Just-In-Time Monte Carlo for Path-Dependent American Options

SAMIR K. DUTT AND GERD M. WELKE

SAMIR K. DUTT
is an associate professor of finance at the Orfalea College of Business, California Polytechnic State University in San Luis Obispo, CA, and a consultant to Quantal International, Inc.
sdutt@calpoly.edu

GERD M. WELKE
is an assistant professor in the Department of Real Estate at Zicklin School of Business, Baruch College/CUNY in New York, NY.
gerd_welke@baruch.cuny.edu

We establish simple analytical and numerical methods for propagating stochastic price processes backwards *in time, step by step, to the initial value while satisfying all cross-sectional and serial requirements. This proves useful in dealing with complex path-dependent options with American triggers, where storing the history of the underlying can become computationally onerous. Examples involving the Wiener, Ornstein-Uhlenbeck, Clark, and Cox-Ingersoll-Ross processes illustrate our techniques. Our "just-in-time" method, which can be thought of as stochastic* involution, *extends the reach and accuracy of Monte Carlo pricing techniques.*

The extension of Monte Carlo pricing methods to American options by Longstaff and Schwartz [2001] has greatly simplified the pricing of complex path-dependent options with early exercise features. Given the widespread use of such options in "all major financial markets including the equity, commodity, foreign exchange, insurance, energy, sovereign, agency, municipal, mortgage, credit, real estate, convertible, swap, and emerging markets (ibid.)," and the ease with which general stochastic processes can be accommodated, the least squares Monte Carlo (LSM) method has been rapidly adopted by practitioners and academics alike.[1]

A practical difficulty that arises in applications of the LSM method is the need for storing the underlying price process. For example, a one-year vanilla American put option evaluated with 50 time steps (early exercise dates) and 100,000 paths requires at least 40 megabytes (MB) of storage, while a 30-year interest rate product evaluated with a weekly time step and a million paths requires more than 12 gigabytes (GB) of storage, putting it quite beyond the realm of any 32-bit operating system. By contrast, the binomial method of Cox-Ross-Rubinstein [1997] requires a mere 408 bytes to store the 51 terminal price points for the vanilla put of our example. Earlier price points required for the backward pricing algorithm can be generated on the fly and require even less by way of storage.[2] This "just-in-time" feature of the CRR algorithm is key to its speed and low storage requirements and invites imitation.[3]

Computational efficiency is not the only or even the most important reason for seeking to generate stochastic processes in reverse. Very few price processes of interest to financial economists can be generated (integrated) exactly. The Wiener and Ornstein-Uhlenbeck processes can be generated without approximation, but the Cox-Ingersoll-Ross process, of central importance in pricing fixed-income products, can only be generated by approximations like the stochastic Euler scheme. This necessarily introduces errors of a non-statistical character in Monte Carlo simulations of option values. Furthermore, these errors compound into the future, where the value of any option, by its very nature as a bet on the future, resides. Yet, in many cases of interest, an

analytical expression for the transition density exists, so that the final distribution of the process can, in fact, be generated with no errors other than sample size effects. If one could use the final distribution as the *input* and work out the underlying stochastic process backwards as one prices the option, three desirable objectives would be achieved: 1) the backward option pricing algorithm would be married to a backward price process algorithm, 2) storage requirements would be greatly reduced, and 3) statistical and systematic errors could be ameliorated by the ability to "turn on the tap" for the number of paths used in Monte Carlo simulations. This article shows how to achieve these objectives.

To this end, we first show that the Wiener process (and thus the lognormal price process) is easily generated in reverse.[4] We then extend our technique to the Ornstein-Uhlenbeck process by expressing it as a subordinated Wiener process with a deterministic driving process. The next section takes up the important question of whether more general stochastic processes like the Cox-Ingersoll-Ross process can be reversed. We first discuss the possibility of achieving reversal as in the case of the Ornstein-Uhlenbeck process, but with the use of a stochastic driving process. Theoretical and practical difficulties inherent in this approach are examined, and it is shown that, as a purely practical matter, the use of pseudo-random numbers in digital simulations provides a very simple technique for generating some processes in reverse. A different approach—the theory of time-reversed Itô diffusions—is used to obtain a diffusion equation for the reverse CIR process. We present examples, applications, and numerical issues.

Finally, we anticipate some of our results to point out when one can start with an arbitrary terminal, or intermediate, stock price distribution and work backwards to the initial condition: 1) Working backwards from an intermediate or terminal distribution is possible in all cases where the algorithm for reversal can be achieved analytically, i.e., for the Wiener, Ornstein-Uhlenbeck, and CIR processes considered in this work, and also for the Itô diffusions we consider. 2) The numerical technique developed for very general stochastic processes requires that we first generate the process going forward, storing only the random seeds used at every time step, and the terminal distribution. When this process is run in reverse using either the terminal distribution (or a previously stored intermediate distribution) the use of stored seeds dictates the involution of the process and will generate only the original forward process in reverse. Apart from sample size considerations, there are no other limitations on the terminal distribution. One can use a terminal distribution that is small (even a single point), or very large (millions of points), and work backwards to the unique initial point from which the stochastic process evolved.

GAUSSIAN PROCESSES

The Wiener Process

It is useful to recall the simple method by which Brownian motion is generated in discrete time. Let $\tilde{z}_1$, $\tilde{z}_2$, $\tilde{z}_3$,... be mutually independent and identical Gaussian random variables, $\tilde{z}_i \sim N(0, 1)$. Further, let z_1, z_2, z_3,... be independent draws from $\tilde{z}_1$, $\tilde{z}_2$, $\tilde{z}_3$,... We can construct a sample path ω of the Brownian process $\tilde{W}_t$ according to

$$\tilde{W}_0(\omega) = 0,\ \tilde{W}_1(\omega) = z_1,\ \tilde{W}_2(\omega) = z_1 + z_2,$$
$$\tilde{W}_3(\omega) = z_1 + z_2 + z_3,\ldots$$

The collection of all possible paths can be defined recursively by

$$\tilde{W}_0 = 0,\ \tilde{W}_i = \tilde{W}_{i-1} + \tilde{z}_i,\ i = 1, 2, 3,\ldots \qquad (1)$$

As defined in Equation (1), the stochastic process $\tilde{W}_i$ is a discrete Wiener process because it satisfies 1) $\tilde{W}_0 = 0$, 2) $\tilde{W}_j - \tilde{W}_i \sim N(0, j - i)$, for all $0 \le i \le j$, and 3) non-overlapping increments are independent, that is for all $i < j < k$, $\tilde{W}_k - \tilde{W}_j$ and $\tilde{W}_j - \tilde{W}_i$ are independent. A fourth property, that of continuity, is meaningful only in continuous time.[5] These four properties are necessary and sufficient for a stochastic process to be a Wiener process.

The link between a Wiener process and a lognormal price process with initial price S_0, volatility σ, drift μ, and time step Δt is given by

$$\tilde{S}_i = S_0 \times e^{(\mu - \sigma^2/2)\Delta t \times i} \times e^{\sigma\sqrt{\Delta t}\,\tilde{W}_i},\quad i = 1, 2, 3,\ldots \qquad (2a)$$

or, equivalently, by

$$\tilde{S}_{t_i} = S_0 \times e^{(\mu - \sigma^2/2)t_i} \times e^{\sigma\tilde{W}_{t_i}},\quad i = 1, 2, 3,\ldots \qquad (2b)$$

where we have used the scaling property $\sqrt{\Delta t}\,\tilde{W} \stackrel{d}{=} \tilde{W}_{\Delta t \times i}$, and $t_i \equiv \Delta t \times i$. Note that the first factor on the RHS of Equation (2) is the initial stock price S_0, the second factor is a deterministic boost, and the final factor is an exponentiated Wiener process. Any lognormal price process can be generated in this way once the appropriate Wiener process is in hand. Since it suffices to deal with the Wiener process we shall work directly with $\tilde{W}_i$ in what follows.

A key property of a Wiener process is the unit increase in cross-sectional variance as we advance one unit in time step i. The reason for this can be seen in the iterative definition of $\tilde{W}_i$ in Equation (1): $\tilde{W}_{i-1}$ and $\tilde{z}_i$ enter into the expression for $\tilde{W}_i$ with equal weights, $\tilde{W}_{i-1} \sim N(0, i-1)$, $\tilde{z}_i \sim N(0, 1)$, and $\tilde{W}_{i-1}$ and $\tilde{z}_i$ are independent and, therefore, uncorrelated. The variance of their sum must equal the sum of their variances. Hence, $\tilde{W}_i \sim N(0, i)$. This motivates us to look for a similar construction, but with weights chosen so that the resulting random variable has an appropriately *lower* variance, allowing for the interpretation that one is now moving in reverse. Thus, let $\tilde{X} \sim N(0, i)$, $\tilde{x} \sim N(0, 1)$ be independent Gaussian random variables, and let $\tilde{Y} \equiv a\tilde{X} + b\tilde{x}$. $\tilde{Y}$ is obviously a mean zero Gaussian random variable. We look for weights a, b such that the variance of $\tilde{Y}$ is *reduced* by 1, so $\tilde{Y} \sim N(0, i-1)$. It is straightforward to show that $\tilde{Y} = (1 - 1/i)\tilde{X} + \sqrt{1-1/i}\,\tilde{x}$ works. We formalize this simple intuition in

Definition 1. *For any positive integer n, let $\tilde{W}_n$ be a Gaussian random variable such that $\tilde{W}_n \sim N(0, n)$. Define $\tilde{W}_0$, $\tilde{W}_1, \ldots, \tilde{W}_{n-1}$ by the backward recursion*

$$\tilde{W}_{i-1} \equiv \left(1 - \frac{1}{i}\right)\tilde{W}_i + \sqrt{1 - \frac{1}{i}}\,\tilde{z}_i, \quad i = 1, 2, \ldots, n \qquad (3)$$

where $\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_n$ are Gaussian random variables such that $\tilde{z}_i \sim N(0, 1)$, and $\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_n, \tilde{W}_n$ are mutually independent.

Proposition 1. *For every $i = 0, 1, \ldots, n$ the random variable $\tilde{W}_i$ defined by the backward recursion in (3) is Gaussian and satisfies*

$$\tilde{W}_i \sim N(0, i) \qquad (4)$$

Proof: All proofs are shown in Appendix A.

Remark: Equation (3) sets up a backward recursion for the random variables $\tilde{W}_0$, $\tilde{W}_1$, $\tilde{W}_2, \ldots, \tilde{W}_{n-1}$ starting from $\tilde{W}_n \sim N(0, n)$, where n is any positive integer. We first compute $\tilde{W}_{n-1}$ from $\tilde{W}_n \sim N(0, n)$ and $\tilde{z}_n \sim N(0, 1)$ to get

$$\tilde{W}_{n-1} = \left(1 - \frac{1}{n}\right)\tilde{W}_n + \sqrt{1 - \frac{1}{n}}\,\tilde{z}_n$$

$\tilde{W}_n$ and $\tilde{z}_n$ being independent. Next, we compute $\tilde{W}_{n-2}$ using $\tilde{W}_{n-1}$ from the expression above and $\tilde{z}_{n-1} \sim N(0, 1)$

$$\tilde{W}_{n-2} = \left(1 - \frac{1}{n-1}\right)\tilde{W}_{n-1} + \sqrt{1 - \frac{1}{n-1}}\,\tilde{z}_{n-1} \qquad (5)$$

with $\tilde{W}_n$, $\tilde{z}_n$ and $\tilde{z}_{n-1}$ being mutually independent. The iteration ends at $\tilde{W}_0$, which is easily seen to be 0.[6] Note that while n can be any positive integer, the backward recursion constructs $\tilde{W}_i$ relative to the choice of the terminal time step n. $\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_n, \tilde{W}_n$ are chosen to be mutually independent Gaussian random variables, which implies, for example, that $\tilde{W}_i$ and $\tilde{z}_i$ are independent for all $i = 1, 2, 3, \ldots, n$, but that $\tilde{W}_3$ and $\tilde{W}_8$ are not independent since $\tilde{W}_3$ can be written as a linear combination of the random variables $\tilde{z}_4, \tilde{z}_5, \ldots, \tilde{z}_8$ and $\tilde{W}_8$.

Proposition 2. *The stochastic process $\tilde{W}_0, \tilde{W}_1, \ldots, \tilde{W}_n$ defined by the backward recursion in Equation (3) is a discrete Wiener process. In particular, it satisfies:*

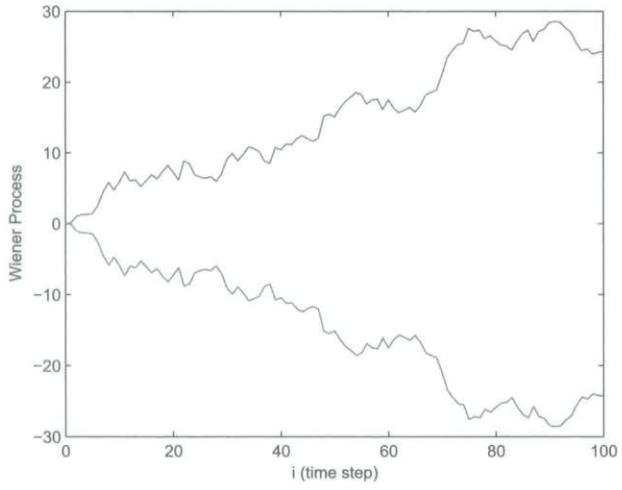
i. $\tilde{W}_0 = 0$*: the process starts at 0.*

ii. $\tilde{W}_j - \tilde{W}_i \sim N(0, j - i)$*, for all $0 \le i \le j \le n$: all marginals are normally distributed with mean zero and variance equal to the time difference between the relevant epochs.*

iii. $cov(\tilde{W}_k - \tilde{W}_j, \tilde{W}_j - \tilde{W}_i) = 0$*, for all $i < j < k \le n$: all non-overlapping increments are independent.*

We shall refer to the backward recursion algorithm for a Wiener process as the "just-in-time" method. Exhibit 1 displays the results from simulating a path and its antithetic complement backwards using the just-in-time method. Matlab code for implementing the just-in-time method for a Wiener process is provided in the appendix, along with pseudo-code-like comments. The method is exceedingly parsimonious, requiring only about six lines of Matlab code for implementation. We also provide Matlab code for calculating the value of an American put using the LSM method, with the stock price process

EXHIBIT 1
Weiner Process



Note: Weiner Process: Two paths (1 + 1 antithetic) generated backwards by the just-in-time method. Inclusion of the antithetic path in Monte Carlo simulations is customary to ensure that the cross-sectional means of the simulated Wiener process be zero in sample.

computed using the just-in-time method. We use 1,000,000 paths to obtain accuracy to four significant places. Details may be looked up in the appendix.

Remark: For the relationship between the backward recursion procedure in Equation (3) and a Brownian bridge, see the remarks consequent upon Equation (20).

Remark: We apply the method outlined here in the applications section to value a deep-out-of-the-money American put and show that the LSM method can lead to *overestimation* of the option value despite the fact that the LSM method necessarily uses a suboptimal stopping rule. The ability to generate a very large number of paths by the just-in-time method proves crucial in resolving this anomaly. Readers may wish to peruse the applications section first for a practical application of the just-in-time method, and then return to the intervening sections, where we develop methods for reversing general stochastic processes starting with the example of the Ornstein-Uhlenbeck process.

The Ornstein-Uhlenbeck Process

The Ornstein-Uhlenbeck (OU) process

$$d\tilde{r}_t = -\kappa(\tilde{r}_t - \theta)dt + \sigma d\tilde{w}_t, \quad \tilde{r}_0 = r_0 \tag{6}$$

with $\kappa > 0$ is a mean-reverting process used originally in Vasicek [1977] to model the equilibrium short rate. The OU process also finds use in modeling commodity prices and other economic variables that tend to fluctuate around a long-term mean (e.g., Dixit and Pindyck [1994]). $\tilde{r}_t$ can be solved for explicitly by making the transformation $\tilde{y}_t \equiv e^{\kappa t}(\tilde{r}_t - \theta)$ to get the simple SDE $d\tilde{y}_t = e^{\kappa t} d\tilde{w}_t$, leading to

$$\tilde{r}_t = r_0 e^{-\kappa t} + \theta(1 - e^{-\kappa t}) + \sigma e^{-\kappa t} \int_0^t e^{\kappa s} d\tilde{w}_s \tag{7}$$

$\tilde{r}_t$ is seen to be a normally distributed random variable. The conditional mean μ_t and variance σ_t^2 of $\tilde{r}_t$ can be computed from Equation (7) with use of the elementary identity

$$E\left[\int_0^t g(s)d\tilde{w}_s \times \int_0^{t'} h(s)d\tilde{w}_s\right] = \int_0^{t \wedge t'} g(s)h(s)ds \tag{8}$$

to get

$$\mu_t = r_0 e^{-\kappa t} + \theta(1 - e^{-\kappa t}), \quad \sigma_t^2 = \frac{\sigma^2}{2\kappa}(1 - e^{-2\kappa t}) \tag{9}$$

It is seen easily from Equation (9) that $\mu_t \to \theta$ and $\sigma_t^2 \to \sigma^2/2\kappa$ as $t \to \infty$: in the asymptotic limit, the OU process becomes a constant variance Gaussian process fluctuating about its long-term mean of θ. The rate at which the asymptotic limit is approached is controlled by the "spring constant" parameter κ. We proceed to developing the time-reversed OU process.

It proves convenient to rewrite Equation (7) in terms of a time-changed Wiener process thus:

Proposition 3. *The Itô integral $\int_0^t e^{\kappa s} d\tilde{w}_s$ is a time-changed Wiener process according to*

$$\int_0^t e^{\kappa s} d\tilde{w}_s = \tilde{W}_{\tau(t)}, \quad \tau(t) \equiv \frac{e^{2\kappa t} - 1}{2\kappa} \tag{10}$$

Note that $\tau(0) = 0$, and that $\tau(t)$ is a monotonically increasing function of t. Equation (10) allows us to rewrite $\tilde{r}_t$ as

$$\tilde{r}_t = r_0 e^{-\kappa t} + \theta(1 - e^{-\kappa t}) + \sigma e^{-\kappa t} \tilde{W}_{\tau(t)} \tag{11}$$

Consider, now, the problem of generating the OU process in reverse. We see from the RHS of Equation (11) that the first two terms of $\tilde{r}_t$ constitute a deterministic shift, while the third term is proportional to a Wiener process at time $\tau(t)$. Given that we wish to generate $\tilde{r}_t$ in reverse at times $0 < t_1 < t_2 < \cdots < t_n$, we only need to generate $\tilde{W}_0$, $\tilde{W}_{\tau_1}$, $\tilde{W}_{\tau_2}$,..., $\tilde{W}_{\tau_n}$ in reverse, $\tau_i \equiv \tau(t_i)$, $\tau_0 = t_0 \equiv 0$. This can be done by generalizing the unit-time-step backward recursion defined in Equation (3) to arbitrary time steps.

Definition 2. *Let* $0 < \tau_1 < \tau_2 < \cdots < \tau_n$, and let $\tilde{z}_1$, $\tilde{z}_2$,..., $\tilde{z}_n$, $\tilde{W}_{\tau n}$ *be mutually independent Gaussian random variables normed by* $\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_n \sim N(0, 1)$, *and* $\tilde{W}_{\tau_n} \sim N(0, \tau_n)$. *Define* $\tilde{W}_0$, $\tilde{W}_{\tau_1}$, $\tilde{W}_{\tau_2}$,..., $\tilde{W}_{\tau_{n-1}}$ *by the backward recursion*

$$\tilde{W}_{\tau_{i-1}} \equiv \frac{\tau_{i-1}}{\tau_i}\tilde{W}_{\tau_i} + \sqrt{\frac{\tau_{i-1}}{\tau_i} \times (\tau_i - \tau_{i-1})}\,\tilde{z}_i, \quad i = 1, 2, \ldots, n \tag{12}$$

Remark: We can recover the unit-time-step backward recursion is Equation (3) from the general expression in Equation (12) above by setting $\tau_i = i$. Given $\tilde{W}_{\tau_n} \sim N(0, \tau_n)$, we first compute $\tilde{W}_{\tau_{n-1}}$ from the prescription above, next $\tilde{W}_{\tau_{n-2}}$ from $\tilde{W}_{\tau_{n-1}}$, and so on until we reach $\tilde{W}_0 = 0$. The recursive construction is vital to creating a Wiener process, for it embeds the requirement of continuity in the limit of infinitesimal time step. To appreciate this, consider the process $\tilde{X}_t \equiv \sqrt{t}\tilde{z}_t$, where the mutually independent random variables $\tilde{z}_t$ are normed by $N(0, 1)$ for all t. $\tilde{X}_t$ is distributed as $N(0, t)$ for all t, but it is not a Wiener process. The non-recursive construction of $\tilde{X}_t$ fails to ensure continuity in the limit, so $\tilde{X}_t$ fails to satisfy the self-similar properties of Brownian motion.

Proposition 4. *The stochastic process* $\tilde{W}_0$, $\tilde{W}_{\tau_1}$, $\tilde{W}_{\tau_2}$,..., $\tilde{W}_{\tau_n}$ *defined by the backward recursion in Equation (12) is a discrete Wiener process. In particular, it satisfies:*

i. $\tilde{W}_0 = 0$: *the process starts at 0.*

ii. $\tilde{W}_{\tau_j} - \tilde{W}_{\tau_i} \sim N(0, \tau_j - \tau_i)$, *for all* $0 \le i \le j \le n$: *all marginals are Gaussian random variables with mean zero and variance equal to the time difference between the relevant epochs.*

iii. $cov(\tilde{W}_{\tau_k} - \tilde{W}_{\tau_j}, \tilde{W}_{\tau_j} - \tilde{W}_{\tau_i}) = 0$, *for all* $i < j < k \le n$: *all non-overlapping increments are independent.*

Exhibit 2 displays an OU path and its antithetic complement generated backwards by the just-in-time method.

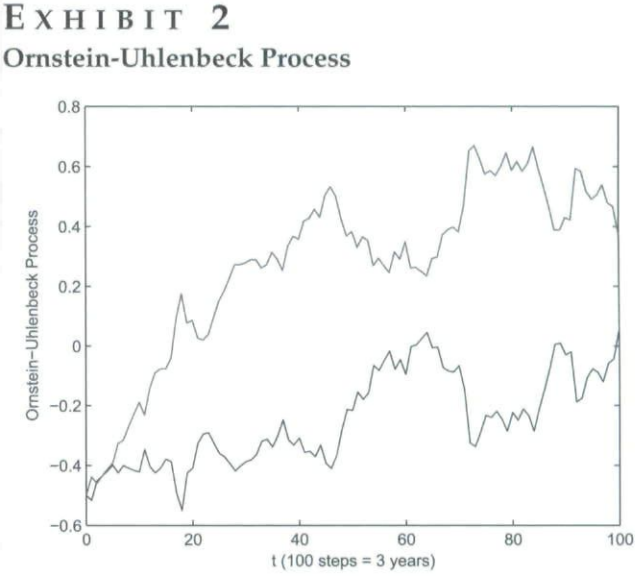
Remark: The extension of the methods developed above for Gaussian processes to more than one dimension (multiple Wiener or OU processes) is straightforward. Given a stochastic process driven by more than one Wiener process, the essential step is to first express the stochastic process in terms of independent Wiener processes. This is easily accomplished (with use of the associated variance-covariance matrix for the underlying Wiener processes) by Gramm-Schmidt orthonormalization.

GENERAL PROCESSES

Subordinated Processes

The treatment of the Ornstein-Uhlenbeck process in the preceding section indicates one possible way in which an arbitrary price process may be generated in reverse. It is well known that the log of any price process must be a semi-martingale in order to satisfy the "no arbitrage" requirement (Delbaen and Schachermayer [1994]) and also that any continuous semi-martingale can be expressed as a Wiener process evaluated at a stochastic clock time $\tau(t)$ (Monroe [1978]), where t is the calendar time.

EXHIBIT 2
Ornstein-Uhlenbeck Process



Note: Two paths (1 + 1 antithetic) over 100 time steps generated backwards by the just-in-time method for the parameter set $r_0 = -0.5$, $\theta = 0.25$, $k = 1$, $\sigma = 0.3$. *Note the forward movement toward the long term mean* $\theta = 0.25$.

The OU process is a simple example of a subordinated Wiener process, with, in this case, a deterministic directing process $\tau(t)$ given in (10).[7] More generally, if Δt represents the ticking of the calendar clock, $\Delta\tau(t)$ will be drawn from a positive-valued distribution and will proxy for the evolution of economic time.[8] The rationale for this view is expressed compellingly in Clark [1973]: "The different evolution of price series on different days is due to the fact that information is available to traders at a varying rate. On days when no new information is available, trading is slow, and the price process evolves slowly. On days when new information violates old expectations, trading is brisk, and the price process evolves much faster."

Our scheme for reversing the price process as in the Ornstein-Uhlenbeck case in the preceding section requires that we work out a method for inverting the directing process. It appears likely that this would entail explicit use of conditional distributions, an eventuality we were able to avoid in reversing the Wiener and Ornstein-Uhlenbeck processes.

Example. Let $t = 0, 1, 2,...$ be the calendar clock ($\Delta t = 1$), and let the increments n_t of the directing process N_t be drawn from a Poisson distribution with an arrival rate of λ per unit time. N_t is a Poisson random walk, with $N_t = N_{t-1} + n_t$, $t = 1, 2, 3,...$, where n_t is independent of N_{t-1}, and

$$\mathbf{P}\{n_t = i\} = e^{-\lambda}\frac{\lambda^i}{i!}, \quad \mathbf{P}\{N_{t-1} = m\} = e^{-\lambda(t-1)}\frac{[\lambda(t-1)]^m}{m!},$$
$$i,\ m = 0, 1, 2,...$$

If along some path we realize the value n at epoch t, we may infer that $N_{t-1} = n - i$ and $n_t = i$, $i = 0, 1, 2,..., n$. In order to recede by one step we would have to draw from the distribution of n_t conditioned on the fact that $N_t = n$ and subtract the draw from n. It is easily shown that

$$\mathbf{P}\{n_t = i \mid N_t = n\} = \frac{\mathbf{P}\{N_{t-1} = n-i, n_t = i\}}{\sum_{i=0}^{n}\mathbf{P}\{N_{t-1} = n-i, n_t = i\}},$$
$$= \binom{n}{i}\frac{[t-1]^{n-i}}{t^n}, \quad i = 0, 1, 2,..., n \quad (13)$$

where we have used the independence of N_{t-1} and n_t. Now, given a sample of $N_t \sim \text{Poisson}(\lambda t)$ at epoch t, we would reverse each realization one step by drawing from the appropriate conditional distribution of n_t and subtracting the draw. This would be computationally expensive without access to a vector processor. By contrast, the forward generation process involves only independent Poisson random variables and can therefore be accomplished in "one fell swoop."

The example above is relevant to a model considered in Geman, Madan, and Yor [2001], wherein the log price process is the difference of independent up and down moves distributed identically as compound Poisson processes with reflected normal shocks. In this particular model, the log price process may be written as $\log(p_t/p_0) = \sigma W_{N(t)}$, $N(t) \equiv N_1(t) + N_2(t)$, where $N_1(t)$, $N_2(t)$ are independent and identical simple Poisson processes with arrival rate λ per unit time. Since Poisson distributions are closed under convolution, $N(t)$ is also a Poisson process, with arrival rate 2λ per unit time, and we can use the conditional distribution in Equation (13) to generate the log price process in reverse. In general, we must be prepared to reverse subordinators from the class of infinitely divisible distributions. Geman Madan, and Yor [2001], Geman [2002], and Carr, Geman, Madan, and Yor [2003] provide many examples of such subordinators,[9] with conditional distributions that are costly to compute. This raises the question of whether it is practical to reverse a price process through its subordinator.

It turns out that there is a simple trick by which we can invert any stochastic process that is made up of independent and homogeneous increments. Some processes with non-homogeneous increments such as the Cox-Ingersoll-Ross (CIR) process can also be generated in reverse by this technique. Recall that the random numbers generated in a digital simulation are, in fact, pseudo-random, that is they can be regenerated if the seed is known. The trick is to first generate the process forward and to store only the seed at each time step. Having arrived at the final distribution, we then use the stored seeds to generate the process in reverse. The storage required depends only on the number of time steps, and not on the number of paths generated, which effectively decouples the number of paths from the number of time steps. The storage required for a single seed is 2 bytes, so a thousand steps require a mere 2 KB of seed storage, *no matter how many paths are generated.*

Example. Consider the lognormal-normal process in Clark [1973]. The increments $\Delta\tau(t)$ of the subordinator are distributed lognormally

$$\Delta\tau(t) \sim \frac{1}{\sqrt{2\pi}\,\sigma_1 x}\exp\left[-\frac{(\log x-\mu)^2}{2\sigma_1^2}\right], \quad x>0$$

and the log price process increments $\Delta Y(t)$ are given by

$$\Delta Y(t) = \sqrt{\Delta\tau(t)}\,Z(t)$$

where $Z(t) \sim N(0, \sigma_2^2)$, so that

$$\Delta Y(t) \sim \frac{1}{2\pi\sigma_1\sigma_2}\int_0^\infty \upsilon^{-3/2}\exp\left[-\frac{y^2}{2\upsilon\sigma_2^2}\right] \times \exp\left[-\frac{(\log\upsilon-\mu)^2}{2\sigma_1^2}s\right]d\upsilon, \quad -\infty<y<\infty$$

The expression above is analytically intractable, and we would be hard-pressed to reverse the process using conditional distributions at every step. Instead, we do the following: *Forward*: 1) store the current seed. 2) Draw, say, one million numbers from a lognormal distribution with parameters μ and σ_1.[10] 3) Store the seed again, and 4) draw one million numbers from $N(0, \sigma_2^2)$ (0.047 seconds). 5) Compute $\Delta Y(i) = \sqrt{\Delta\tau(i)}\,Z(i)$. 6) Add to the preceding value $Y(i-1)$. 7) Repeat for 100, or 1000, or 10,000 time steps, as desired. Note that only the new cross-sectional distribution at each time step is saved, not the history of the process. There are two seed processes in this example, one for $\Delta\tau(t)$ and one for $Z(t)$. *Reverse*: Use the saved seeds in reverse order to generate the preceding increment, and subtract from the current cross-sectional distribution at each step. These are fast and simple operations. The Longstaff-Schwartz LSM pricing algorithm is easily inserted into the reverse generation algorithm. Exhibit 3 displays two paths generated by the forward-backward algorithm for the lognormal-normal subordinated process in Clark [1973].

Remark: The forward-backward technique will reverse *only* the final distribution generated by the forward process and no other. If the increments are not time-homogeneous, as happens in the CIR process, the reverse process requires us to solve for the values of the process one step before. In the case of the CIR process this is a simple quadratic equation with all solutions lying on a single branch. The reversion is therefore easily accomplished. A more complicated SDE can require solving a transcendental equation at every time step going backwards. This is neither practical nor desirable, nor do we recommend it. A more elegant method exists in the case of Itô diffusions, as the next section shows.

Time-Reversed Itô Diffusion

We turn, now, to a technique that appears to have attracted little attention in the finance literature—the time-reversal of Itô diffusion. The problem arises naturally in the quantum theory of measurement (Schrödinger [1931]), in signal processing (Ljung and Kailath [1976]), and in population diffusion models (Nagasawa [1980]). We state a version of the result which applies to one-dimensional nonlinear diffusions.

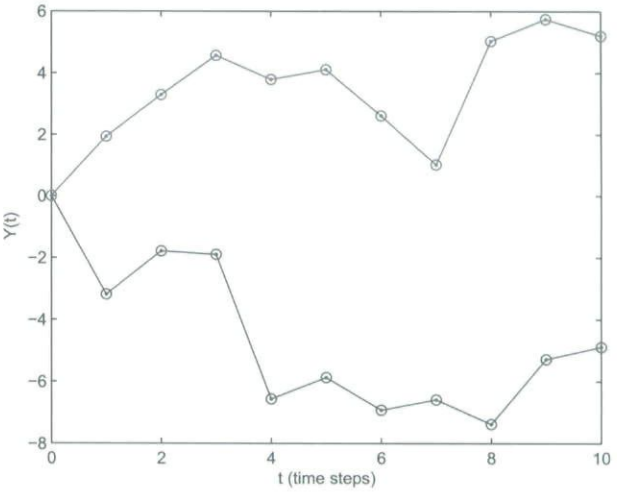
Theorem: (Anderson [1982], Haussmann and Pardoux [1986]): *If* $\{X_t: 0 \le t \le T\}$ *is a diffusion process*[11] *in* $\mathbb{R}^1$ *such that* X_t *is a strong solution of*

$$dX_t = \mu(t, X_t)dt + \sigma(t, X_t)dw_t, \quad X_0 = x_0$$

where w_t *is a 1-dimensional Wiener process in* $\mathbb{R}^1$, *then the time-reversed process* $\bar{X}_t \equiv X_{T-t}$ *is also Markov and is a weak solution of*

EXHIBIT 3

Two Paths Generated by the Forward-Backward Method



Note: Two paths for the longnormal-normal process in Clark (1973) generated by the forward-backward method over ten time steps. Dots represent forward generation and circles backward generation. Paths have been connected for ease of viewing. The parameters correspond to $\mu = 1$, $\sigma_1 = 1$, $\sigma_2 = 1$, *and* $\Delta t = 1$.

$$d\bar{X}_t = \bar{\mu}(t, \bar{X}_t)dt + \bar{\sigma}(t, \bar{X}_t)d\bar{w}_t, \quad \bar{X}_0 = X_T \quad (14a)$$

where $\bar{w}_t$ is a 1-dimensional Wiener process in $\mathbb{R}^1$ which is independent of $\bar{X}_0 = X_T$, wih

$$\bar{\mu}(t,x) = -\mu(T-t,x) + \frac{\partial_x[\sigma^2(T-t,x)p_{T-t}(x)]}{p_{T-t}(x)} \quad (14b)$$

$$\bar{\sigma}(t,x) = \sigma(T-t,x) \quad (14c)$$

and where $p_t(x)$ is the law of X_t at time t.

Note that $p_t(x)$ will depend on the initial condition $X_0 = x_0$. These equations are best examined with the aid of a simple example.

Example: We shall use the time-reversal theorem above to solve for the backward Wiener process treated in the prior section on Wiener processes. We note that the results presented in the previous section, Gaussian Processes, were obtained by the simple considerations contained therein and not from the analysis which now follows. The SDE for a Wiener process is $dX_t = dw_t$, $X_0 = 0$, and its transition density relative to the initial condition is $p_t(x) = e^{-x^2/2t}/\sqrt{2\pi t}$. Using $\mu = 0$, $\sigma = 1$, and applying Equation (14) we get for the time-reversed process $\bar{X}_t$,

$$d\bar{X}_t = -\frac{\bar{X}_t}{T-t}dt + d\bar{w}_t, \quad \bar{X}_0 = w_T \quad (15)$$

Note that w_T represents the terminal cross-sectional distribution of the forward process X_T, and is independent of $\bar{w}_t$ in Equation (15) by stipulation. To recover the result in Proposition 4, define a new process $Y_t \equiv \bar{X}_t/(T-t)$, $0 \le t < T$, so that $dY_t = d\bar{w}_t/(T-t)$ and

$$\begin{aligned} Y_t &= Y_0 + \int_0^t \frac{d\bar{w}_s}{T-s} \\ &= \frac{\bar{X}_0}{T} + \hat{w}_{\varphi(t)} \end{aligned} \quad (16)$$

where $\bar{X}_0 = X_T = w_T$, $\hat{w}_r$, $r \ge 0$, is another Wiener process, independent of w_T, (see the proof of Proposition 3) and

$$\begin{aligned} \varphi(t) &\equiv \int_0^t \frac{ds}{(T-s)^2} \\ &= \frac{t}{T(T-t)}, \quad 0 \le t < T \end{aligned} \quad (17)$$

Combining the definition of Y_t above with Equations (16) and (17) we get

$$\begin{aligned} \bar{X}_t &= \frac{T-t}{T}\bar{X}_0 + (T-t)\hat{w}_{\varphi(t)}, \\ &= \frac{T-t}{T}\bar{X}_0 + \sqrt{\frac{T-t}{T}t}\,Z \end{aligned} \quad (18)$$

where $\hat{w}_{\varphi(t)} \sim N(0, \varphi(t)) \overset{d}{=} \sqrt{\varphi(t)}\ Z$, with $Z \sim N(0, 1)$, and $\bar{X}_0 = X_T$, Z mutually independent. Finally, since $X_t = X_{T-t}$, $0 \le t < T$, we get

$$X_{T-t} = \frac{T-t}{T}X_T + \sqrt{\frac{T-t}{T}t}\,Z \quad (19)$$

Equation (19) is essentially identical to the backward recursion in Equation (12). Since t is arbitrary except in that $0 \le t < T$, we can make the substitution $t \to T-t$ to rewrite Equation (19) in the more familiar form

$$X_t = \frac{t}{T}X_T + \sqrt{\frac{t}{T}(T-t)}\,Z \quad (20)$$

and use it recursively going backwards.

Remark: A Brownian bridge (Karatzas and Shreve [1996], pp. 358–360) is defined as

$$dX_t = \frac{b - X_t}{T-t}dt + dw_t, \quad 0 \le t < T, \quad X_0 = a$$

and can be thought of as Brownian motion conditioned to start at $X_0 = a$ and end at $X_T = b$. The SDE of a Brownian bridge differs from the time-reversed SDE of a Wiener process in (15) in that a and b represent single points. If we set the final point b to 0 and allow the initial value a to vary over w_T, we arrive at the time-reversed process in Equation (15). In this sense, the time-reversal of a Wiener process is a Brownian bridge.

We take up the non-trivial example of the time-reversed Cox-Ingersoll-Ross process in the next section.

APPLICATIONS

The ability to generate a stochastic process in reverse decouples the number of paths from the number of time steps in Monte Carlo simulations. This proves valuable in many ways. First, it can reduce storage requirements dramatically: memory management is an exponentially more costly affair in units of computational time, so, even when one has the luxury of unlimited memory, a small memory footprint is to be preferred. In addition, the ability to increase the number of paths used in Monte Carlo simulations can be essential for accurate estimation of option values and hedging requirements close to complex barriers and near expiration. Options written on baskets of assets require multi-factor simulations, which can quickly become infeasible if a large number of paths is desired. String models (Santa-Clara and Sornette [2001]) for interest rates, for example, can require as many as 20 independent factors. Or consider a continuously (de)activating FX corridor option, as in Geman [2001]. Unlike an equity corridor option, which is priced using daily settlements, the FX option can activate or deactivate at any time. Geman [2001] provides an analytical technique using a lognormal process for the underlying. This, although useful, ignores the fact that on small time scales like a few minutes, the FX process is not lognormal at all but likely a subordinated process. The just-in-time method is ideally suited to exploring the values of options on very fine time scales. These and other studies will be presented in a subsequent work. In what follows, we consider two applications that illustrate use of the just-in-time method.

Deep-Out-of-the-Money American Put

We use the power of the just-in-time method to highlight a property of the Longstaff-Schwartz LSM pricing technique that is not, perhaps, sufficiently appreciated–that it is possible to *overestimate* an option's value in certain circumstances despite the fact that the LSM method necessarily uses a *sub-optimal* stopping rule. We use the familiar American put on a non-dividend-paying stock to illustrate this point.

To recapitulate, the central difficulty in calculating the value of any option which allows for early exercise is to compute its continuation value at times prior to expiration. Optimal exercise consists of comparing the continuation value with the value of immediate exercise and following the more profitable course. At any time prior to expiration there is, therefore, a value of the underlying below which (for a put) early exercise is optimal, and there is a unique early exercise policy which optimizes the value of the option. Any other policy is sub-optimal (e.g., Duffie [1996] pp. 172–78) *unless* one has inadvertently built some degree of *foresight* into the early exercise policy (stopping rule) in a simulation.

To see how this can happen, consider the LSM method for calculating an option's continuation value. At any time prior to expiration, we regress the value of the underlying for in-the-money paths *alone* against future payoffs on these paths. It can be shown (Carriere [1996], Longstaff and Schwartz [2001]) that in the limit as the number of paths goes to infinity and the time step goes to zero, the regression correctly calculates the continuation value of the option conditional on the value of the underlying and the time to expiration.[12] The key to correct implementation of the LSM method lies in this stepwise regression, which, in turn, depends on 1) the functional relationship posited between the underlying and future payoffs in the regression model and 2) the number of in-the-money paths upon which the continuation value is estimated. It is here that one can go wrong, either by using too many basis functions in an attempt to capture the convexity of the option's continuation value (overfitting) or, more subtly, by failing to realize that when one hits a patch where there are only a few in-the-money paths, the regression can bias an option's value upward.[13]

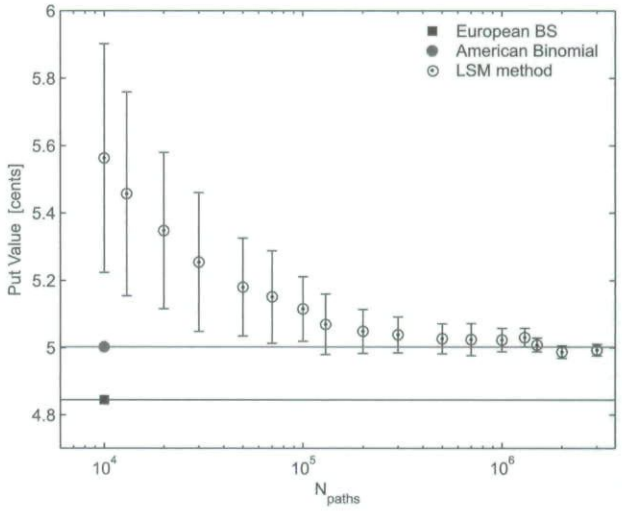
The first peril, that of overfitting, can be avoided by examining the regression at a few, or even every, time step: oscillations in the fitted curve would indicate the need for reducing the number of basis functions or the degree of the polynomial used in the regression. An oscillatory fit would indicate that one has built foresight into the simulation stopping rule—instead of basing early exercise decisions on the average of future payoffs, the simulation stopping rule "knows" that on particular paths the future is better than the present, even when the present is paying off handsomely. The second peril is more difficult to guard against since a quadratic regression is the least one can do in trying to capture an option's convexity, and a quartic is by no means extravagant. Nonetheless, this can lead to systematic overestimation of an option's value.

A deep-out-of-the-money American put illustrates this point. Consider a put on a non-dividend-paying stock with initial price $S_0 = \$100$, annualized volatility of returns $\sigma = 40\%$, and a risk-free rate $r_f = 5\%$. The put's strike and time-to-expiration are $K = \$20$ and $T = 3$ years. With these parameters, only about 1.4% of paths are in the money at expiration. The put's Black-Scholes European value is 4.84 cents, while a binomial tree calculation (10,000 time steps over 3 years) gives an American value of 5.00 cents. We value this put using the Longstaff-Schwartz LSM method with a fourth-order polynomial least squares regression. Paths are generated backwards by the just-in-time method discussed earlier. The time step is one trading day, with 252 trading days in a year. Exhibit 4 displays the value of the put as a function of the number of paths used. Two things stand out: First, a relatively high number of paths, beyond a reasonable memory limit (about 800 MB for Matlab) if all paths are to be stored, is required to obtain a consistent estimate of the value. Second, there is a systematic upward bias to the valuation when the number of paths used is low. At first sight this is puzzling since the correct American value requires use of the unique optimal stopping rule—any departure from the optimal exercise boundary should *underestimate* the put value. What is happening, however, is that for low path counts there are so few in-the-money paths at intermediate times that the basis set is perfectly forecasting future prices. As we increase the number of paths the systematic upward bias deceases significantly, and at high path counts (over a million) the LSM method slightly underestimates the true value, as it should.

For this relatively simple example, generating paths forward and storing them is still feasible up to a point. More complex options with multiple factors, complex barriers, many triggers, underlying values close to barriers, or very long (and very short) times to expiration cannot be handled easily, if at all, by the usual forward propagation and storage route, nor does one usually have the luxury of a lattice-based method against which Monte Carlo results can be examined for any systematic bias. The Monte Carlo method of Longstaff-Schwartz [2001] is simple, intuitive, and flexible. It is the preferred route for practitioners and even academics as lattice models can be very difficult to work out, specially when the underlying process is not lognormal or when two or more underlying processes are involved. The just-in-time method proves invaluable in these cases to bring Monte Carlo studies to a satisfactory conclusion.

EXHIBIT 4

Valuation of a Deep-out-of-the-Money Put as a Function of Path Count Using the Longstaff-Schwartz LSM Method for $S_0 = \$100$, $\sigma = 40\%$, $r_f = 5\%$, $K = \$20$, and $T = 3$ years



Note: The error bars represent one standard error in the mean. The number of runs is decreased in proportion to the number of paths used per run.

The Cox-Ingersoll-Ross Process

The Cox-Ingersoll-Ross process

$$dX_t = -\kappa(X_t - \theta)dt + \sigma\sqrt{|X_t|}\,dW_t, \quad X_0 = r_0 \quad (21)$$

is an equilibrium interest rate model that is fundamental to the pricing of fixed income products. It can be shown (e.g., Ikeda and Wantanabe [1981]) that for $\kappa, \theta \in \mathbb{R}$ and $\sigma > 0$, the CIR process has a unique strong solution on $[0,\infty)$ for every $X_0 = r_0$, and that for $r_0 \geq 0$, $\theta \geq 0$, the solution $X_t \geq 0$. Like the Ornstein-Uhlenbeck process in Equation (6), the CIR process is a mean-reverting process when $\kappa > 0$, with the desirable property of being non-negative when $r_0 \geq 0$, $\theta \geq 0$. We shall assume throughout this work that κ, θ, and $\sigma > 0$. The absolute value in Equation (21) can therefore be dropped. The purpose of this section is to apply the time-reversal theorem to this non-trivial example, to implement the time-reversed CIR process numerically, and to examine the quality of the sample time-reversed CIR process thus generated. Code for the implementation is available on request.

The transition density of the CIR process with $X_0 = r_0$ may be written as (Göing-Jaeschke [1998], Ch. 3)

$$p_t(x) = \frac{2\kappa}{\sigma^2}\frac{e^{\kappa t}}{e^{\kappa t}-1}\left(\frac{x}{r_0}e^{\kappa t}\right)^{\nu/2} \times \exp\left[-\frac{2\kappa}{\sigma^2}\frac{r_0+e^{\kappa t}x}{e^{\kappa t}-1}\right] I_\nu\left[\frac{4\kappa}{\sigma^2}\frac{e^{\kappa t/2}}{e^{\kappa t}-1}\sqrt{r_0 x}\right] \quad (22)$$

where $\nu \equiv 2\kappa\theta/\sigma^2 - 1$, and $I_\nu(\cdot)$ is a modified Bessel function of the first kind. Defining the time-reversed process $\bar{X}_t \equiv X_{T-t}$, $0 \le t < T$, $\bar{X}_0 = X_T$, we apply the time-reversal theorem in Equation (14) to get (after some algebra)

$$d\bar{X}_t = \left[\kappa(\bar{X}_t - \theta) + \sigma^2\left(1+\frac{\nu}{2}\right) - 2\kappa\frac{e^{\kappa(T-t)}}{e^{\kappa(T-t)}-1} \times\left\{\bar{X}_t - e^{-\kappa(T-t)/2}\sqrt{r_0\bar{X}_t}\,\frac{I_{\nu-1}(\psi_t)+I_{\nu+1}(\psi_t)}{2I_\nu(\psi_t)}\right\}\right]dt + \sigma\sqrt{\bar{X}_t}\,d\bar{W}_t \quad (23)$$

where $\bar{W}_t$ is a Wiener process independent of $\bar{X}_0 = X_T$, and

$$\psi_t \equiv \frac{4\kappa}{\sigma^2}\frac{e^{\kappa(T-t)/2}}{e^{\kappa(T-t)}-1}\sqrt{r_0\bar{X}_t} \quad (24)$$

Hint. The calculations required for the time-reversed CIR process are made much simpler by writing the second term on the right of Equation (14b) as

$$\frac{\partial_x\left[\sigma^2(T-t,x)p_{T-t}(x)\right]}{p_{T-t}(x)} = \partial_x\left[\sigma^2(T-t,x)\right] + \sigma^2(T-t,x)\partial_x \log p_{T-t}(x)$$

Remark. The numerical behavior of the drift term in Equation (23) can be awkward when $t \to T^-$, although the limit is perfectly well behaved. Indeed, we note from Equation (24) that $\psi_t \to \infty$ when $t \to T^-$. The asymptotic behavior (Gradshteyn and Ryzhik [1965], p. 962) of a modified Bessel function of the first kind is given by

$$I_\nu(z) \sim \frac{e^z}{\sqrt{2\pi z}}\left[1+o\left(\frac{1}{z}\right)\right]$$

and is seen to be independent of ν to leading order, so

$$\frac{I_{\nu-1}(\psi_t)+I_{\nu+1}(\psi_t)}{2I_\nu(\psi_t)} \to 1$$

as $t \to T^-$, and, therefore (see Equation (23)),

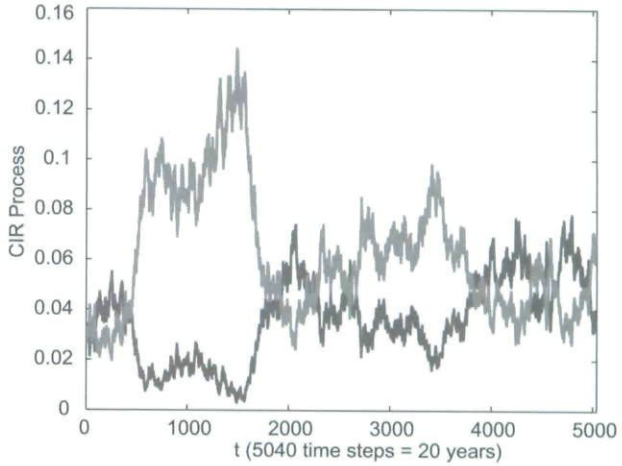
$$\left\{\bar{X}_t - e^{-\kappa(T-t)/2}\sqrt{r_0\bar{X}_t}\,\frac{I_{\nu-1}(\psi_t)+I_{\nu+1}(\psi_t)}{2I_\nu(\psi_t)}\right\} \to \bar{X}_t - \sqrt{r_0\bar{X}_t} \to 0$$

because $\bar{X}_t \to r_0$ when $t \to T^-$. In Matlab simulations with one time step for each trading day (with 252 trading days in a year) over 20 years (5040 time steps), the last three time steps, when $\bar{X}_t \to r_0$, can blow up unless numerical precautions are implemented. The simple expedient of setting the ratio of Bessel functions to 1 when $\psi_t > 700$ rectifies the problem. Note that this correction is necessarily code and platform dependent, so the expedient must be tailored to one's circumstances.[14]

We implement the time-reversed CIR process using the strong order 1.0 Platen scheme (see, e.g., Kloeden [2001]; details of the Platen scheme are provided in the appendix) rather than a simple stochastic Euler scheme. Recall that the Platen scheme is an implicit scheme, and therefore, derivative-free. This is

EXHIBIT 5

Two Paths Generated Backwards Over 20 Years



Notes: Two paths (1+1 antithetic) for the CIR process generated backwards by the time-reversal method over 252 × 20 = 5040 time steps (20 years) using the strong order 1.0 Platen scheme. The annualized parameters correspond to $r_0 = 0.03$, $\theta = 0.06$, $\kappa = 0.25$, and $\sigma = 0.11$.

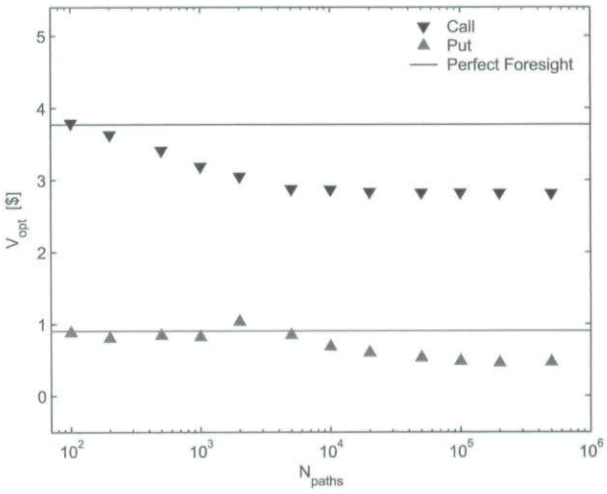
not essential, but is desirable given the complexity of the drift term in Equation (23). Even greater numerical accuracy can be obtained by using the strong order 1.5 Milstein scheme. Exhibit 5 displays two paths generated backwards over 20 years for a representative parameter set.

To illustrate the use of a time-reversed CIR process, we consider the LSM valuation of a bond with embedded options. Specifically, let this bond be a fully and continuously amortizing 30-year fixed-rate, fixed-payment mortgage. The homeowner typically holds two American-style options: 1) the right to prepay the mortgage, i.e., call the mortgage in exchange for the unpaid principal, and 2) the right to default on the mortgage, i.e., put the mortgage to the lender in exchange for the house.[15] The prepayment option increases in value as interest rates drop relative to the original coupon because the borrower can refinance the mortgage at a lower rate. The default option is in-the-money when house values drop to a point where the owner's equity is sufficiently negative.[16] A mortgage valuation clearly has the potential for being memory intensive in a forward implementation of LSM. The natural choice of time-step is monthly, and there are at least two underlying stochastic processes—interest rate and house price. Practitioners generally use multiple-factor term structure models, which exacerbates the problem further.

Here, we choose the aforementioned single-factor CIR process, r_t, parameterized as in Pearson and Sun [1994], and a log-normal house price process, H_t, with a fixed drift of 4% and volatility of 10%.[17] Our other parameter choices are governed by the desire to have the options relatively deep-out-of-the-money (DOTM). This corresponds to the American put considered earlier, where many simulation paths were needed for accurate valuation of the option. A low initial short rate of 2% (implying generally rising rates early on) and a prepayment fee of 5% of the outstanding balance reduces the call option value. While our choice of the initial loan-to-value ratio of 95% is quite high, the 30% default cost chosen ensures that the put option is also relatively DOTM. A positive correlation of 0.7 between the interest rate and house price process also reduces the value of the default put. The mortgage coupon rate is chosen by iteration as the one that sets the mortgage market value at $t = 0$ equal to the initial principal amount.[18] Exhibit 6 shows the value of the embedded options (per $1,000 of initial loan amount) as a function of N_{paths}, the number of paths used in the simulation. The processes H_t and r_t are generated in reverse, using

EXHIBIT 6

The Value of Embedded Options as Functions of the Number of Paths



Note: The $t = 0$ value of the prepayment call (down triangle) and default put (up triangle) per $1,000 of initial loan for the mortgage described in the text, as a function of the number of paths used in the simulation. The horizontal lines show the respective option values if perfect foresight is possible along each path.

Equations (3) and (23) respectively.[19] The basis functions used to estimate the expected continuation values is the monomial set with a highest combined power of 3, i.e., $\{(r/r_0)^i(H/H_0)^j\}$, $0 \le i + j \le N_{basis} = 3$. Downward-pointing triangles correspond to the prepayment call value, while upward-pointing triangles refer to the default put value. As in our DOTM American put example, we see that for low path multiplicity the least squares method overestimates the option value, as inferred from the value at large N_{paths}. In fact, the horizontal lines in Exhibit 6 are the expected present value of the embedded options if the borrower were somehow able to exercise at the maximum present value payoff along *each path*, i.e., with "perfect foresight." Using a low number of paths in the Monte Carlo valuation reproduces this "perfect foresight" option value because with ten basis functions and very few in-the-money (ITM) paths, one is no longer computing the *expected* continuation value in a least squares sense. The "true" option value is only correctly estimated once the ITM path multiplicity is significantly larger than the number of degrees of freedom of the basis set. In our simple example, this happens once N_{paths} exceeds 10,000 for the call, and about 100,000 for the put. For more general multi-factor interest rate models,

a forward least squares implementation with complete storage of all paths would be constrained by memory management issues, and would likely be hard pressed to produce reliable estimates.

CONCLUSION

American options valuation by the least squares Monte Carlo method of Longstaff and Schwartz [2001] is a valuable tool for practitioners and academics alike. Its utility lies in the ease with which the LSM method can accommodate general stochastic processes and complex exercise policies. Its limitation arises from the need for storing the underlying process over the lifetime of the option being evaluated, especially when a very long life, a very fine time step, a complex exercise boundary close to expiration, or any combination of these properties comes into play. Accurate simulation can require a high density of paths and a very fine time step, leading to storage and memory management requirements that can rapidly become infeasible. In this article, we demonstrate that it is possible to start with the final distribution of the underlying process(es) and to generate prior realizations in reverse such that all serial and cross-sectional requirements are satisfied. In the "just-in-time" least squares options valuation method, the state-of-the-world realizations are generated by sampling the final distribution at expiration and proceeding in reverse to the initial condition, discarding future realizations as one proceeds backward when they are no longer needed for the pricing algorithm.

There is a simple intuition for why stochastic processes can be generated in reverse. Given a price point of some price process, it is, naturally, impossible to reconstruct the path (history) of the asset—an infinite number of paths can reach that price point from the past. However, the entire *process*, *which includes every possible path*, can always be reconstructed given sufficient information about the underlying process and the distribution of prices at some future time, provided that the associated stochastic differential equation generates a unique solution going forward.

As pointed out earlier, the just-in-time method does not offer an advantage if the entire history of price paths is required. This is the case for some variants of Asian options, where the strike can be the average over the entire history of the path or the strike may be fixed but the payoff is the difference between the fixed strike and the average stock price, the average being calculated over the entire stock price history. In these cases, the storage requirement is the same for both the usual Monte Carlo method and the just-in-time method. This is one limitation on the use of the reverse Monte Carlo method. A second limitation arises when one uses the numerical trick employed in the subordinated processes section to reverse general processes. The seed storage method cannot be used to reverse the underlying stochastic process from an arbitrary terminal distribution. Apart from these limitations, the just-in-time method is very general and flexible and can be used to extend the reach and accuracy of many Monte Carlo simulations.

In conclusion, we have successfully married the backward valuation technique required for option pricing with a backward generation technique for many price processes of interest to financial economists and practitioners. The just-in-time method, which can be thought of as stochastic *involution*, extends the reach and accuracy of Monte Carlo techniques beyond what has hitherto been possible and should prove useful in investigating the properties of options on very fine time scales, options written on single or multiple assets with complex American triggers, long-lived options, or any combination of these properties.

APPENDIX A

PROOFS

The following stipulation holds throughout this work: $\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_n, \tilde{W}_n$ are mutually independent Gaussian random variables such that $\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_n \sim N(0,1)$, and $\tilde{W}_n \sim N(0, n)$. Also, we make frequent use of an elementary equivalence: two Gaussian random variables are independent if and only if they are uncorrelated. This implies that three or more Gaussian random variables are mutually independent if and only if they are pairwise uncorrelated. Proofs of the independence of Gaussian random variables will, accordingly, reduce to establishing that their covariance goes to zero.

Proof of Proposition 1. We proceed by backward induction. Assume that for some $i \leq n$, $\tilde{W}_i \sim N(0, i)$. From the recursion defined in Equation (3) we infer that $\tilde{W}_i$ is a linear polynomial in $\tilde{z}_{i+1}, \tilde{z}_{i+2}, \ldots, \tilde{z}_n, \tilde{W}_n$ alone, that is, $\tilde{W}_i$ has no dependence on $\tilde{z}_i$. Therefore, $\tilde{W}_i$ and $\tilde{z}_i$ are independent. Now, since

$$\tilde{W}_{i-1} \equiv \left(1 - \frac{1}{i}\right)\tilde{W}_i + \sqrt{1 - \frac{1}{i}}\,\tilde{z}_i$$

is the sum of two mean zero Gaussian random variables, it is itself a mean zero Gaussian random variable. Furthermore, since $\tilde{W}_i$ and $\tilde{z}_i$ are independent their covariance is zero, so

$$\mathrm{var}(\tilde{W}_{i-1}) \equiv \left(1-\frac{1}{i}\right)^2 \times i + \left(1-\frac{1}{i}\right) \times 1 = i-1$$

Thus, $\tilde{W}_{i-1} \sim N(0, i-1)$. Since $\tilde{W}_i \sim N(0, i-1) \Rightarrow \tilde{W}_{i-1} \sim N(0, i-1)$ and since the proposition clearly holds for the case $i = n$ because $\tilde{W}_n \sim N(0, n)$ is true by stipulation, we have

$$\tilde{W}_i \sim N(0, i),\ i = 0, 1, \ldots, n \qquad \text{(A1)}$$

Before proceeding to Proposition 2 we state and prove

Lemma 1. Let $0 \le i \le j \le n$, and let $\tilde{W}_i$, $\tilde{W}_j$ be constructed from $\tilde{W}_n$ by the recursion defined in Equation (3). Then,

$$\mathrm{cov}(\tilde{W}_i, \tilde{W}_j) = \min(i, j) = i \qquad \text{(A2)}$$

Proof of Lemma 1. Let $0 \le i \le j \le n$. When $i = j$, Proposition 1 tells us that $\mathrm{var}(\tilde{W}_i) = i$, $i = 0, 1, \ldots, n$. Now consider the case $i < j$. From the recursion relationship in Equation (3) we write

$$\begin{aligned}
\tilde{W}_i &= \frac{i}{i+1}\tilde{W}_{i+1} + \sqrt{\frac{i}{i+1}}\tilde{z}_{i+1} \\
&= \frac{i}{i+1} \times \frac{i+1}{i+2}\tilde{W}_{i+2} + \frac{i}{i+1} \times \sqrt{\frac{i+1}{i+2}}\tilde{z}_{i+2} + \sqrt{\frac{i}{i+1}}\tilde{z}_{i+1} \\
&= \frac{i}{i+1} \times \frac{i+1}{i+2} \times \cdots \times \frac{j-1}{j}\tilde{W}_j \\
&\quad + \text{linear polynomial in } \{\tilde{z}_{i+1}, \tilde{z}_{i+2}, \ldots, \tilde{z}_j\} \\
&= \frac{i}{j}\tilde{W}_j + \text{linear polynomial in } \{\tilde{z}_{i+1}, \tilde{z}_{i+2}, \ldots, \tilde{z}_j\} \qquad \text{(A3)}
\end{aligned}$$

while, by the same recursive expansion

$$\tilde{W}_j = \text{linear polynominal in } \{\tilde{z}_{j+1}, \tilde{z}_{j+2}, \ldots \tilde{z}_n, \tilde{W}_n\} \qquad \text{(A4)}$$

From the mutual independence of $\tilde{z}_1, \tilde{z}_2, \ldots \tilde{z}_n, \tilde{W}_n$ we infer that the covariance of $\tilde{W}_j$ with the linear polynomial term in $\{\tilde{z}_{i+1}, \tilde{z}_{i+2}, \ldots, \tilde{z}_j\}$ on the RHS of (A3) is zero. Combining Equations (A3) and (A4) leads to

$$\mathrm{cov}(\tilde{W}_i, \tilde{W}_j) = \mathrm{cov}\left(\left[\frac{i}{j}\tilde{W}_j + \text{linear polynomial in } \{\tilde{z}_{i+1}, \tilde{z}_{i+2}, \ldots, \tilde{z}_j\}\right], \tilde{W}_j\right) = i$$

since, from Equation (A1), $\mathrm{var}(\tilde{W}_j) = j$.

Proof of Proposition 2.

1. Setting $i = 1$ in the recursive definition of $\tilde{W}_{i-1}$ in (3) leads to $\tilde{W}_0 = 0$.
2. Let $0 \le i \le j \le n$. Since $\tilde{W}_i \sim N(0, i)$, $\tilde{W}_j \sim N(0, j)$ are mean zero Gaussian random variables (see Proposition 1), so is $\tilde{W}_j - \tilde{W}_i$ a mean zero Gaussian random variable. Also

$$\begin{aligned}
\mathrm{var}(\tilde{W}_j - \tilde{W}_i) &= \mathrm{var}(\tilde{W}_j) + \mathrm{var}(\tilde{W}_i) - 2\,\mathrm{cov}(\tilde{W}_j, \tilde{W}_i) \\
&= j + i - 2\min(j,i) = j - i
\end{aligned}$$

   where we have used Equation (A2).
3. Let $0 \le i \le j \le k \le n$. Independence of the non-overlapping increments $\tilde{W}_k - \tilde{W}_j$ and $\tilde{W}_j - \tilde{W}_i$ is most easily established by calculating their covariance, which must be zero. Thus

$$\begin{aligned}
\mathrm{cov}(\tilde{W}_k - \tilde{W}_j, \tilde{W}_j - \tilde{W}_i) &= \mathrm{cov}(\tilde{W}_k, \tilde{W}_j) - \mathrm{cov}(\tilde{W}_k, \tilde{W}_i) \\
&\quad - \mathrm{cov}(\tilde{W}_j, \tilde{W}_j) + \mathrm{cov}(\tilde{W}_j, \tilde{W}_i) \\
&= \min(k, j) - \min(k, i) \\
&\quad - \min(j, j) + \min(j, i) \\
&= 0
\end{aligned}$$

where we have repeatedly used Equation (A2).

Proof of Proposition 3. This is a standard exercise and is worked out in many places. We sketch out a proof here. Let $f(t)$ be such that $f(t)$ is continuous on $[0,\infty)$ and $f(t) \neq 0$ almost everywhere. Then $\int_0^t f^2(s)ds$ is a monotonically increasing function of t, so the map $\varphi : t \to \tau$ defined below is invertible

$$\tau = \varphi(t) \equiv \int_0^t f^2(s)ds \qquad \text{(A5)}$$

Now define

$$\tilde{W}_\tau \equiv \int_0^{\varphi^{-1}(\tau)} f(s)d\tilde{w}_s = \int_0^t f(s)d\tilde{w}_s \qquad \text{(A6)}$$

Since $\tau(0) = 0$ from Equation (A5), we have from Equation (A6)

$$\tilde{W}_0 = 0 \qquad \text{(A7)}$$

Futher, assuming $\tau_i < \tau_j$, and, thus, $t_i < t_j$, along with use of Equation (A6),

$$\tilde{W}_{\tau_j} - \tilde{W}_{\tau_i} = \int_{t_i}^{t_j} f(s)d\tilde{w}_s \tag{A8}$$

so $\tilde{W}_{\tau_j} - \tilde{W}_{\tau_i}$ is a mean zero Gaussian random variable (it is the sum of infinitesimal mean zero Gaussian random variables). The variance of $\tilde{W}_{\tau_j} - \tilde{W}_{\tau_i}$ follows from Equation (A8) and the identity in Equation (8)

$$E\left[\tilde{W}_{\tau_j} - \tilde{W}_{\tau_i}\right]^2 = \int_{t_i}^{t_j} f^2(s)ds$$
$$= \int_0^{t_j} f^2(s)ds - \int_0^{t_i} f^2(s)ds = \tau_j - \tau_i \tag{A9}$$

Next, we can write

$$\tilde{W}_{\tau_j} = \int_0^{t_i} f(s)d\tilde{w}_s + \int_{t_i}^{t_j} f(s)d\tilde{w}_s \tag{A10}$$

so that

$$\text{cov}(\tilde{W}_{\tau_i}, \tilde{W}_{\tau_j}) = \int_0^{t_i} f^2(s)ds = \tau_i = \min(\tau_i, \tau_j) \tag{A11}$$

The second term on the RHS of Equation (A10) does not contribute to the covariance since it is independent of the first term. Note that we have again used the identity Equation (8). Equation (A11) suffices to prove that for all $0 < \tau_i < \tau_j < \tau_k$

$$\text{cov}(\tilde{W}_{\tau_k} - \tilde{W}_{\tau_j}, \tilde{W}_{\tau_j} - \tilde{W}_{\tau_i}) = 0 \tag{A12}$$

Thus we have established that the process $\tilde{W}_{\tau(t)}$ is 1) 0 at $\tau = 0$, 2) has mean zero marginals with variance equal to the difference $\Delta\tau$ between the relevant epochs, and 3) has independent non-overlapping increments. This establishes that $\tilde{W}_{\tau(t)}$ is a Wiener process in the transformed time τ. Finally, $f(t) = e^{\kappa t}$ obviously satisfies the technical requirements mentioned earlier, and

$$\tau(t) = \int_0^t e^{\kappa s}ds = \frac{e^{2\kappa t} - 1}{2\kappa}$$

so the desired result obtains.

Proof of Proposition 4. The proof follows exactly along the lines of Proposition 2.

Strong Order 1.0 Platen Scheme. Let $0 = t_0 < t_1 \cdots < t_n = T$ be the times at which we wish to compute X_t, given that

$$dX_t = \mu(t, X_t)dt + \sigma(t, X_t)dW_t, \quad X_0 = x_0$$

and let $\Delta_i \equiv t_{i+1} - t_i$. The Platen scheme, which is an implicit scheme consistent with Itô calculus, provides that

$$X_{i+1} = X_i + \mu(t_i, X_i)\Delta_i + \sigma(t_i, X_i)\Delta W_i$$
$$+ \frac{1}{\sqrt{\Delta_i}}[\sigma(t_i, \Upsilon_i) - \sigma(t_i, X_i)]J_i$$

where

$$\Upsilon_i = X_i + \mu(t_i, X_i)\Delta_i + \sigma(t_i, X_i)\sqrt{\Delta_i}, \quad J_i = \frac{1}{2}[(\Delta W_i)^2 - \Delta_i]$$

Details of the Platen scheme and other numerical integration schemes consistent with Itô calculus may be found in Kloeden [2001].

Matlab/Pseudo-code for Generating a Wiener Process in Reverse: Given below is Matlab code for generating a Wiener process in reverse. The accompanying comments serve as pseudo-code.

```
% Generate M Brownian motion
% (Wiener Process) paths
% in reverse over n steps.

M =     200;    % number of paths
n =     100;    % number of steps

% Generate terminal distribution:
% the volatility of a standard
% normal distribution, N(0, 1), is
% one, and scales with the square root
% of the number of steps; we make M
% independent draws form N(0, 1), and
% multiply by sqrt(n) to get the
% correct terminal volatility.
%
W(:,n+1)     = sqrt(n)*randn(M, 1);

% iterate backwards
%
for i=n: -1:1
    W(:,i) = (1-1/i)*W(:,i+1)+...
    sqrt(1-1/i)*randn(M,1);
end

% Matlab numbers array elements form
% 1, rather than 0 as in C/C++,
% so the first colum or W corresponds
```

```
% to t = 0.
%
plot(W' )    % display paths
```

Matlab code for an American put using reverse Monte Carlo in the LSM Algoritm:

```
% clear the stack and screen

clear;clc

% initialize

% we assume 252 tradding days/year

% … is Matlab syntax for

% "continues on next line"

%
S0   = 1;        % initial stock price
K    = 1;        % strike price
rf   = 0.05;     % annualized riskfree rate
sig  = 0.40;     % annualized volatility
T    = 1;        % option life
n    = …
floor(252*T);    % number of steps in T yrs;
dT   = T/n;      % step size
M    = 1000000;  % number of paths

% -> generate terminal distribution
% half the final distribution
Zl      = sqrt(n)*randn(M/2,1);
% antithetize
Send    = exp(sig*sqrt(dT)*[Zl;-Zl]);
% martingalize and boost
Send    = Send/mean(Send)*S0*exp(rf*T);

% -> European value
EVend  = max(0,K-Send);
EValue = mean(EVend)*exp(-rf*T);

% -> American value: LSM method begins
tEx     = T;
nl      = n;
for j=n-1:-1:1
    %
    tEx       = tEx-dT;
    % generate prior step distribution
    Zb        = (1-1/nl)*…
    Zl+sqrt(1-1/nl)*randn(M/2,1);
    Sj        = exp(sig*sqrt(dT)*[Zb;-Zb]);
    Sj        = Sj/mean(Sj)*S0*exp(rf*tEx);
    %
    EVend     = EVend*exp(-rf*dT);
    % early exercise values
    EVj       = max(0,K-Sj);
    % in-the-money paths
    itm       = EVj > 0;
    x         = Sj(itm);
    X         = [ones(length(x),1)…
     x x.^2 x.^3 x.^4];
    y         = EVend(itm);
    b         = regress(y, X);
    CVj       = zeros(M,1);
    % continuation values
    CVj(itm) = X*b;
    % early exercise
    ee        = EVj > CVj;
    EVj(~ee) = 0;
    EVend(ee)= 0;
    EVend     = EVend+EVj;
    nl        = nl-1;
    Zl        = Zb;
end


% American put value

AValue        = mean(EVend)*exp(-rf*dT)
```

ENDNOTES

We thank Sanjiv Das, Robert Edelstein, Dwight Grant, Dwight Jaffee, Yuan Ma, Terry Marsh, Paul Pfleiderer, and Jacob Sagi for discussions that have contributed materially to this work.

[1]In Longstaff and Schwartz [2001], the authors illustrate the flexibility and power of their least squares Monte Carlo method by using it to value a sequence of increasingly more complex contingent claims, starting with a vanilla American put (which is, of course, more easily handled using a binomial tree) and ending with a deferred American swaption in a 20 factor interest rate string model. Finite difference techniques and tree (lattice) methods can quickly become formidably difficult to implement as the number of stochastic factors increases, if this can be done at all. The use of non-lognormal processes adds to the already considerable difficulty.

[2]While 64-bit operating systems and computers with many gigabytes of random access memory are becoming increasingly more common, it should be borne in mind that managing large amounts of memory is very costly in units of computational time. Code written to use the smallest memory footprint possible executes exponentially faster.

[3]The usefulness of "just-in-time" Monte Carlo can be brought out by an example. A popular style of investment in private equity markets is structured thus: An investment house provides a sum of capital to a publicly traded company. In return, it is allowed to purchase, at a time of its choosing, a variable number of shares at an average price. The average price is computed over, say, a 30-day moving window. Additional triggers

might apply. For example, the average price can be the smaller of the moving window average and the smallest 10-day average within the moving window, or the smallest of the moving window average and the average of the first 3 trading days and the last 3 trading days of the window. In principle, the LSM algorithm easily deals with the complex American triggers built into this option. In practice, the algorithm needs a large number of paths in order to compute conditional expectations correctly from crosssectional data. In addition, these private equity options can be very long-lived, exceeding 10 years. Some can even be perpetuities. Accurate simulation of the option value again requires a large number of paths. Storing a million paths with a daily time step over, say, 14 years is not practicable (storage = 14 years × 252 trading days/year × 1 million paths × 8 bytes/number ≥ 28 GB). However, since at any point in the backward pricing algorithm all one needs is the price history for the last 30 days, the just-in-time method makes it possible to simulate the American option value accurately by allowing for the use of a very large number of paths, even a million (storage ≥ 240 MB). It should be noted that the just-in-time method provides no particular advantage in those cases where the entire history of the price path is required.

[4]Needless to say, this cannot be done by the naive expedient of sampling a return distribution of volatility $\sqrt{n}\sigma$ at time step n, and so on. The resulting process would not be Brownian.

[5]Continuity can be inferred in the appropriate limit from the recursive construction of $\tilde{W}_i$ from $\tilde{W}_{i-1}$. The limit is effected with use of the self-similar scaling property $\sqrt{\Delta t}\tilde{W}_i \stackrel{d}{=} \tilde{W}_{\Delta t \times i}$ of a Wiener process.

[6]As we move backwards, n reduces by 1 each step until we reach $n = 2$. It is easily seen from Equation (5) that for $n = 2$, the LHS is $\tilde{W}_1$, while the RHS is zero. Thus, all paths terminate at zero, as they should.

[7]If a process can be written as $Y_t = X_{\tau(t)}$, where X_u, $u \geq 0$ is any stochastic process and $\tau(t)$ is an increasing process with non-negative, homogeneous, and independent increments, Y_t is said to be subordinated to the process X_u, and $\tau(t)$ is called the directing process (Feller [1984]), or the subordinator. X_u and $\tau(t)$ are taken to be independent processes. $\tau(t)$ is variously called the business time, economic time, or transaction time in the finance literature.

[8]See the seminal papers by 1) Mandelbrot [1963], who models variations in cotton futures prices as an α-stable distribution with infinite variance, and 2) Clark [1973], who examines the same variation in cotton futures prices as a finite-variance subordinated process. Mandelbrot's use of an α-stable distribution to model price increments is equivalent to the use of an $\frac{\alpha}{2}$-stable subordinator, although he does not take this point of view.

[9]The related and important question of the correct proxy for the economic clock is examined in a number of works. Karpoff [1987] is an early survey of the relationship between trading volume and price changes. Jones, Kaul, and Lipson [1994] argue that price changes are driven by the number of trades and "not their size." Ané and Geman [2000] concur that the cumulated number of trades is the better proxy, volume having negligible explanatory power when conditioned on the number of trades. In this interesting paper, Ané and Geman [2000] study high frequency returns of Cisco and Intel stock. The subordinator is proxied by the cumulated number of trades and modeled by its first four moments. Given the remarkable fit to normality of Cisco and Intel returns consequent upon subordination, the work of Ané and Geman [2000] suggests that high frequency asset price dynamics may credibly be viewed as a subordinated Wiener process with cumulated number of trades as the subordinator.

[10]This is a fast operation in Matlab, taking only about 0.313 seconds for a million draws from a lognormal distribution on a 2.8 GHz Xeon running Matlab in interpreted mode. Compiled code would be faster.

[11]See Haussmann and Pardoux [1986] for the d-dimensional version of this result, and for technical restrictions on the coefficients μ: $[0,T] \times \mathbb{R}^d \to \mathbb{R}^d$, and σ: $[0,T] \times \mathbb{R}^d \to \mathbb{R}^d \otimes \mathbb{R}^l$ of the process X_t, and its law $p_t(x)$, when w_t is an l-dimensional Wiener process in $\mathbb{R}^l$.

[12]Note that the regression provides the option's *continuation* value at each time step conditional on the value of the underlying and the time-to-expiration, not the option value itself. The option value is computed from the early exercise boundary value implied at each time step.

[13]It is sometimes observed, by way of objection to the just-in-time method, that the least squares Monte Carlo technique requires, at any particular time step, use of all future cash flows relative to that time step. If true, this would imply that as one moves backwards in the pricing algorithm, an increasingly larger fraction of the complete cash flow matrix would need to be stored, thus nullifying the usefulness of the just-in-time method. This observation is not correct. The LSM method generates a stopping rule as one proceeds backwards. In the case of a vanilla put, for example, it generates, at each time step t, the optimal exercise boundary value at t. Computation of the stopping rule at time $t - 1$ requires knowledge of the stopping rule only at times t, $t + 1$,..., T, and of all cash flows at $t - 1$, but *not* details of cash flows along every path at each future time step. This is nicely illustrated in the example provided in Longstaff and Schwartz [2001], Section 1, pp. 115–120.

[14]Using the inbuilt Bessel functions in Matlab R13, the ratio of Bessel functions $[I_{\nu-1}(\psi_t) + I_{\nu+1}(\psi_t)] / 2I_\nu(\psi_t)$ equals 0.999286 (six significant places) when $\psi_t = 700$, but blows up for larger values of ψ_t. The value of ν corresponds to the parameter set used in generating Exhibit 5 ($\nu \approx 1.47934$). Comparison with output from Mathematica 5.1, *which can compute to arbitrary precision*, yields 0.999288 (6 significant places) for the same values of ν and ψ_t, indicating that the Matlab R13 result is accurate to 5 significant places. Mathematica returns 0.999950

(6 significant places) for ν as before, and $\psi_t = 10^4$, while Matlab returns NaN. The error introduced by approximating the ratio of Bessel functions by 1 when $\psi_t > 700$ is, therefore, very small, and, in any case, swamped by sample size effects. This issue is of no particular concern if the code is implemented in Mathematica, or if special purpose code is written for modified Bessel functions of the first kind for use with Matlab. Users of C or C^{++} may wish to keep these considerations in mind.

[15]See Hendershott and van Order [1987] for a review of early work pricing mortgages as risk-free debt with embedded contingent claims. As is customary, we assume that the mortgage is non-recourse, so that the lender is not entitled to seize other assets of the borrower.

[16]In principle, the borrower minimizes the mortgage debt by optimally exercising the call or put ("rational" exercise), subject to the payment of a fee. These fees involve not only financial payments (refinancing or lower credit scores), but also non-financial costs (effort) and are generally found to be quite large. For an illustration of empirical work on mortgage pricing see, for example, Stanton [1995]. In practice, mortgage values are also affected by "exogenous" termination, such as relocation, unemployment, etc. Our expositional calculation here ignores such events, though they may be incorporated using an empirical external hazard.

[17]In principle, the drift should be given by the risk-free rate, less a maintenance adjusted dividend flow.

[18]With our parameters, we find a spread of 4.4% over the initial CIR rate, the bulk of which arises from the rising term structure. Longstaff [2002] has pointed out that the correct exercise decision takes into account the lifetime cost of refinancing the loan. This requires recomputing the spread on a grid of interest rates and house prices. Here, we have ignored this issue and assumed that the spread is embedded in the exercise fees.

[19]Note that the processes r_t and H_t are correlated, so that the final distributions at $t = T$ are not independent. In practice, we sample $P_r(r_T, H_T)$ by generating correlated paths forward without storing intermediate values.

REFERENCES

Anderson, Brian D.O. "Reverse-Time Diffusion Equation Models." *Stochastic Processes and Their Applications*, 12 (1982), pp. 313–326.

Ané, Thierry, and Hélyette Geman. "Order Flow, Transaction Clock, and Normality of Asset Returns." *The Journal of Finance*, 55 (2000), pp. 2259–2283.

Carr, Peter, Hélyette Geman, Dilip B. Madan, and Marc Yor. "Stochastic Volatility for Lévy Processes." *Mathematical Finance*, 13 (2003), pp. 345–382.

Carriere, Jacques F. "Valuation of the Early-Exercise Price for Options Using Simulations and Non parametric Regression." *Insurance: Mathematics and Economics*, 19 (1996), pp. 19–30.

Clark, Peter K. "A Subordinated Stochastic Process Model with Finite Variance for Speculative Prices." *Econometrica*, 41 (1973), pp. 135–155.

Cox, John C., Stephen A. Ross, and Mark E. Rubinstein. "Option Pricing: A Simplified Approach." *Journal of Financial Economics*, 7 (1997), pp. 229–263.

Delbaen, F., and W. Schachermayer. "A General Version of the Fundamental Theorem of Asset Pricing." *Mathematische Annalen*, 300 (1994), 463–520.

Dixit, A.K., and R.S. Pindyck. *Investment Under Uncertainty*. Princeton, New Jersey: Princeton University Press, 1994.

Duffie, Darrell. *Dynamic Asset Pricing Theory*. Princeton, New Jersey: Princeton University Press, 1996.

Feller, William. *An Introduction to Probability Theory and Its Applications:* Vol. II. New Delhi: Wiley Eastern, 1984.

Geman, Hélyette. "Time Changes, Laplace Transforms and Path-Dependent Options." *Computational Economics*, 17 (2001), pp. 81–92.

——. "Pure Jump Lévy Processes for Asset Price Modelling." *Journal of Banking and Finance*, 26 (2002), pp. 1297–1316.

Geman, Hélyette, Dilip B. Madan, and Marc Yor. "Time Changes for Lévy Processes." *Mathematical Finance*, 11 (2001), pp. 79–96.

Göing-Jaeschke, Anja. "Parameter Estimation and Bessel Processes in Financial Models and Numerical Analysis in Hamiltonian Dynamics." Unpublished dissertation, Diss. ETH No. 12566, Swiss Federal Institute of Technology, Zurich (1998).

Gradshteyn, I.S., and I.M. Ryzhik. *Table of Integrals, Series, and Products*. New York: Academic Press, 1965.

Haussmann, U.G., and E. Pardoux. "Time Reversal of Diffusions." *The Annals of Probability,* 14 (1986), pp. 1188–1205.

Hendershott, Patric H., and Robert van Order. "Pricing Mortgages: An Interpretation of the Models and Results." *Journal of Financial Services Research*, I (1987), pp. 77–111.

Ikeda, N., and S. Watanabe. *Stochastic Differential Equations and Diffusion Processes*. Amsterdam: North-Holland, 1981.

Jones, Charles M., Gautam Kaul, and Marc L. Lipson. "Transactions, Volume, and Volatility." *The Review of Financial Studies*, 7 (1994), pp. 631–651.

Karatzas, Ioannis, and Steven E. Shreve. *Brownian Motion and Stochastic Calculus*. New York: Springer-Verlag, 1996.

Karpoff, Jonathan M. "The Relation Between Price Changes and Trading Volume: A Survey." *The Journal of Financial and Quantitative Analysis*, 22 (1987), pp. 109–126.

Kloeden, P.E. "A Brief Overview of Numerical Methods for Stochastic Differential Equations." (2001), http://www.maths.uq.edu.au/~pmd/milano.ps.

Ljung, L., and T. Kailath. "Backwards Markovian Models for Second-Order Stochastic Processes." *IEEE Transactions on Information Theory*, 22 (1976), pp. 488–491.

Longstaff, Francis A. "Optimal Recursive Refinancing and the Valuation of Mortgage-Backed Securities." Anderson School of Business, UCLA, and NBER (2002).

Longstaff, Francis A., and Eduardo S. Schwartz. "Valuing American Options by Simulation: A Simple Least-Squares Approach." *The Review of Financial Studies*, 14 (2001), pp. 113–147.

Mandelbrot, Benoit B. "The Variation of Certain Speculative Prices." *The Journal of Business*, 36 (1963), pp. 394–419.

Monroe, Itrel. "Processes that Can Be Embedded in Brownian Motion." *Annals of Probability*, 6 (1978), pp. 42–56.

Nagasawa, M. "Segregation of a Population in an Environment." *Journal of Mathematical Biology*, 9 (1980), pp. 213–235.

Pearson, N.D., and T.-S. Sun. "Exploiting the Conditional Density in Estimating the Term Structure: An Application to the Cox, Ingersoll, and Ross Model." *Journal of Finance*, 49 (1994), pp. 1279–1304.

Santa-Clara, Pedro, and Didier Sornette. "The Dynamics of the Forward Interest Curve with Stochastic String Shocks." *The Review of Financial Studies*, 14 (2001), pp. 149–185.

Schrödinger, Erwin. "Über die Umkehrung der Naturgesetze." *Sitzungsberichte der Preussischen Akademie der Wissenschaften Berlin*, Phys.-Math. Kl. 8–9 (1931), pp. 144–153.

Stanton, Richard. "Rational Prepayment and the Valuation of Mortgage-Backed Securities." *Review of Financial Studies*, 8 (1995), pp. 677–708.

Vasicek, Oldrich. "An Equilibrium Characterization of The Term Structure." *Journal of Financial Economics*, 5 (1977), pp. 177–188.