic savings in overhead with little loss in variance reduction. For example, with $T = 10.0$, the full problem has 117 ($= 3 \times 39$) variables. With the Linear(3) method, the number of variables is reduced to 9 and the number of optimization paths is reduced by about a factor of 10 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 Table 8.

Results using approximate optimization based on the principal components as described in Sections 4.1 and 4.2 produced significantly worse results for a comparable number of independent variables. For example, for the caplet in Table 7 with $T = 10$, using the first 4 principal components yielded a variance ratio of 26 (34) when stratfiying on μ (v_1) as opposed to a variance ratio of 75 (156) for the Linear(4) method.

T	Method	Opt. 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

Table 7: Overheads and estimated variance ratios using approximate optimization for caplets in 3 factor HJM model. The strike is $K = 0.07$.

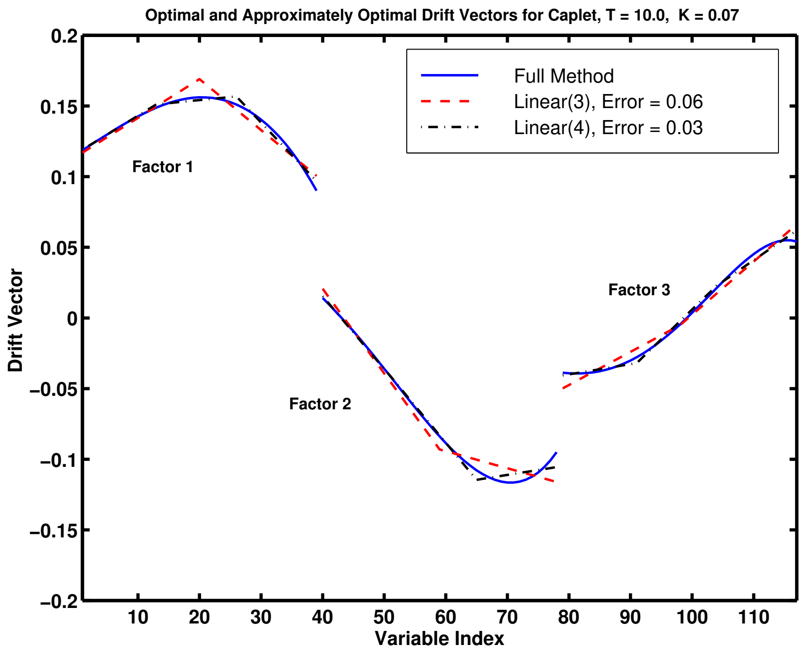

Figure 6: Optimal and approximately optimal drift vectors for a caplet with $T = 10.0$ and $K = 0.07$. The error listed is the norm of the difference between the optimal and approximately optimal drift vectors.

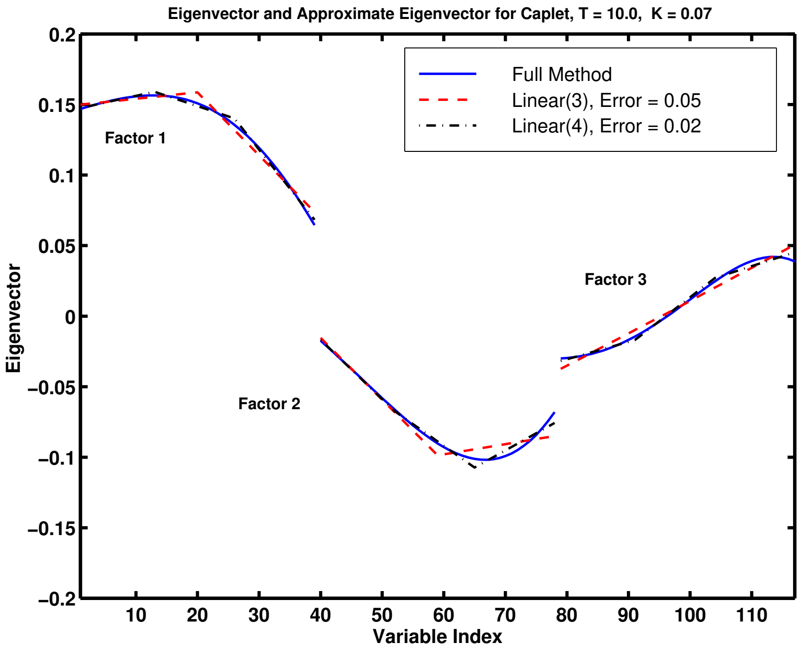

Figure 7: Eigenvector and approximate eigenvector for a caplet with $T = 10.0$ and $K = 0.07$. The error listed is the norm of the difference between the eigenvector and approximate eigenvector.

T_E	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

Table 8: Overheads and estimated variance ratios using approximate optimization for yield spread option in 3 factor HJM Model. The long term bond maturity is 15 years, the short term bond maturity is 3 years, and the strike is $K = 2$ times the current spread.

6 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 upon 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 of the potential variance reduction can be achieved with greatly reduced overheads.

References

[1] Boyle, P., M. Broadie, and P. Glasserman, "Simulation Methods for Security Pricing," *J. Economic Dynamics and Control*, 21, 1267-1321, 1997.

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

[3] Hammersley, J., and D. Handscomb, *Monte Carlo Methods.* Methuen & Co. Ltd., London, 1964.

[4] Glasserman, P., P. Heidelberger, and P. Shahabuddin, "Asymptotically Optimal Importance Sampling and Stratification for Pricing Path-Dependent Options," *Math. Finance*, 9, no. 2, 1999 (to appear).

[5] Heath, D., Jarrow, R., and Morton, A., "Bond pricing and the Term Structure of Interest Rates: a New Methodology for Contingent Claims Valuation," *Econometrica*, 60, 77-105 , 1992.

[6] Heidelberger, P., "Fast Simulation of Rare Events in Queueing and Reliability Models," *ACM Trans. Modeling and Computer Simulation*, 5, no. 1, 43–85, 1995.

[7] Marsaglia, G., A. Zaman, and J. Marsaglia, "Rapid Evaluation of the Inverse of the Normal Distribution Function," *Statistics and Probability Letters*, **19**, 259–266, 1994.

[8] Moro, B., The Full Monte, *RISK*, **8** (February), 57–58, 1995.

[9] Newton, N.J., "Continuous-Time Monte Carlo Methods and Variance Reduction," in *Numerical Methods in Finance*, L.C.G. Rogers and D. Talay, eds., Cambridge University Press, 22–42, 1997.

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

[11] Schoenmakers, J.G., and A.W. Heemink, "Fast Valuation of Financial Derivatives," *J. Computational Finance*, **1**, 47–62, 1997.