RC 20570 (10/4/96)
Computer Science / Mathematics

IBM Research Report

Comparison of Path Generation Methods for Monte Carlo Valuation of Single Underlying Derivative Securities: Working Paper

Leonard Berman

IBM Research Division
T.J. Watson Research Center
Yorktown Heights, New York

IBM Research Division
Almaden · T.J. Watson · Tokyo · Zurich

Comparison of Path Generation Methods for Monte Carlo Valuation of Single Underlying Derivative Securities
Working Paper

Leonard Berman*

$Revision: 1.7 Date: 1996/09/09 19:18:35 GMT$
LaTeX'ed: October 4, 1996
IBM Research Report RC 20570

Abstract

Monte Carlo simulations based on quasi-random Sobol sequences and pseudo-random numbers are compared with and without a stratified sampling variance reduction procedure (VR). We find that Sobol sequencs are markedly superior to pseudo-random without VR and comparable with VR.

1 Introduction

In a risk neutral world, the fair value of a financial contract is the present value of the expected future cash flows. These cash flows occur as the world evolves into one of its possible future states. The value of the contract can be expressed as:

$$\int_{\mathcal{P}} v(p)\, d\mathcal{P}(p) \tag{1}$$

where $\mathcal{P}$ is a probability measure on the space of paths describing the possible evolution of the world, and v is the *payoff function* describing the value of the contract if the world follows path p. The probability measure on paths is determined by a set of *stochastic differential equations* (SDEs) chosen to model the evolution of the factors on which the value of the instrument depends.

Boyle [Boy77] first observed that, to use the *Monte Carlo simulation* (MCS) method to estimate this value, the continuous evolution can be discretized in time, the set of paths approximated by piece-wise linear d-segment paths[1], and

*IBM MACC Center and Research Division. *e-mail address:*berman@watson.ibm.com *phone:*212-493-2142

[1]We use d to indicate the number of steps in a path and n to indicate the number of paths in a simulation. We also use the phrases *(d-segment) paths* and *(d-dimensional) points* interchangeably.

the integral (1) approximated by the d-fold integral[2]:

$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} v(\mathbf{x})\, d\mathcal{P}(\mathbf{x}) \tag{2}$$

where $\mathbf{x} \in \Re^d$ is an d-step path, $v(\mathbf{x})$ is the payoff function restricted to d-step paths, and $\mathcal{P}(\mathbf{x})$ is the measure induced on these paths by the system of SDEs. The law of large numbers then permits the integral to be estimated as

$$\frac{\sum_{i=1}^{n} v(\mathbf{x}_i)}{n}$$

where the $\mathbf{x}_i$ are chosen randomly in accordance with the measure $\mathcal{P}$. Furthermore, Chebychev's inequality [KW86] states that for these x_i:

$$\frac{\sum_{i=1}^{n} f(x_i)}{n} - \int f d\mathcal{P}(\mathbf{x}) = \mathcal{O}(\frac{1}{\sqrt{n}}) \tag{3}$$

if the variance of the integral exists.

Monte Carlo methods have advantages and disadvantages. The primary advantages are ease of use and breadth of application, i.e. MCS valuation procedures can easily be constructed for complicated pay-off functions involving multiple underlying factors. Furthermore, for a fixed accuracy, the run time of an MCS procedure is independent of the number of underlyings. The primary disadvantage is that they are expensive to apply and therefore, when alternative valuation methods exist, the Monte Carlo method is almost never the method of choice; however, for many types of contract, Monte Carlo is the only valuation method available.

Recently, there has been a good deal of attention [TW94, PT95] directed towards accelerating "Monte Carlo" type calculations, by using quasi-random sequences of points rather than pseudo-random sequences of points[3] to approximate (2). The use of quasi-random sequences is motivated by the desire to avoid the $1/\sqrt{n}$ complexity inherent in the law of large numbers. It can be shown [Nie92] that for continuous f and carefully chosen quasi random sequences, $< q_i >$,

$$\frac{\sum_{i=1}^{n} f(x_i)}{n} - \int f d\mathcal{P}(\mathbf{x}) = \mathcal{O}(\frac{\log^{d-1} n}{n}) \tag{4}$$

where d is the dimension of the space over which the integral is being evaluated and n is the number of points at which f is evaluated. For large n, this

[2] In most cases, for fixed d, the value of (2) is not equal to the value of the integral over continuous paths, (1); however, it is the value of the discretized integral (2) which is most often desired.

[3] We will not define either quasi-random or pseudo-random formally; however, intuitively, a d-dimensional quasi-random sequence is a sequence of points in d-dimensional space which is distributed "uniformly" through the unit cube while in a pseudo-random sequence of points, the points are distributed "randomly".

approaches zero *almost* as fast as $1/n$. At first glance this suggests enormous potential; however, if we observe that for a typical financial application d might easily be 20 (e.g. a quarterly cap on a five year floating loan) and that the right hand side of (4) achieves its maximum when $n = e^{d-1}$, we see that the limiting behavior of (4) does not *guarantee* improvements for calculations of modest size. One goal of this paper is to determine experimentally whether, in realistic financial applications, quasi-random sequences realize this potential.

The remainder of the paper is organized as follows. In section 2, we discuss the problem of evaluating a MCS method in the context of financial calculations and describe the experimental methodology we have adopted. In section 3.1, we describe how we construct and incorporate different types of numbers into MCS, in section 3.2, we introduce a a stratified sampling technique for variance reduction using. In section 4, we compare simulations using Sobol sequences of quasi-random points with two sequences of pseudo-random points both with and without variance reduction. Our simulations use an assortment of exotic options with 8, 32, and 128 step time discretizations. We find that without variance reduction, Sobol sequences are superior to pseudo-random, but that the advantage decreases with dimension. We find that with variance reduction, the 3 sequence types are comparable. In fact, all of the "treated" sequences yield better results than the original Sobol sequences. The final section contains conclusions and directions for further work.

2 Methodology

The experiments reported here cover a variety of single underlying derivatives. Tests include underlyings with evolution governed by log normal SDE, Cox-Ingersoll-Ross (*CIR*) and log normal with mean reversion (*LMR*) processes. They include exotic stock options (with log normal underlying) and interest rate caps and floors (with CIR and LMR rate models). The exact form of the LMR model was taken from Black and Karasinski [BK91]. See figures 3-6 on page 11 for details of terminology, process, instrument specifications, and the full list of derivatives tested.

The manner of testing was chosen with the following factors in mind:

1. Desired accuracy. Will the method be used to calculate values to within a few percent or a few parts in 10^6? While Monte Carlo methods can be used for arbitrary accuracy, for valuation of financial instruments where the models are empirical and inexact, more than $2-3$ digits ($1-.1\%$) is not likely to be relevant.

2. Speed and accuracy. We consider both standard deviation and relative error as measures of "quality" since the goal of most calculations is to estimate a value to within a given per cent error with a given confidence.

3. Security and Robustness. How likely is the method to make a serious error? The likelihood of error depends on the particular application. If

a MCS is being used to value a large portfolio, the relevant question might be: What is the *expected behavior on the average contract*? If, however, the method is to be used in a more hostile environment, e.g. by a market maker to value customized contracts, one must ask: If a competitor were to learn the details of the algorithms used, would they be able take advantage of this information? Here the relevant question concerns the *expected behavior on the worst case contract*.

This last item is a consequence of the observation that no purely deterministic MCS can be secure in a hostile environment since for any fixed finite the set of paths there is a contract on which the method will have arbitrarily high relative error. This is particularly significant with respect to the manner in which quasi-random sequences are incorporated into MCS.

We compared 3 methods of point generation (see below for details) each with and without the variance reduction technique described in section 3.2. Our experiments consisted of a sequence of non-adaptive *experiments*. Each experiment examines the accuracy achieved by the various sequence types when simulating a fixed instrument using a fixed number of time steps with the points chosen prior to any function evaluation. Each experiment consisted of 6 *observations* (one for each basic method of point generation with and without variance reduction). Each observation consisted of 8 *runs*. Each run used distinct sequences (of the appropriate type) and consisted of either 256 or 1024 paths. For each observation using 256 path runs, $r_1, \ldots, r_8$, there was an observation using 1024 path runs, $r_1^*, \ldots, r_8^*$ with $r_i^* \supset r_i$. Similarly for each run r based on a sequence without variance reduction, there is a run r^* whose paths are obtained by applying variance reduction to the paths of r.

The value of a run was obtained by determining the present value of the instrument along each of the paths comprising the run and using the average of these values as the value of the run. The value of the observation is the average of the values of the runs, and the variance of the observation is the variance of these values[4].

The test results are presented as relative error and relative standard deviation. For each test, the true value of the instrument was taken as the value of a simulation using 2^{20} paths with the ESSL pseudo-random number generator, `durand`. For each sequence type, the ratio of the standard deviation or absolute error to the true value was determined. The average over instruments is reported.

We also computed normalized results in which for each test, the average standard deviation or absolute error was computed, and for each sequence type, the ratio of the value for the observation to this average was determined. The analysis of scaled data was intended to eliminate masking effects which might arise if a subset of instruments yielded large errors and/or standard deviations.

[4]We observe that each run yields its own variance. This variance reflects the underlying stochastic nature of the instrument and not the effect of finite sample size. It is the degree of variability due to finite sample size which we must measure in order to gauge the reliability of our estimate.

The analysis using this data yielded results consistent with the relative data described above and we have, therefore, not included these tables.

The results of our experiments are displayed in the graphs and tables on pages 10 through 12. The results for each point generation method are displayed based on the number of paths (256 or 1024) and the number of time steps (8,32, or 128) in the observation[5].

3 Sequences of paths

3.1 Quasi-random and pseudo-random sequences

Pseudo-random points When using pseudo-random numbers for MCS, there is general agreement about how to choose paths. A random number generator is selected, one or more "seeds" are chosen at random, and for each seed, the coordinates of one or more points are generated. The number of seeds used might reflect concern about imperfections in the generator or about security issues; however, the expected number of paths needed to achieve a desired accuracy is independent of the number of distinct seeds.

For pseudo-random number experiments, we used two generators from IBM's ESSL [IBM] (`durand` and `durxor`, a long sequence Tausworthe exclusive-or generator [Tau65]). One seed was used for each run. The data for these two sequences is reported as ESSL and Taus respectively.

Quasi-random points In terms of use in MCS, there are two factors which distinguish quasi-random and pseudo-random points. The first is, of course, the potential for high performance suggested by equation 4. The second relates to the effect which the use of multiple seeds has on performance. For quasi-random sequences a seed corresponds to an entire sequence (see (5) below). The performance hinted at by (4) refers to performance attained by a single sequence and therefore, even if this equation governs the convergence obtained by a single quasi-random sequence, the expected number of paths needed to achieve a desired accuracy is *not independent* of the number of distinct seeds used. This results in a conflict between achieving the highest performance and using randomization to improve security or to guard against possible imperfections in the generator.

Remark. A single quasi-random run, i.e. simulating and averaging with q points from a single quasi-random sequence, is equivalent to drawing a single sample from a distribution with mean equal to the true value and with *unknown variance*. Equation (4) offers hope but no guarantee, that for fixed q, the

[5] The caps and floors each had 12 payments and so were simulated for (12,32, and 128) time steps; however, the data for 12 step simulation is reported with the other 8 step data.

variance of this distribution is smaller than one might expect if one drew q points from a pseudo-random sequence.

Traub [PT95, PT96] has proposed simulation using points from a single quasi-random sequence. In this approach, a sequence, must perform well for all instruments. The problem with this stems from the asymptotic nature of the accuracy and performance guaranteed by (4). There is no guarantee that a simulation running for any fixed number of paths will converge quickly or, in fact, approach the right answer. In fact, as noted above, for any fixed finite set of paths, a derivative exists which the given set of paths values incorrectly. Such a derivative might be created by chance or by a competitor who had unexpected information about the single sequence being used.

The method tested here retains randomness in the use of quasi-random numbers. This will pay a price in performance; however, hopefully, the method will achieve some performance gains while minimizing concerns about robustness, security, and accuracy which arise if single sequence simulation is used.

Sobol sequences of points in $\Re^d$. A one dimensional Sobol sequence [Sob67, PTVF92], $S_q(i)$, is a sequence of real numbers generated from a primitive polynomial, q, in $GF(2)$. Sobol showed that given a $d-$tuple of distinct primitive polynomials, $\{p_i\}$, and any integer, L, the sequence of points

$$\{(S_{p_1}(L+j), \ldots S_{p_d}(L+j)) | j = 1 \ldots\} \tag{5}$$

is quasi-random.

For Sobol points, each run involved choosing a sequence of primitive polynomials and a skip length, L, and using the first n of the points defined by 5.

3.2 Variance reduction

Variance reduction methods reorganize the calculations of a MCS in order to obtain a more reliable estimate. Boyle *et al* [BBG95] reviews the application of MCS in finance and discusses a number of variance reduction methods. Kalos [KW86, p. 92] groups variance reduction methods into three broad classes.

1. Correlation Methods. These methods include both control variates, in which the existence of a similar instrument with known value is used to improve the accuracy of the desired instrument, and antithetic variables, in which samples expected to be negatively correlated are grouped together.

2. Importance sampling. These methods use additional samples in regions where the function is large or rapidly varying.

3. Stratified sampling. These methods collect related samples into groups with the goal of reducing the variance of each group and thereby reducing the variance of the full MCS.

The variance reduction method tested here is a form of stratified sampling based on mean and variance. As noted above, the MCS problem reduces to the evaluation of an d-fold integral over paths. Our approach to evaluating the integral is to use a change of variables and rewrite integral (2) as

$$\int_{-\infty}^{\infty}\int_{0}^{\infty}\left\{\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} v(\mathbf{y}_{\mu,\sigma})d\mathbf{y}_{\mu,\sigma}\right\} dX(\sigma)\, d\Phi(\mu) \qquad (6)$$

where μ ranges over the possible total displacements of a path, σ ranges over the possible variances of a path, the $\mathbf{y}_{\mu,\sigma}$ are paths with given mean and variance, and Φ and X are appropriate induced probability distributions. We evaluate the outer two integrals using appropriately chosen quadrature formulae, and use Monte Carlo to estimate the remaining iterated integral. Since there is no statistical error in the quadrature, we have reduced the dimensionality of the MCS by 2. Furthermore, for piecewise linear pay-off functions, we expect a strong correlation between the mean and variance of a path and the value of pay-off. If this is the case, we will find that the estimate for each (μ, σ) pair will have comparatively low variance. We note, however, that quadrature formula introduces non-statistical error which cannot be removed solely by increasing the number of paths used.

4 Simulation Results

Figures 1 and 2 page 10 show relative standard deviation and relative error respectively. Data for the base sequences is shown in the left half of the figures. (The numerical data is shown in Tables 1 and 2 page 12.)

This data shows:

1. Without variance reduction, at comparatively low dimension, quasi-random points yield lower standard deviations and errors than pseudo-random.
2. The advantage which quasi-random shows over pseudo-random at low dimension decreases as dimension rises. At dimension 128, the average standard deviation was only 30% less for Sobol sequences than for pseudo-random.
3. With variance reduction, the three sequence types are comparable.
4. The error and standard deviation achieved with variance reduction and 256 points is comparable to error achieved by 1024 quasi-random points.

5 Conclusions and Future Work

Our tests show that for use in simulations organized as described in section 2, at low dimension, with no variance reduction, quasi-random points are superior to pseudo-random. We find that for higher dimensional problems, the advantages gained through variance reduction overwhelm the advantage due to the type of the base sequence.

We note that aside from increased accuracy in these non-adaptive tests, the variance reduction method we describe offers a natural framework in which to use adaptive integration methods to further accelerate simulations. In Kalos' terminology, this would combine importance sampling with the stratified sampling described here. Possibile extensions along these lines include additional sampling or refinement at points with high variance, Richardson extrapolation, or using a minimum principle together with variance information to choose a more appropriate approximation for integration. We intend to pursue these topics in future work.

It is also important to emphasize what we have not done. We have not searched for a single sequence with special properties. We argued earlier that such a collection of paths cannot be used in a hostile environment, and, we feel that it would be imprudent to evaluate a portfolio of derivatives using only a single sequence. It is, however, possible that such a methodology might be acceptable in the context of evaluating a large portfolio for market or credit risk.

6 Acknowledgments

I would like to thank John Hosking for many valuable conversations and for excellent implementations of a number of special functions. I would also like to thank Jim Shearer and Charles Micchelli for valuable discussions.

References

[BBG95] Phelim P. Boyle, Mark Broadie, and Paul Glasserman. Monte Carlo methods for security pricing. Technical report, Columbia University, June 7 1995.

[BK91] Fischer Black and Piotr Karasinski. Bond Option Pricing when Short Rates are Lognormal. *Financial Analysts Journal*, pages 52–59, July-August 1991.

[Boy77] Phelim P. Boyle. Options: A Monte Carlo Approach. *J. of Financial Economics*, 4:323–338, 1977.

[IBM] IBM Corp. *Engineering and Scientific Subroutine Library, version 2*. SC23-0526-00.

[KW86] Malvin H. Kalos and Paula A. Whitlock. *Monte Carlo Methods, volume I: Basics*. Wiley-Interscience, 1986.

[Nie92] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Number 63 in CBMS-NSF Regional Conf. Series in Appl Math. SIAM, 1992.

[PT95] Spassimir H. Paskov and Joseph F. Traub. Faster valuation of financial derivatives. *Journal of Portfolio Management*, Fall 1995.

[PT96] Anargyros Papageorgiou and Joseph Traub. Beating Monte Carlo. *Risk Magazine*, 9, June 1996.

[PTVF92] William H. Press, Saul A. Tekovsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C, Second Edition.* Cambridge University Press, 1992.

[Sob67] I. M. Sobol. The distribution of points in a cube and the approximate evaluation of integrals. *USSR Comp. Math and Math Phys*, pages 86–112, 1967.

[Tau65] R. C. Tausworthe. Random number generators by linear recurrence modulo two. *Math. Comp..*, 1965.

[TW94] Joseph F. Traub and H. Wozniakowski. Breaking Intractability. *Scientific American*, pages 102–107, January 1994.

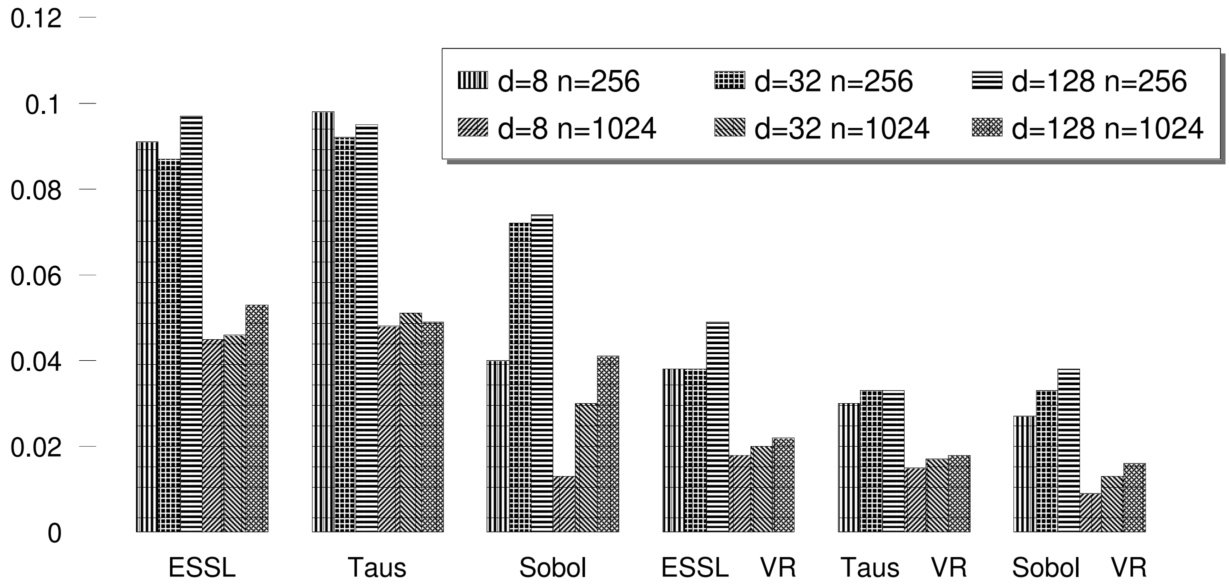

Figure 1: Average % standard deviation by sequence type. (d = time steps,n = points per run)

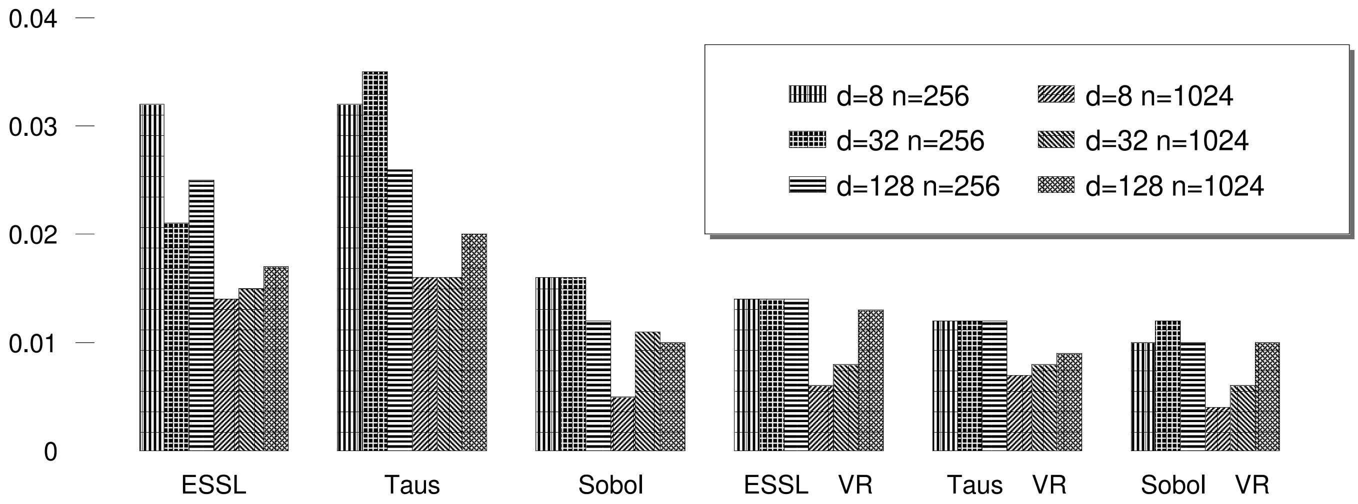

Figure 2: Average % absolute error by sequence type. (d = time steps,n = points per run)

	Underlying		Derivative
$P(t)$	Value of underlying at t.	K	Strike.
μ	Drift of the underlying.	T	Time to expiration in years.
σ	The volatility of the underlying.	$\mathcal{M}$	Number of monitoring times.
η	The mean reversion target.	$\mathcal{B}$	The barrier value.
d	Time steps/year.	R	The risk free rate.

Figure 3: Factors influencing evolution of underlying and derivative prices

1. 4 knock-out calls. $P(0) = K = 100$, $T = 1.0$, $\mathcal{B} \in \{80, 90, 95, 98\}$, $R = 0.0$, $\sigma = .3$.
2. 4 knock-in calls. $P(0) = K = 100$, $T = 1.0$, $\mathcal{B} \in \{102, 105, 110, 120\}$, $R = 0.0$, $\sigma = .3$.
3. 4 Lookbacks. Calls and puts. $T \in \{.25, .5\}$, $R = 0.10$, $\sigma = .2$.
4. 12 Extremas. Calls and puts. $K \in \{95, 100, 105\}$, $T \in \{.25, .5\}$, $R = .10$, $\sigma = .2$.
5. 2 Asian calls. One geometric and one arithmetic. $P(0) = 7.5$, $R = 0.0$, $\sigma = .5$.
6. 2 Asian rate calls. One geometric and one arithmetic. $P(0) = K = 7.5$, $R = 0.0$, $\sigma = .5$.
7. 12 interest rate caps and floors. One CIR and one LMR interest rate model. $P(0) = .08$, $K \in \{.09, .10, .12\}$(caps), and $K \in \{.07, .06, .04\}$(floors), $T = 3.0$ Both models used $\sigma = .3$, $\eta = .08$, $\mu = .1$, and $d \in \{12, 32, 128\}$
   (a) A CIR process: $dr = .1 \times (.08 - r) \times dt + .3sqrt(r) \times dX$
   During simulation, if the rate of the CIR became negative, it was set to 0.
   (b) A LMR process: $d(\log r) = .1 \times (\log .08 - \log r) \times dt + .3 \times dX$

Figure 4: The instruments tested. Except as noted, all instruments were values using $d \in \{8, 32, 128\}$.

Process	SDE	Discretized equation
Log Normal	$dP/P = \mu dt + \sigma dX$	$d(\log P) = (\mu - .5\sigma^2)dt + \sigma * \sqrt{dt} * \rho$
LMR	$d(\log P) = \mu(\log \eta - \log P)dt + \sigma P dX$	$d(\log P) = \mu(\log \eta - \log P)dt + \sigma\sqrt{dt} * \rho$
CIR	$dP = \mu(\eta - P)dt + \sigma\sqrt{P}dX$	$dP = \mu(\eta - P)dt + \sigma\sqrt{P}\sqrt{dt}\rho$

Figure 5: SDEs simulated. (ρ is chosen from a standard normal)

Instrument	payoff function
Vanilla	$\max(0, P(T) - K)$
CAP	$\{\text{Vanilla Call}(\tau) \mid \tau \in \mathcal{M}\}$
Asian	$\max(0, P(T) - \text{average}\{P(t) : t \in \mathcal{M}\})$
Asian rate	$\max(0, \text{average}\{P(t) : t \in \mathcal{M}\} - K)$
Lookbacks	$\max(0, P(T) - \min\{P(t) : t \in \mathcal{M}\})$
Extremas	$\max(0, \max\{P(t) : t \in \mathcal{M}\} - K)$
Knock-out	if $\exists t \in \mathcal{M}\{(P(t) - \mathcal{B}) \times (P(0) - \mathcal{B}) \leq 0\}$ then 0.0 else P(T)-K
Knock-in	if $\exists t \in \mathcal{M}\{(P(t) - \mathcal{B}) \times (P(0) - \mathcal{B}) \leq 0\}$ then P(T)-K else 0.0

Figure 6: Call instruments.

Sequence Type	8	32	128
ESSL 256	0.091 ± 0.0578	0.087 ± 0.0484	0.097 ± 0.0614
Taus 256	0.098 ± 0.0698	0.092 ± 0.0468	0.095 ± 0.0632
Sobol 256	0.040 ± 0.0386	0.072 ± 0.0802	0.074 ± 0.0664
ESSL 256 VR	0.038 ± 0.0291	0.038 ± 0.0312	0.049 ± 0.0603
Taus 256 VR	0.030 ± 0.0238	0.033 ± 0.0331	0.033 ± 0.0341
Sobol 256 VR	0.027 ± 0.0292	0.033 ± 0.0385	0.038 ± 0.0499
ESSL 1024	0.045 ± 0.0287	0.046 ± 0.0253	0.053 ± 0.0530
Taus 1024	0.048 ± 0.0343	0.051 ± 0.0306	0.049 ± 0.0345
Sobol 1024	0.013 ± 0.0118	0.030 ± 0.0274	0.041 ± 0.0354
ESSL 1024 VR	0.018 ± 0.0143	0.020 ± 0.0200	0.022 ± 0.0243
Taus 1024 VR	0.015 ± 0.0123	0.017 ± 0.0184	0.018 ± 0.0240
Sobol 1024 VR	0.009 ± 0.0105	0.013 ± 0.0123	0.016 ± 0.0187

Table 1: Average % standard deviation.

Sequence Type	8	32	128
ESSL 256	0.032 ± 0.0293	0.021 ± 0.0190	0.025 ± 0.0206
Taus 256	0.032 ± 0.0233	0.035 ± 0.0455	0.026 ± 0.0221
Sobol 256	0.016 ± 0.0205	0.016 ± 0.0263	0.012 ± 0.0145
ESSL 256 VR	0.014 ± 0.0147	0.014 ± 0.0143	0.014 ± 0.0144
Taus 256 VR	0.012 ± 0.0129	0.012 ± 0.0155	0.012 ± 0.0110
Sobol 256 VR	0.010 ± 0.0130	0.012 ± 0.0189	0.010 ± 0.0107
ESSL 1024	0.014 ± 0.0153	0.015 ± 0.0146	0.017 ± 0.0193
Taus 1024	0.016 ± 0.0150	0.016 ± 0.0189	0.020 ± 0.0289
Sobol 1024	0.005 ± 0.0052	0.011 ± 0.0162	0.010 ± 0.0130
ESSL 1024 VR	0.006 ± 0.0078	0.008 ± 0.0115	0.013 ± 0.0164
Taus 1024 VR	0.007 ± 0.0100	0.008 ± 0.0105	0.009 ± 0.0133
Sobol 1024 VR	0.004 ± 0.0039	0.006 ± 0.0071	0.010 ± 0.0139

Table 2: Average % absolute error

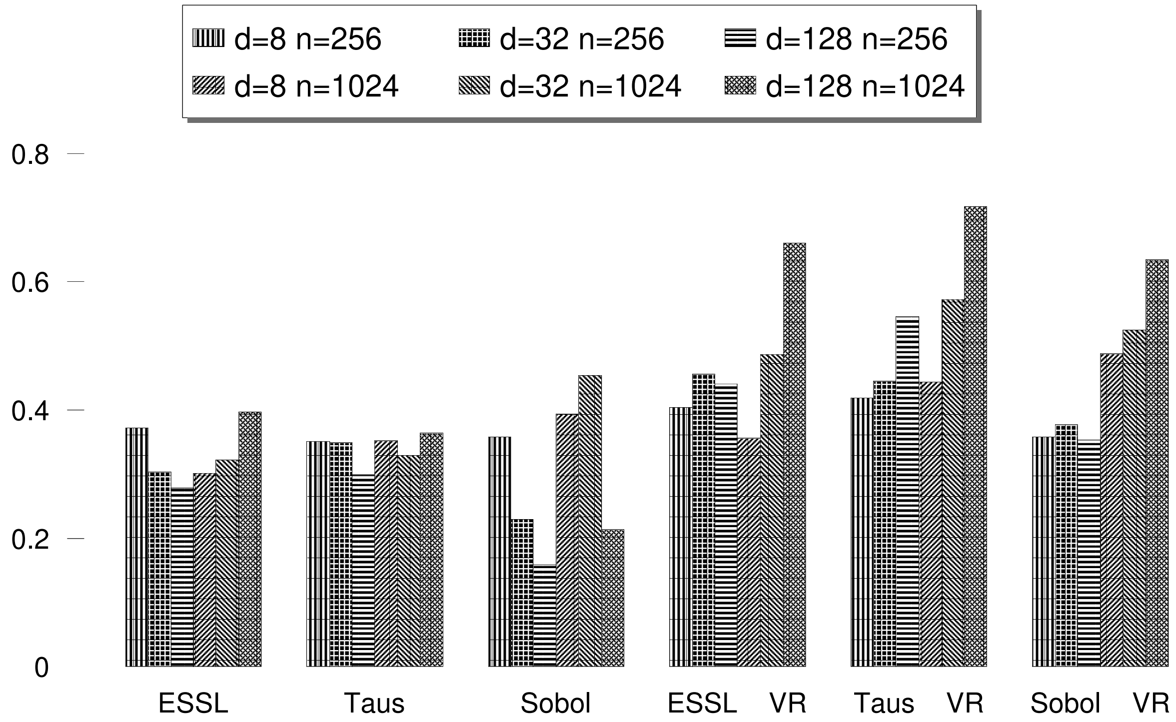

Figure 7: Average absolute error in standard deviations by sequence type. Exotic options. (d = time steps,n = points per run)