UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Generating Quasi-Random Paths for Stochastic Processes

William J. Morokoff

February 1997

CAM Report 97-6

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90024-1555

Generating Quasi-Random Paths for Stochastic Processes

William J. Morokoff
Mathematics Dept., UCLA
and C.ATS Software, Inc.
morokoff@math.ucla.edu

February 23, 1997

Abstract

The need to numerically simulate stochastic processes arises in many fields. Frequently this is done by discretizing the process into small time steps and applying pseudo-random sequences to simulate the randomness. This paper address the question of how to use quasi-Monte Carlo methods to improve this simulation. Special techniques must be applied to avoid the problem of high dimensionality which arises when a large number of time steps are required. Two such techniques, the generalized Brownian bridge and particle reordering, are described here. These methods are applied to a problem from finance, the valuation of a 30 year bond with monthly coupon payments assuming a mean reverting stochastic interest rate. When expressed as an integral, this problem is nominally 360 dimensional. The analysis of the integrand presented here explains the effectiveness of the quasi-random sequences on this high dimensional problem and suggests methods of variance reduction which can be used in conjunction with the quasi-random sequences.

1 Introduction

In many applications ranging from finance to fluid dynamics it is necessary to evaluate the expectation of a function of a random path generated by a stochastic process. For a continuous time process this expectation may often be

expressed as a Feynman-Kac type integral over Brownian motion. For numerical simulation, the continuous process is often modeled as a discreet process such that the expectation reduces to a standard integral weighted by the distribution function associated with each step of the discrete process. The time discretization often has physical relevance to the problem, as in the example presented below in which the time step of one month corresponds to monthly cash flows.

An important question is how to effectively numerically evaluate the integrals which arise in stochastic simulations. Because the time discretization often requires using a large number of small steps, the resulting expected value integrals are often very high dimensional (see [20, 1, 17, 21] for examples of such integrals in finance). This high dimensionality is generally dealt with by using Monte Carlo simulation, whereby N "random" paths associated with the stochastic process are generated from a pseudo-random number sequence and the desired integral is approximated by the average of the integrand evaluated along all the paths. The standard Monte Carlo method can be quite slow, however, because its convergence rate is only $O(N^{-1/2})$.

We consider here improvements to this method by generating the "random" paths with deterministic, quasi-random sequences. The resulting paths are in fact not random, but have superior distribution properties in the space of all paths associated with the given process, and thus lead to lower integration error. Such quasi-Monte Carlo methods can be much faster with errors approaching size $O(N^{-1})$ in optimal cases. This dramatic improvement in convergence rate has the potential for significant gains both in computational time and in range of application of Monte Carlo methods.

The effectiveness of quasi-Monte Carlo methods does have some important limitations. First, quasi-Monte Carlo methods are valid for integration problems, but may not be directly applicable to simulations, due to the correlations between the points of a quasi-random sequence. This problem can be overcome in many cases by writing the desired result of a simulation as an integral, as was mentioned above. However, as the resulting integral is often of very high dimension (e.g. dimension 360 for the example below), this leads to a second limitation: the improved accuracy of quasi-Monte Carlo methods is generally lost for problems of high dimension or problems in which the integrand is not smooth. This loss of effectiveness has been documented for a series of test problems in [12, 13, 14]. Several researchers in computational finance have recently reported great success with quasi-Monte Carlo computation of problems of very high dimension [1, 17, 21]. One purpose of this paper is to introduce techniques which effectively recast such nominally high dimensional problems into more moderate dimensional forms. These techniques allow for the range of application of quasi-Monte Carlo methods to be significantly extended, in particular when combined with other variance reduction methods.

The paper is organized as follows. First a brief review of quasi-random sequences in presented in Section 2. This is followed by the description of a general technique, the generalized Brownian bridge, for generating quasi-random paths

for standard Brownian motion in Section 3. An alternate approach for generating paths involving only a one dimensional sequence is described in Section 4. An example involving a 30 year coupon bond is presented and analyzed in Section 5, while Section 6 gives the numerical results for the various techniques. This is followed by conclusions.

2 Review of Quasi-Random Sequences

Quasi-Monte Carlo methods are based on the idea that random Monte Carlo techniques can often be improved by replacing the underlying source of random numbers with a more uniformly distributed deterministic sequence. Quasi-Monte Carlo methods often include standard approaches of variance reduction, although such techniques do not necessarily directly translate. Other approaches, such as described in Sections 3 and 4, are unique to quasi-Monte Carlo and have no effect when used with random sequences. The fundamental feature underlying all quasi-Monte Carlo methods, however, is the use of a quasi-random sequence. Therefore we now present a brief review of certain properties of such sequences.

Quasi-random sequences are often referred to as low discrepancy sequences. This term refers a measure of the uniformity of a sequence, the discrepancy, which is defined as follows. Consider a set of N points in the d dimensional unit cube $\{x_i\}$. The discrepancy of this set is

$$D_N = \sup_E \left| \frac{\#of x_i \in E}{N} - m(E) \right| . \tag{2.1}$$

Here E is a sub-rectangle of the unit cube, $m(E)$ is the volume of E, and the sup is taken over all such sub-rectangles. This definition is based on the idea that for any given rectangle, the percentage of points from a uniformly distributed set which lie in the rectangle should be close to the volume of the rectangle. Thus the more uniformly distributed, the smaller the discrepancy of a set.

A uniformly distributed infinite sequence of points in the d dimensional unit cube can them be defined [8] as a sequence for which

$$\limsup_{N\to\infty} D_N \to 0 . \tag{2.2}$$

Here the discrepancy is take to be the discrepancy (2.1) of the first N terms of the sequence. There are many sequences which have this property. For example, a uniform random sequence satisfies (2.2) almost surely. In fact, by the law of iterated logarithms, we have that for a random sequence the expected value of the discrepancy satisfies

$$E(D_N) \leq \frac{\log\log N}{\sqrt{N}} . \tag{2.3}$$

A quasi-random, or low discrepancy, sequence is one which satisfies the condition that

$$D_N \leq C_d \frac{\log^d N}{N}. \tag{2.4}$$

For a given function $f(x)$ defined on the unit cube and a set of N integration nodes $\{x_i\}$, the simple Monte Carlo integration error ϵ is defined as

$$\epsilon = \left| \int f(x)\, dx - \frac{1}{N} \sum_{i=1}^{N} f(x_i) \right|. \tag{2.5}$$

For functions of bounded variation, the Koksma-Hwalka inequality [16] states that

$$\epsilon \leq V(f)\, D_N, \tag{2.6}$$

where $V(f)$ is the variation of the function and D_N is the discrepancy of the set. This may be contrasted with the mean square error for random Monte Carlo

$$E(\epsilon^2) = \frac{\sigma^2}{N}, \tag{2.7}$$

where σ^2 is the variance of the integrand, defined as

$$\sigma^2 = \int f(x)^2\, dx - \left(\int f(x)\, dx \right)^2. \tag{2.8}$$

Inequalities (2.3) and (2.4) then suggest that low discrepancy sequences will lead to smaller integration error, at least for large enough N.

A substantial body of work has been devoted to developing low discrepancy sequences, much of which is described in Niederreiter's monograph [16]. Numerous other papers have focused on the applications of quasi-Monte Carlo methods [6, 5, 9, 12, 23, 18, 20]. In the present paper, we work with the Sobol' sequence [22, 24]. This is a sequence for which each individual dimension is a permutation (at every 2^k points, $k = 1, 2, \ldots$) of the Van der Corput sequence

$$\frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{5}{8}, \frac{3}{8}, \frac{7}{8}, \frac{1}{16}, \cdots \tag{2.9}$$

This is a very uniform one dimensional sequence, so that the one dimensional projections of the Sobol sequence are also quite uniform. The choice of permutations is crucial to ensure that higher dimensional projections, as well as the entire d dimensional sequence, are also uniformly distributed. As the dimension grows large, it becomes increasingly difficult to guarantee that all projections are well distributed, at least for relatively small N. This difficulty is reflected in the discrepancy bound (2.4)

3 The Generalized Brownian Bridge

We now consider a class of methods for generating random walks associated with stochastic processes of the form

$$dx = (a(t) + b(t)x)dt + \sigma(t)dz\,, \tag{3.1}$$

where dz is the derivative of standard Brownian motion and a, b and σ are integrable in time. These methods may lead to substantial error reduction when combined with quasi-random sequences. The presentation here deals with standard Brownian motion with no drift and constant variance. The extension to include the drift term of a mean reverting process is addressed in Section 5.

Since Brownian motion is a Markov process, it is most natural to generate a discrete time Brownian motion random walk $x_{n+1} = b(t_{n+1})$ at time $t_{n+1} = t_n + \Delta t$ as a random jump from its value $x_n = b(t_n)$ through the formula

$$x_{n+1} = x_n + \sigma\sqrt{\Delta t}\; z \tag{3.2}$$

in which z is sampled from $N(0,1)$. More generally, any future point x_m, $(m > n)$ may be generated by

$$x_m = x_n + \sigma\sqrt{(m-n)\Delta t}\; z\,. \tag{3.3}$$

Any point of the walk in the middle can then be generated from knowledge of the past, x_n, and the future x_m according to the *Brownian bridge* formula [7]

$$x_k = (1-\gamma)x_n + \gamma x_m + \sigma\sqrt{\gamma(1-\gamma)(m-n)\Delta t}\, z \tag{3.4}$$

where $n \le k \le m$ and

$$\gamma = \frac{k-n}{m-n}\,.$$

We remark that this formula is valid only for generating one step k between steps n and m, as any subsequently generated steps must be correlated with x_k. Equation (3.4) may continue to be used, however, by simply replacing one of the endpoints with the most recently generated point x_k. Note that variance of the random part of the Brownian bridge formula (3.4) for generating x_k is reduced by a factor $1/(1-\gamma)$ compared with the variance for generating x_k with formula (3.3).

The standard method of generating a random walk x_k is based on the updating formula (3.2). The initial value is $x_0 = 0$. Each subsequent value x_{k+1} is generated from the previous value x_k using formula (3.2) with independent normal variables z_k.

Another method, which we refer to as the *Brownian bridge discretization* can be based on (3.4). Suppose we wish to determine the path $x_0, x_1, \ldots, x_D$, and

for convenience assume that D is a power of 2. The initial value is $x_0 = 0$. The next value generated is $x_D = \sigma\sqrt{D\Delta t}\, z_1$. Then the value at the mid point $x_{D/2}$ is determined from the Brownian bridge formula (3.4) with $\gamma = 1/2$. Subsequent values are found at the successive mid-points; i.e. $x_{D/4}, x_{3D/4}, x_{D/8}, \cdots$, sweeping along the breadth of the domain at each level of refinement.

Although the total variance associated with each x_k in this representation is the same as in the standard discretization, the variance associated with the z_k is no longer constant. It has been redistributed so that much more of the variance is contained in the first few steps of the Brownian bridge discretization, while the later steps have significantly smaller variance due to the factor of 2 reduction in the variance arising in formula (3.4). This reduces the effective dimension of the random walk simulation, which increases the accuracy of quasi-Monte Carlo. Moskowitz and Caflisch [15] applied this method to the evaluation of Feynman-Kac integrals and showed the error to be substantially reduced when the number of time steps, which is equal to the dimension of the corresponding integral, is large.

The Brownian bridge approach allows for a great deal of generalization. Another possibility involves a rearrangement of the breadth-first discretization described above in a depth-first fashion, such that the x_k are generated in the following order:

$$x_0,\ x_D,\ x_{D/2},\ x_{D/4},\ x_{D/8},\ \ldots,\ x_1, x_{3D/4},\ x_{3D/8},\ \ldots,\ x_3,\ \ldots x_{(D-1)}$$

In fact, formula (3.4) provides the means for generating the steps of the random walk in any order desired. Moreover, the number of terms in the walk D, representing the dimension of the problem, need not be a power of two.

To formalize these extensions, we introduce now the *generalized Brownian bridge discretization.* The path of the random walk may be expressed as a vector

$$x = (x_1, \ldots, x_D)^T,$$

as may the independent random numbers

$$z = (z_1, \ldots, z_D)^T.$$

The standard method of generating the random walk sets $x_1 = z_1$, $x_2 = z_1 + z_2$, etc. This may be written in matrix notations as

$$x = \sigma A\, z \tag{3.5}$$

where the matrix A is defined as

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & \ldots & 0 \\ 1 & 1 & 0 & 0 & \ldots & 0 \\ 1 & 1 & 1 & 0 & \ldots & 0 \\ & & & \cdot & & \\ & & & \cdot & & \\ & & & \cdot & & \\ 1 & 1 & 1 & 1 & \ldots & 1 \end{pmatrix}. \tag{3.6}$$

The Brownian bridge discretization described above can also be seen as a linear combination of the z, so that there exist a matrix B such that $x = \sigma B\, z$. We define now the generalized Brownian bridge discretization to be any matrix B such that the paths $\sigma B\, z$ correspond to the same stochastic process as the paths $\sigma A\, z$. Because a Gaussian process is completely specified by its covariance, if the paths $\sigma B\, z$ and $\sigma A\, z$ have the same covariance, they will necessarily be sampled from the same process. The covariance of the paths x_A and x_B are given by

$$E(x_A x_A^T) = \sigma^2\, AA^T \tag{3.7}$$

$$E(x_B x_B^T) = \sigma^2\, BB^T \tag{3.8}$$

Thus the matrix B will correspond to a generalized Brownian bridge discretization if and only if

$$B\, B^T = A\, A^T . \tag{3.9}$$

It is important to remember that random Monte Carlo methods will not be affected by how the random walk path is generated. From the integration point of view, this follows from the fact that for any function $f(x)$, under the change of variables $z' = B^{-1}Az$, we have that

$$E(f(x)) = E(f(Az)) = E(f(Bz')) \tag{3.10}$$

for any B satisfying (3.9). In particular, the variance of a given function of the path, which is expressed as an integral, is independent of the path generating matrix, so the random Monte Carlo error is also not affected.

As demonstrated in [15, 2], the combination of quasi-random sequences and the Brownian bridge discrectization can lead to significant error reduction. One can imagine that if the last step of the walk is more uniformly distributed, then the set of paths so generated is necessarily more uniformly chosen from the space of all possible paths, leading to smaller integration errors. More generally, properties of the integrand may be used to construct paths that more uniformly sample the space which has the greatest influence on the integrand. This idea is explored more in Section 5.

The first Brownian bridge discretization described above is an example of an important sub-class of the generalized discretizations, namely those that concern generating the steps of the walk sequentially according to a specified permutation $\Pi = (\pi_1, \ldots, \pi_D)^T$ of the first D integers. The unique Brownian bridge matrix B_Π corresponding to this permutation may be generated as follows. Let P be the permutation matrix defined by $\Pi = P \cdot (1, \ldots, D)^T$. Then

$$B_\Pi = P\, R_\Pi \tag{3.11}$$

where R_Π is the unique lower triangular matrix obtained from the Cholesky decomposition

$$R_\Pi\, R_\Pi^T = P^T\, A\, A^T\, P \tag{3.12}$$

where again A is given by (3.6). It is easily checked that B_Π satisfies (3.9).

The Brownian bridge formula (3.4) shows that each term of a permutation defined discretization may be express as a linear combination of exactly two previously determined steps of the path. Thus the path x may be generated recursively in $\mathcal{O}(D)$ steps. On the other hand, the matrix R_Π will in general be a dense lower triangular matrix, so that generating $x = \sigma B_\Pi z$ will be an $\mathcal{O}(D^2)$ operation. This may lead to a significant increase in the computation time necessary to generate the paths x. We prove now in this matrix formulation that the vector x may in fact be generated recursively in $\mathcal{O}(D)$ operations, and thereby indicate how to generate the necessary coefficients.

We observe first that the relationship $x = \sigma B_\Pi z$ may be rewritten with the help of (3.11) as

$$R_\Pi^{-1}\,(P^T x) = \sigma z\,. \tag{3.13}$$

We may easily obtain x from $P^T x$ in $\mathcal{O}(D)$ steps, and the following lemma shows that system (3.13) may also be solved in $\mathcal{O}(D)$ operations.

Lemma 3.1 *Let the matrix A be defined by (3.6), P be a permutation matrix, and R be the unique lower triangular matrix satisfying the Cholesky decomposition in (3.12). Then R^{-1} is a lower triangular matrix with at most three non-zero entries in each row.*

Proof: That R^{-1} exists and is lower triangular follows from A being invertible and the fact that the inverse of a triangular matrix is triangular. Inverting Equation (3.12) leads to the relationship

$$M_D\, M_D^T = S_D \tag{3.14}$$

where $M_D^T = R^{-1}$, so that M_D is upper triangular, and

$$S_D = P^T\,(A\,A^T)^{-1}\,P\,. \tag{3.15}$$

It is easily computed that

$$(A\,A^T)^{-1} = \begin{pmatrix} 2 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 1 \end{pmatrix}. \tag{3.16}$$

Therefore S_D is a symmetric matrix for which each row and column has at most three non-zero entries, and that one of these three occurs at the diagonal element.

We now decompose M_D in terms of the $(D-1)\times(D-1)$ upper triangular matrix M_{D-1}, the $D-1$ vector V_{D-1} and the scalar element m_D as

$$M_D = \begin{pmatrix} M_{D-1} & V_{D-1} \\ 0 & m_D \end{pmatrix}. \tag{3.17}$$

As M_D is invertible, m_D is necessarily non-zero. We also decompose S_D in terms of the $(D-1)\times(D-1)$ symmetric matrix T_{D-1}, the $D-1$ vector W_{D-1} and the scalar element s_D as

$$S_D = \begin{pmatrix} T_{D-1} & W_{D-1} \\ W_{D-1}^T & s_D \end{pmatrix}. \tag{3.18}$$

The matrix T_{D-1} shares the same property of S_D that no row or column contains more than three non-zero elements and that one of the three occurs at the diagonal element.

Equation (3.14) can then be written

$$\begin{pmatrix} M_{D-1}\,M_{D-1}^T + V_{D-1}\,V_{D-1}^T & m_D V_{D-1} \\ m_D V_{D-1}^T & m_D^2 \end{pmatrix} = \begin{pmatrix} T_{D-1} & W_{D-1} \\ W_{D-1}^T & s_D \end{pmatrix}. \tag{3.19}$$

Thus $m_D^2 = s_D$, and therefore $s_D \neq 0$. Because the last column of S_D contains no more than three non-zero entries, the vector W_{D-1} necessarily contains no more than two non-zero entries. Hence, V_{D-1} must also contain no more than two non-zero entries, which occur, say, in rows k_1 and k_2. Define now the matrix

$$S_{D-1} = T_{D-1} - V_{D-1}\,V_{D-1}^T. \tag{3.20}$$

The rank one matrix contains at most four non-zero entries, two of which occur on the diagonal. The other two occur in rows k_1 and k_2. As T_{D-1} already had non-zero diagonal entries, the diagonal elements of the rank one matrix will not change the count of zeros. The off-diagonal of the rank one matrix may introduce one additional non-zero term to each row k_1 and k_2. However, as there were at most three non-zero elements in rows k_1 and k_2 of S_D, and one of these appeared in the last column, this leaves at most two non-zero elements for each row k_1 and k_2 of T_{D-1}. Therefore S_{D-1} is a symmetric matrix with at most three non-zero elements in each row and column, one of which occurs in the diagonal position.

This argument shows that the last column of M_D, composed of the vector V_{D-1} and m_D, contains at most three non-zero elements. The same argument may now be repeated on the equation

$$M_{D-1}\,M_{D-1}^T = S_{D-1} \tag{3.21}$$

to show that the second to last column of M_D (i.e., the last column of M_{D-1}) contains at most three non-zero elements. This process may be repeated until

the matrix M_3 is reached, as which point the statement of at most three non-zero elements follows by definition.

This proves that each column of M_D has at most three non-zero elements, and therefore the same is true of each row of R^{-1}.

The proof suggests an efficient algorithm for generating the recursion coefficients stored in R^{-1} given the permutation vector Π. The non-zero entries of R^{-1} may be stored in a $D \times 3$ array, while the indices of the non-zero off-diagonal elements may be stored in a $D \times 2$ array. Once generated, these arrays are easily applied to solve the systems

$$R^{-1}\, y = \sigma z \qquad , \qquad x = P\, y\,. \tag{3.22}$$

A permutation based Brownian bridge discretization has the interpretation that the individual steps of the random walk x are generated is a specific order. Equation (3.9) allows, however, for a more general interpretation. Using the fact that B satisfies (3.9) if and only if $B = A\,Q$ for some orthogonal matrix Q, we see that generating the random walk as $\sigma B\, z$ is equivalent to applying A to an orthogonal transformation of z. Such a transformation may lead to a diagonalization of the integrand, concentrating much of the variance of the problem into a few principle directions, and thereby reducing the effective dimension. This is discussed further in the Section 5. There is a computational price to be paid for this approach in that, for a general orthogonal matrix Q, it will not be possible to generate the corresponding random walk $\sigma B\, z$ by recursion, but will require an $\mathcal{O}(D^2)$ procedure.

4 Reordering

For many problems associated with a stochastic process, there is a natural underlying time discretization. For example, for the bond described in the next section, monthly coupon payments suggest that monthly time steps are the most convenient to work with (leading to a 360 dimensional integral). However, this division and the corresponding dimension of the expected value integral are somewhat arbitrary. For example, daily time steps could be taken (representing the daily fluctuations of interest rates), which would lead to an integral of with dimension close to 11000. There would be little noticeable difference in the value of these integrals as long as the discretization error for the stochastic process is small. It is often the case that the acceptable error of the Monte Carlo computation is much larger than the time discretization error. While a minimum of 360 random variables are required to simulate the bond cash flows, it may be more correct to consider this as a one dimensional problem (corresponding to the use of a one interest rate model) which evolves in time.

With this in mind, we now describe an alternate approach to generating random walk paths. We consider a collection of N "simulation particles" x_j.

The position of particle j ($j = 1, \ldots, N$) at time step $n+1$ is determined by sampling from the distribution generated by the stochastic process with initial data x_j^n. The approach is described here for standard Brownian motion, but is easily adapted to all stochastic processes, making it a more general approach than the Brownian bridge discretization. The particle positions at time $n+1$ may be expressed as

$$x_j^{(n+1)} = x_j^{(n)} + \sigma\sqrt{\Delta t} z_j \tag{4.1}$$

where z_j is sampled from a $N(0,1)$ distribution. The ensemble of particles evolves in time according to (4.1).

If the z_j at each time step are sampled randomly, then this procedure is equivalent to the simple Monte Carlo generation of the random walk. Likewise, if z_j at time step d is taken from the d^{th} dimension of a D dimensional quasi-random sequence, then the procedure is identical to the simple quasi-Monte Carlo generation of the random walk.

In light of the fact that the problem is "spatially" one dimensional, it is possible to use the same one dimensional quasi-random sequence to generate each time step. Because the terms of the sequence are not independent, however, it is necessary to take special measures to insure that correlations across time steps are controlled and to make optimal use of the greater uniformity associated with one dimensional quasi-random sequences. These goals are achieved through the additional step of reordering the particles at each time step. This approach was developed by Lecot [9, 10, 11] for use in the simulation of kinetic equations, and has been applied to the periodic one dimensional heat equation, for which convergence close to $\mathcal{O}(1/N)$ was proved [12].

This method proceeds as follows for a one dimensional stochastic process. The particles are assigned positions according to the initial distribution function at time 0 such that

$$x_1^0 \leq x_2^0 \leq \ldots \leq x_N^0 . \tag{4.2}$$

The particles may all start from the same initial value (as in the bond example of the next section), or these initials values may be generated using a uniform cell centered discretization of $(0,1)$ with N points. The positions of the particles after the first time step are generated according to Equation (4.1) whereby the N numbers z_j correspond to the first N terms of a one dimensional quasi-random sequence (the Van der Corput sequence is a natural choice). The results is that the particles positions are no longer ordered according to (4.2). The final step is to reorder the particles by sorting and relabeling them, such that the particle labeled with index 1 is always furthest to the left, and the particle labeled with index N is always furthest to the right. The next time step is then carried out by assigning the next N terms of the quasi-random sequence in order to the particles $x_1^1, \ldots, x_N^1$. It is easily seen that for a random sequence, this method will have no effect and will lead to the same results as the standard Monte Carlo random walk.

The reordering procedure has two effects. First it breaks correlations associated with the low discrepancy sequence. If no relabeling were done, the first particle would be assigned the terms $(1, N+1, 2N+1, \ldots)$ from the sequence. If the Van der Corput sequence is used and N is even, this particle would always received a $z_1 \geq 0$ (assuming a direct mapping of $(0,1)$ to $(-\infty, \infty)$ is used obtain normally distributed numbers) to generate it's next position, with the result that the path for the first particle would always be increasing. The other particles would have similar problems. Reordering effectively decouples the generating value a particle receives from its position in the sequence. For this reason, it is also advisable to avoid using N equal to a power of two (assuming a base 2 quasi-random sequence is used) to further minimize structural correlations of the sequence.

The second effect helps explain the advantage this procedure offers. If the distribution function at time step n is well represented by the N particle positions, then in general there will be many particles in intervals which are relatively small compared with changes in the distribution function. Thus those particles in a given interval are essentially alike. Reordering groups these particles together and ensures that these like particles receive a contiguous subsequence of the quasi-random sequence, which itself is uniformly distributed. The effect is to obtain a more uniform sampling from the distribution function at time step $n+1$ than would have been obtained from a random process.

There are several limitations to this quasi-Monte Carlo method. First, it does not immediately generalize to higher dimensional stochastic process because there is no absolute ordering of two or more variables. Partial orderings are feasible in two or three dimensions, but become less effective as dimension increases. A decomposition of the stochastic process into principal components could however be used to identify two or three "directions" to which this procedure could be applied, while the other components are simulated with a random sequence.

A second requirement for the success of this method is that there be enough particles to adequately represent the distribution function in the sense described above. If the particles become separated so that their paths do not cross, then the reordering will have no effect. For Brownian motion this will eventually happen, but on time scales such that the diffusion is small in relation to the number of particles, this is not a problem. Processes with mean reversion help contain this effect.

Finally, based on the computational results present in Section 6, this method appears to only exhibit superior convergence properties compared with random Monte Carlo for functions which are predominantly the sum of univariate functions. A detailed discussion of the decomposition of functions into lower dimensional components and effective dimensionality for quasi-Monte Carlo integration can be found in [2, 19]. For example, consider $x = (x_1, \ldots, x_D)$ as a path sampled from a standard discrete Brownian motion process with a fixed

starting point x_0. The function

$$f(x) = \sum_{i=1}^{D} (x_i - x_{i-1})^2 = \sum_{i=1}^{D} z_i^2 , \tag{4.3}$$

where the z_i are the independent increments, will exhibit low integration error and convergence close to $\mathcal{O}(1/N)$ when reordering is used. However, the introduction of cross terms (e.g., $f(x) = x_D$) results in much larger error and a significant reduction in the convergence rate.

For appropriate stochastic processes, particle reordering may be combined with a permutation based Brownian bridge discretization. The effectiveness of this combination depends on how well the Brownian bridge discretization, when viewed as a change of variables, diagonalizes the integrand into the sum of univariate functions of the underlying independent normal variates. In the example above, the function $f(x) = x_D$ can be easily mapped to a single univariate function by using a permutation with $\pi_1 = D$. When such a Brownian bridge is used, the question arises as to what is the appropriate quantity to reorder. The answer is given by expressing the d^{th} path position of the i^{th} particle as

$$x_i^d = a^d z_i + b_i^d . \tag{4.4}$$

Here z_i is the normal variate used to determine x_i^d, and a^d is a constant. The quantity b_i^d depends only on the portion of the path of particle i which has already been determined because the matrix R^{-1} defined in Section 3 is lower triangular. The correct application of the reordering method here is to reorder the quasi-random normal samples $(z_1, \ldots, z_N)$ according to the ordering of the b_i^d.

5 Application to a Mean Reverting Process

5.1 Bond Valuation for a Stochastic Interest Rate

We now investigate the effectiveness of the quasi-Monte Carlo techniques outlined here by applying them to a problem from finance involving a mean reverting stochastic process. The problem is to find the fair price of a 30 year bond with a face value of \$100. which pays a monthly coupon of 1% (i.e., \$1). This question is closely related to pricing more complicated financial instruments such as mortgage backed securities to which quasi-Monte Carlo methods have also been applied [20, 1, 17, 21]. The simpler bond has been chosen here to help clarify the exposition.

For our purposes, all months will be considered of equal length, so that the 30 year bond has 360 payments of \$1 which are equally spaced in time. There is an additional payment of the face value of the bond, \$100, which occurs at month 360. The present value PV of the bond is simply the sum of all future cash flows discounted back to the present to account for interest accrual:

$$PV = \sum_{k=1}^{360} p_k \, Z_k \, . \tag{5.1}$$

Here p_k and Z_k are the payment and discount factor at month k, respectively. For the bond, $p_k = 1$ for $1 \leq k \leq 359$ and $p_{360} = 101$. Z_k is the price of a \$1 zero-coupon bond which expires at month k; in other words, Z_k is the value today of \$1 paid at month k. This is directly related to the annualized k-month continuously compound interest rate R_k by the formula

$$Z_k = \exp(-R_k T_k) \qquad\qquad T_k = k/12 \, . \tag{5.2}$$

The question thus becomes how to model and compute the discount factors.

In previous studies of quasi-Monte Carlo methods, interest rates have generally been modeled as following geometric Brownian motion. This has the advantage of ensuring that the rates never go negative, but it does not capture the fact that interest rates tend to vary over a limited range and never drift off to large values. For the current work we choose the Vasicek model for the short term rate r to illustrate the flexibility of the techniques proposed, as well as because this model has an exact analytic solution for the discount factors which aids in assessing the computational error of the various methods under consideration. The Vasicek model [25] is a mean reverting process with constant reversion speed a, reversion level b and volatility σ (standard deviation):

$$dr = a(b - r)dt + \sigma dz \qquad\qquad r(0) = r_0 \, . \tag{5.3}$$

Here dz is the derivative of standard Brownian motion.

According to the arbitrage pricing theory [4], the discount factor at time t must be given by

$$Z(t) = E\left(\exp(-\int_0^t r(s)\, ds)\right) \tag{5.4}$$

where the integral is taken along a given path, and the expectation $E(\cdot)$ is taken over all paths generated by the stochastic process (5.3). Because the variable $r(t)$ has a Gaussian distribution, and because the sum of Gaussian variables is also Gaussian, it is possible to evaluate Equation 5.4 analytically. The result is [4, 25]

$$\begin{aligned} Z(t) &= A(t)\, \exp(-r_0 B(t)) \\ B(t) &= (1 - \exp(-at))/a \\ A(t) &= \exp(b(B(t) - t) - \sigma^2(aB^2(t) + 2B(t) - 2t)/(4a^2)) \, . \end{aligned} \tag{5.5}$$

A Monte Carlo evaluation of the present value of the bond price would involve simulating interest rate paths according to the process (5.3). For a given path realization $\hat{r}(t)$, the discount factor for that path could be computed at any time t by evaluating

$$\hat{Z}(t) = \exp(-\int_0^t \hat{r}(s)ds)\,. \tag{5.6}$$

The bond price conditioned on that realization could then be computed from Equation 5.1. The true bond price is then the expectation of the present value, which is approximated in Monte Carlo by

$$PV \approx \frac{1}{N}\sum_{i=1}^{N} PV_i \tag{5.7}$$

when N interest paths with present values PV_i have been simulated.

In order to evaluate the integral 5.6 it is necessary to discretize time. (For a general stochastic process, integration of the stochastic differential equation to obtain a path $\hat{r}(t)$ will also require a time discretization approximation. However, for the Vasicek model, as well as for geometric Brownian motion, this may be done analytically.) The natural discretization is to choose time steps equal to one month, so that the discount factor at month k is related to the sum of interest rates at the previous months.

As the purpose of the current work is to study the effectiveness of various Monte Carlo techniques, we wish to separate the Monte Carlo error from the time discretization error. We will therefore take as the exact solution not Equations 5.6, but that of the time discretized version of Equation 5.4:

$$Z_M(t) = E\left(\exp(-\Delta t \sum_{i=0}^{M-1} r(t_i))\right) \tag{5.8}$$

where $\Delta t = t/M$ and $t_i = i\Delta t$. As long as $\max(a, b, \sigma/\sqrt{a})\,\Delta t << 1$, the approximation will be valid. The exact evaluation of this expectation is given by

$$\begin{aligned}
Z_M(t) &= A_M(t)\,\exp(-r_0\gamma B(t)) \\
B(t) &= (1-\exp(-at))/a \\
A_M(t) &= \exp(b(\beta\gamma B(t) - t + \Delta t) - \sigma^2\gamma(1+\beta)\left(\sum_{i=0}^{M-1}(1-\beta^i)^2\Delta t\right)/(4a^2)) \\
\beta &= \exp(-a\Delta t) \\
\gamma &= \frac{a\Delta t}{1-\beta}
\end{aligned} \tag{5.9}$$

It can be seen that in the limit as $M \to \infty$ ($\Delta t \to 0$), $Z_M(t) \to Z(t)$.

For the purpose of pricing the bond, we set $t = 30$ years and $\Delta t = 1/12$ years, so that number of steps of the random walk is $D = 360$. The standard approach to generating a random path corresponding to the process (5.3) is given by the recursion

$$r_{n+1} = b + (r_n - b)\beta + \widehat{\sigma} z_n \tag{5.10}$$

where r_n is the interest rate at time $n\Delta t$, β is given above (5.10), $\widehat{\sigma}$ is defined as

$$\widehat{\sigma} = \sigma \sqrt{\frac{1-\beta^2}{2a}}, \tag{5.11}$$

and z_n is an $N(0,1)$ variate. Note that $\widehat{\sigma} = \mathcal{O}(\sqrt{\Delta t})$ as $\Delta t \to 0$. In matrix notation, the entire path $r = (r_1, \ldots, r_D)$ may be expressed as

$$r = \widehat{\sigma} A z + \bar{r}. \tag{5.12}$$

Here z is a vector of independent $N(0,1)$ variates and

$$\bar{r}_n = b + (r_0 - b)\beta^n \tag{5.13}$$

with r_0 being the rate at time zero. The lower triangular matrix A is defined as

$$\begin{aligned} A_{ij} &= \beta^{i-j} \qquad i \geq j \\ A_{ij} &= 0 \qquad\quad i < j \end{aligned} \tag{5.14}$$

Note that this reduces to standard Brownian motion when the mean reversion speed a is zero (so that $\beta = 1$).

The generalized Brownian bridge ideas developed in Section 3 remain the same. The goal is to find a suitable matrix B with

$$BB^T = AA^T \tag{5.15}$$

such that the paths generated by

$$r = \widehat{\sigma} B z + \bar{r}. \tag{5.16}$$

have better distribution properties when z is a quasi-random vector of $N(0,1)$ samples. As with standard Brownian motion, permutations of the order in which the points on the path are generated leads to an interesting subset of the possible B's. The proof in Section 3 that a permutation based Brownian bridge may be generated in $\mathcal{O}(D)$ operations carries over exactly to the mean reverting process because the matrix $(A^T A)^{-1}$ with A given by (5.15) is also tri-diagonal. Here the diagonal elements are all equal to $1 + \beta^2$, while the non-zero off-diagonal elements are all $-\beta$.

The Brownian bridge selected for pricing the bond was based on the permutation $(360, 180, 90, 270, 45, 135, 225, 315, 15, 30, 60, 75, \ldots)$. This has the desirable properties that the early points are well spread through the whole of

the sequence, and that the time scale between subsequent points decreases as more points are added. For a financial instrument such as a bond for which the payments are distributed throughout its life, this kind of general permutation works well. There are numerous other similar orderings that could be chosen. Results, however, are likely to be insensitive to reasonable variations.

For the simulation we chose the annualized parameters $a = .32$, $b = .07$, $\sigma = .01$ and initial interest rate $r_0 = .12$. This corresponds to a long term annual interest rate of 7% and a long term annual volatility of 1.25%. With the Δt given above, $\widehat{\sigma} = 0.0028$. Based on Equations 5.1,5.8 and 5.10, the present value of the bond is 143.2973925856, which corresponds to an annual yield of $y = 8.14\%$. The variance in this value is computed as 80.3, while the antithetic variance (described below) is 0.415. The Macaulay duration [3] of this bond, defined as $(\sum_{k=1}^{360} k p_k Z_k^*)/PV$ where $Z_k^* = (1 + y/12)^{-k}$ is the discount factor corresponding to the yield, is 128 months, or 10.7 years. This is the average time to maturity weighted by the present values of the cash flows, and it functions as a measure of the bond price's sensitivity to changes in the yield.

5.2 Analysis of the Integrand

When random walk processes are used to value functions which smoothly depend on the path, a useful approach for understanding the effectiveness of quasi-Monte Carlo, as well as for providing control variates and indications for selecting a generalized Brownian bridge, is to compute a Taylor series expansion. It is natural (particularly for mean reverting processes) to expand the integrand as a function of independent normal increments about the expected mean, which corresponds to the point $z = (0, 0, \ldots, 0)$. As long as the variance of the process is not too large, the Gaussian weights in the expectation integral will cause a sharp drop off in the weighting of paths away from the mean, so that the Taylor expansion is likely to be a good approximation.

The choice of the integrand to be expanded depends of how the random walk is generated. As the default, we take the function $v_A(z) = PV(z)$ which corresponds to the standard random walk generated by (5.12), with A given by (5.15). Denoting the gradient term evaluated at zero by

$$\nabla v_A^0 = \left(\frac{\partial v_A}{\partial z_1}, \ldots, \frac{\partial v_A}{\partial z_D} \right) \Big|_{z=(0,\ldots,0)} \tag{5.17}$$

and the Hessian matrix evaluated at zero by

$$(H_A^0)_{ij} = \frac{\partial^2 v_A}{\partial z_i \, \partial z_j} \Big|_{z=(0,\ldots,0)}, \tag{5.18}$$

we have that

$$v_A(z) = v_A(0) + (\nabla v_A^0) \cdot z + \frac{1}{2} z^T H_A^0 \, z + \ldots . \tag{5.19}$$

Integrating (5.19) against the Gaussian measures from which the z_i are sampled gives

$$E(v_A(z)) = v_A(0) + \frac{1}{2}\Delta v_A(0) + \mathcal{O}(\widehat{\sigma}^4)\,. \tag{5.20}$$

Here $\Delta v_A(0)$ is the Laplacian of v_A evaluated at zero. All of the z_i are scaled by $\widehat{\sigma}$, so that every derivative of $v_A(z)$ introduces a factor of $\widehat{\sigma}$. Thus the Laplacian is of size $\widehat{\sigma}^2$, and the remainder term is $\mathcal{O}(\widehat{\sigma}^4)$. Note that the terms with odd powers of z integrate to zero by symmetry. Direct evaluation shows that for the choice of constants given above,

$$v_A(0) = 142.758 \qquad \Delta v_A(0) = 1.07546\,. \tag{5.21}$$

Evaluating (5.20) to second order gives an approximation with a relative error of less than 2e-5. The constant term alone accounts for all but 0.4% of the value of the integral. We use this information to interpret the effectiveness of various Monte Carlo and quasi-Monte Carlo techniques.

These results suggest that a method which accurately evaluates the linear and quadratic terms of the Taylor expansion will produce small relative error. We consider now how to deal with these terms.

The linear term integrates to zero. This may be computed with no error by using antithetic random (or quasi-random) variables. This requires that when a point z is sampled from the distribution and used to evaluate $v(z)$, the point $-z$ is also taken. This is equivalent to replacing the integrand by

$$v^*(z) = \frac{v(z) + v(-z)}{2}\,. \tag{5.22}$$

As mentioned above, the variance of the present value of the bond has been computed in the Monte Carlo calculation to be 80.3. A similar estimate can be obtained from the Taylor expansion as $||\nabla v_A(0)||^2 = 78.8$. The variance of the antithetic integrand $v^*(z)$ may also be estimated with a Monte Carlo calculation as 0.415, which is roughly 200 times smaller. Thus for random Monte Carlo, the use of antithetic variables leads to a reduction of error by a factor of around 14. There is of course the additional computational time associated with evaluating both $v(z)$ and $v(-z)$ to consider; however, for the bond valuation integrand, the increase is less than 25%.

It is natural to ask how a quasi-random sequence will perform in evaluating the linear term. The integral of this term is approximated by

$$E(\nabla v_A^0 \cdot z) \approx \nabla v_A^0 \cdot \Sigma \tag{5.23}$$

where the elements of the vector Σ are the one dimensional averages of the sample points

$$\Sigma^{(d)} = \frac{1}{N}\sum_{i=1}^{N} z_i^{(d)}\,. \tag{5.24}$$

If antithetic variables are used (so that the sample size is $2N$), then $\Sigma = 0$. If a quasi-random sequence is used without antithetic variables, the dimensions decouple according to (5.23) such that the errors are the sum of the one dimensional errors of (5.24). These errors will in general be $\mathcal{O}(1/N)$, with the constant roughly like $||\nabla v_A^0||$. For small values of N, the error from the linear term will tend to dominate the other errors and convergence close to $\mathcal{O}(1/N)$ may appear. For the current problem, however, the cross over from the dominant linear term error to the dominant higher order term error occurs fairly soon. If, for example, the second order terms come in as random errors with the antithetic variance describing the error size, these errors would begin to dominate around $N = 200$. Particle reordering is appropriate for such linear functions, as they are the sum of univariate functions. However, the relatively large contribution of the second order terms may limit the effectiveness.

We now consider the quadratic term of the Taylor expansion. It is clear that the use of antithetic variables will not lead to any error reduction for this term. It is possible to imagine constructing a set of antithetic points in D dimensions reflecting symmetries across all coordinate axes. In this way, the cross terms of the quadratic could be eliminated; however, the diagonal terms would remain. Moreover, this would require evaluating the function at 2^D points, as opposed to just two (z and $-z$).

An alternative approach to diagonalizing the quadratic term, and thereby decoupling the dimensions to produce a set of one dimensional integrals, is offered by the Brownian bridge discretization of the random walk. Let Q_A be the orthogonal matrix which diagonalizes the Hessian matrix H_A such that the elements of the resulting diagonal matrix appearing in order of decreasing absolute value. We have that

$$D_A = Q_A^T H_A Q_A . \tag{5.25}$$

A generalized Brownian bridge discretization matrix B may be defined by

$$B = A Q_A \tag{5.26}$$

such that

$$\begin{aligned} v(Bz) &= v_A(0) + (\nabla v_A^0) \cdot Q_A z + \tfrac{1}{2}(Q_A z)^T H_A^0 Q_A z + \dots \\ &= v_A(0) + (Q_A^T \nabla v_A^0) \cdot z + \tfrac{1}{2} z^T D_A z + \dots \end{aligned} \tag{5.27}$$

When the random walk is generated with Bz, the effect is to diagonalize the quadratic term. The largest eigenvalue is around 0.893; however, they drop off rapidly, with the 14^{th} being less that 0.0001. Only the first 24 are greater than 0.00001. Thus the effective dimension of the problem is reduced, and the decoupling of dimensions allows for the superior performance of quasi-random sequences on one dimensional problems to be exploited. Again we remark that no advantage arises if this technique is applied in conjunction with random sequences as this decoupling does not alter the total variance of the integrand.

There is a substantial additional computational cost of generating paths with Bz when B is a full matrix such that no recursive generation is possible. It may be more effective to choose a permutation based Brownian bridge discretization matrix for which the associated change of variables $Q = A^{-1}\,B$ leads to a significant (although not complete) diagonalization of the Hessian. Experimentation with various permutations indicates that almost any choice for such a B will be superior in this regard to using the standard random walk generating matrix A.

For the price of the matrix vector multiple which, as just described, could diagonalize the quadratic term, this term may in fact be eliminated by using the Taylor series as a control variate. A control variate is a function $\phi(z)$ which approximates $v(z)$ such that $E(\phi(z))$ is known exactly. The expectation of $v(z)$ is computed according to

$$E(v(z)) = E(v(z) - \phi(z)) + E(\phi(z))\,. \tag{5.28}$$

If $\phi(z)$ is close to $v(z)$, then the variance of the function $v(z) - \phi(z)$ will be small, leading to a more accurate evaluation of $E(v(z))$ through (5.28). A good choice for $\phi(z)$ combines the constant and quadratic terms of the Taylor expansion of $v(z)$ (5.19) (assuming antithetic variates will be used to eliminate the odd terms). It is most natural to choose the expansion which corresponds to the Brownian bridge discretization used to generate the paths, and thus it is more convenient to consider the approximation as a function of the path x (the deviation from the mean path) so that

$$\phi(x) = v_A^0 + \frac{1}{2} x^T \left((A^{-1})^T\, H_A\, A^{-1} \right) x\,. \tag{5.29}$$

The function (5.29) will be correct for all $x = \widehat{\sigma} B z$ such that B satisfies (5.15); thus the Brownian bridge may be used in conjunction with this control variate. Although the matrix in (5.29) is strongly tridiagonal, it is necessary to retain the off diagonal terms to ensure that $E(\phi(x))$ is close to the true solution. A disadvantage of this choice for ϕ is therefore that the evaluate of the quadratic term requires $\mathcal{O}(D^2)$ operations. However, the advantage is that errors should scale like $\widehat{\sigma}^4$.

6 Computational Results

We now describe the accuracy of various integration methods for computing the present value of the bond. The results are presented as a function of N, the number of paths, and as a function of the approximate computational time required for the methods. For each case, we present the root-mean-square of the error over 25 independent computations. Moreover, the computations for different values of N are all independent. For the Sobol' sequence calculations,

independence means that non-overlapping subsequences were used. Here the error is computed with respect to the exact solution.

The present value of the bond was computed with ten methods. The first was straight forward Monte Carlo using a random sequence (MC). Next, this calculation was repeated using a 360 dimensional Sobol' sequence (QMC) (generated with part of the code FINDER obtained from Columbia University). Both of these computations were then performed using antithetic variates (MC-anti and QMC-anti). The Sobol' sequence was then used with the Brownian bridge discretization and antithetic variates to generate the interest rate paths (QMC-BB-anti). Next the one-dimensional Van der Corput sequence was used in the reordering method (REO), and then used with antithetic variates (REO-anti). Reordering was also used with the Brownian bridge representation and antithetic variates (REO-BB-anti). Finally, the Taylor series control variate approach was used first with random samples (MC-Tay), and then with the Sobol' sequence combined with the Brownian bridge (QMC-BB-Tay).

For all but the reordering calculation, the number of sample paths ranged from $N = 256$ to $N = 131072$, with N being chosen as powers of two. For the reordering calculation, in order to avoid possible correlations within the base two sequence, N was chosen for the first two calculations as powers of three, ranging from $N = 243$ to $N = 59049$. For the REO-BB-anti calculation, primes close to powers of two were used. In all cases, the rms error of the 25 runs was computed at each N, and a line was fit to the log-log data to estimate the convergence rate. This assumes that over this range of N, the error may be modeled as $cN^{-\alpha}$. For random Monte Carlo, the constant c is the standard deviation, and $\alpha = .5$. Table 1 summarizes the results. For each method, the estimated size of the error at $N = 10000$ (based on the linear fit), the convergence rate α, and the approximate computation time for one run with this N are given. All the computation times grow linear with N except for the reordering calculations, which grow as $N \log N$ due to the sort. The results are plotted in Figures 1 and 2. Figure 1 shows a log-log plot (base 10) of the relative error as a function of N. Figure 2 shows the same data, but now plotted as a function of time, based on the estimates of computation time per sample.

The results for QMC-anti have not been plotted because as a function of N, the errors in this case are essentially the same as for simple QMC. This indicates that there is no advantage in using antithetic variates with the simple quasi-random sequence for this problem. However, the errors for the MC-anti calculation are significantly smaller than those of QMC and QMC-anti. This result may be interpreted in terms of the Taylor series expansion. The use of antithetic variates will eliminate the error from the linear term, but has no effect on the quadratic term. The fact that QMC and QMC-anti give the same error indicates that when quasi-random sequence is used, the errors from the quadratic term dominate the linear term (which may be converging like $\mathcal{O}(1/N)$). This is not true for the random case as there is a substantial error reduction from MC to MC-anti. The fact that MC-anti has lower error than

Method	Convergence Rate	Relative Error N = 10000	Comp. Time (sec) N = 10000
MC	0.508	6.05e-4	34
QMC	0.632	1.05e-4	22
MC-anti	0.518	4.42e-5	40
QMC-anti	0.591	9.90e-5	28
QMC-BB-anti	0.779	7.33e-6	32
MC-TAY	0.508	4.44e-7	496
QMC-BB-TAY	0.650	1.38e-7	517
REO	0.530	3.04e-5	46
REO-anti	0.491	3.00e-5	61
REO-BB-anti	0.461	4.01e-5	63

Table 1: Comparison of Monte Carlo and quasi-Monte Carlo methods used to value a coupon bond

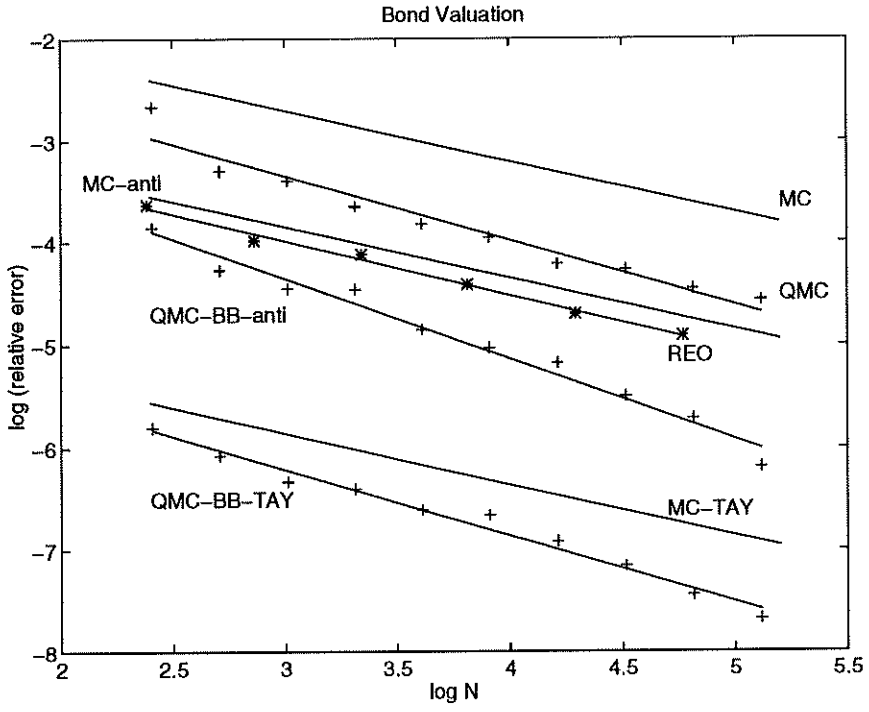

Figure 1: Error vs. N (log base 10)

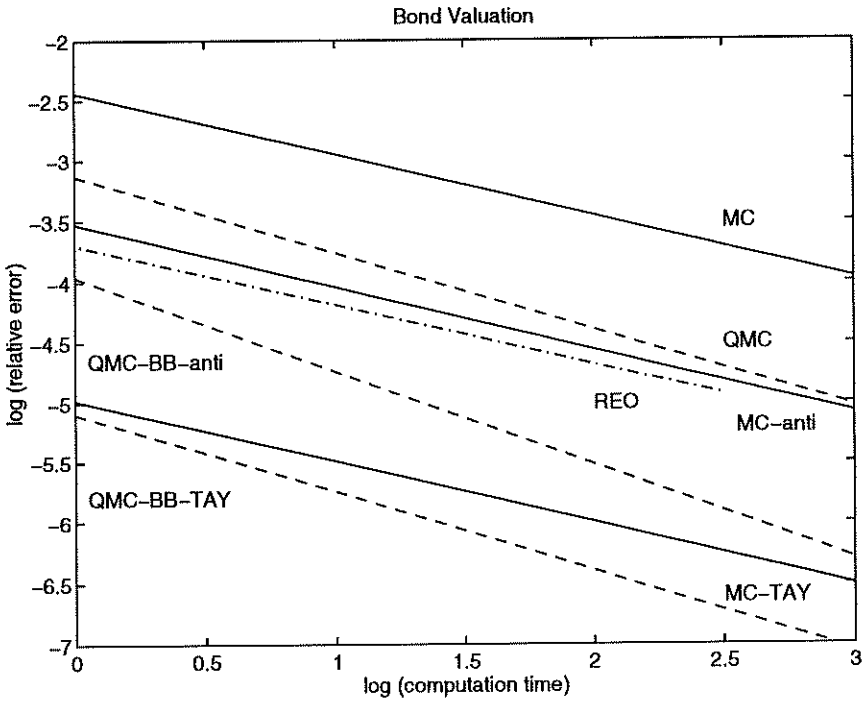

Figure 2: Error vs. Computation Time (log base 10)

QMC (with and without the antithetic variates) indicates that quasi-random sequence introduces larger errors in computing the quadratic term than the random sequence. The reason for this lies in the high dimensional nature of the problem when the standard discretization of the random walk is used. The quadratic term is the sum over all possible two dimensional projections of the 360 dimensional sequence. As has been previously demonstrated [13, 2], the Sobol' sequence has good one dimensional projections in all dimensions; however, some of the two (and higher) dimensional projections can suffer from clumping of points together. The corresponding holes are eventually filled in for large enough N, but this may occur on a scale much larger than the number of points of practical interest. As the dimension of the sequence increases, length of these clumping cycles may increase.

Of the reordering methods, only the simple REO is plotted. The combination of reordering with antithetic variates and the Brownian bridge discretization did not result in any improvement. In fact, the results for REO-BB-anti were slightly worse. This is attributable to the use of a different set of N (powers of 3 for REO and primes close to powers of 2 for REO-BB-anti), which illustrates the sensitivity of reordering to the choice of N. When powers of 2 are used, substantially greater errors appear. For bond valuation problem, reordering shows random-like convergence $1/\sqrt{N}$, and gives errors which are slightly smaller than MC-anti. This indicates that reordering is handling the univariate linear terms well (leading to errors smaller than the dominant quadratic term error), and handling the quadratic term only slightly better than the random sequence. This suggests that the two dimensional projections for the reordering process have essentially random properties, while the diagonal, univariate quadratic terms are integrated more accurately. The dominance of the quadratic error explains why the use of antithetic variates does not improve the error. The fact that the Brownian bridge discretization also does not help stems from the dominance of the random like errors associated with the two dimensional projections. These are still important because this discretization only leads to a partial diagonlization of the quadratic term.

A major improvement over random Monte Carlo is achieved when the quasi-random sequence is used in the Brownian bridge discretization with antithetic variates. In terms of the Taylor series, the number of quadratic cross terms which significantly contribute to the error has been greatly reduced by the partial diagonlization, and those cross terms which are important occur in relatively low dimensions. For low dimensions, the two dimensional projections of the Sobol' sequence are very uniformly distributed. This results in substantially lower integration errors and faster convergence than with antithetic random Monte Carlo.

The validity of the Taylor series analysis of this problem is borne out by the considerable error reduction achieved when the first terms are used as a control variate. In the random case, this leads to a variance of 0.000039, compared with 80.3 for MC and 0.415 for MC-anti. Even greater gains are made by using

the quasi-random sequence in the Brownian bridge discretization. Again a kind of partial diagonalization of the fourth order term is achieved and the greater uniformity of the early four dimensional projections is exploited.

Figure 2 displays these results in terms of the estimated computation time. In Table 1 it can be seen that there is in fact a computational advantage to using quasi-random sequences over random for this problem. This is due to the time required for sequence generation. The Sobol' sequence, which may be generated using bit-wise operations [22], is faster to generate than the 48 bit random generator used (drand48). The use of the Brownian bridge discretization comes at a relatively low computational price, and in fact, QMC-BB-anti was faster than simple MC. It should be noted that for more complicated integrands, the importance of sequence generation time will fade and the cost of antithetic variates may increase. A much larger computational effort was required to use the Taylor series control variate, due to the matrix vector multiply associated with the Hessian term. However, the results in Figure 2 still show that the Taylor series control variate leads to the lowest error, even when time is taken into account. This is related to the fact that for random Monte Carlo, it is necessary to compute 100 times longer to reduce the error by a factor of 10.

7 Conclusions

In this paper two general procedures have been developed for generating discrete sample paths associated with a stochastic process using quasi-random sequences. The properties of these methods have been illustrated by applying them to an example from finance, the valuation of a coupon bond with a mean reverting stochastic interest rate.

The main conclusions are that even for nominally high dimensional integration problems arising from discrete stochastic processes, quasi-Monte Carlo methods can be extremely effective. The best method and the degree of success depends, of course, on the nature of the integrand.

For functions which may be decomposed into the sum of dominant univariate functions, straightforward application of quasi-random sequences should result in improved accuracy over random Monte Carlo. For these cases, however, the use of the particle reordering method allows this to be done using the one dimensional Van der Corput sequence, which avoid the need to generate very high dimensional sequences. Moreover, particle reordering appears to avoid the problem of poor two dimensional projections associated with high dimensional sequences.

For functions with strong multi-variate components, the Brownian bridge discretization of the random walk can lead to a substantial reduction in the effective dimension of the integrand. If special knowledge of the integrand is available, it may be possible to construct a generalized Brownian bridge repre-

sentation tailored to the integrand which results in optimal dimension reduction.

Finally it has been shown that in some circumstances that quasi-Monte Carlo methods may be effectively combined with standard variance reduction techniques such as antithetic sampling and control variates. In particular for smooth functions of a random walk, the use of a Taylor series expansion about the expected mean path as a control variate has been shown to offer superior results when combined with quasi-random paths generated in a Brownian bridge discretization.

Acknowledgments

The author wishes to thank Spassimir Paskov and Joseph Traub for kindly providing the use of the quasi-random sequence generator FINDER and for a number of helpful discussions. Many thanks are also due to Russel Caflisch and Art Owen.

References

[1] R.E. Caflisch and W.J. Morokoff. Quasi-Monte Carlo computation of a finance problem. In K.T. Fang and F.J. Hickernell, editors, *Workshop on Quasi-Monte Carlo Methods and Their Applications*, pages 15–30 and UCLA CAM Report 96–16. 1996.

[2] R.E. Caflisch, W.J. Morokoff, and A.B. Owen. Valuation of mortgage backed securities using brownian bridges to reduce effective dimension. *UCLA CAM Report 97-01*, 1997.

[3] F. J. Fabozzi. *Bond Markets, Analysis and Strategies.* Prentice-Hall, Inc., New Jersey, 1996.

[4] J. Hull. *Options, Futures and Other Derivative Securities.* Prentice-Hall, Inc., New Jersey, 1993.

[5] A. Keller. A quasi-Monte Carlo algorithm for the global illumination problem in a radiosity setting. In H.Niederreiter and P.J.S. Shiue, editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, pages 239–251. Springer Verlag, New York, 1995.

[6] A. Kersch, W.J. Morokoff, and A. Schuster. Radiative heat transfer with quasi-Monte Carlo methods. *Transport Theory and Statistical Physics*, 23(7), 1994.

[7] P.E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations.* Springer-Verlag, Berlin; New York, 1992.

[8] L. Kuipers and H. Niederreiter. *Uniform Distribution of Sequences.* Wiley, New York, 1974.

[9] C. Lecot. Low disecrepancy sequences for the Boltzmann equation. *J. Comput. Appl. Math.*, 25:237–249, 1989.

[10] C. Lecot. A quasi-Monte Carlo method for the Boltzmann equation. *Mathematics of Computation*, 56(194):621–644, April 1991.

[11] C. Lecot. A quasi-monte carlo scheme using nets for a linear boltzmann equation. *SIAM J. Numerical Analysis*, page to appear, 1997.

[12] W. Morokoff and R.E. Caflisch. A Quasi-Monte Carlo approach to particle simulation of the heat equation. *SIAM Journal on Numerical Analysis*, 30:1558–1573, 1993.

[13] W. Morokoff and R.E. Caflisch. Quasi-random sequences and their discrepancies. *SIAM J. Sci. Stat. Computing*, 15:1251–1279, 1994.

[14] W. Morokoff and R.E. Caflisch. Quasi-Monte Carlo integration. *J. Comp. Phys.*, 122:218–230, 1995.

[15] B. Moskowitz and R.E. Caflisch. Smoothness and dimension reduction in quasi-Monte Carlo methods. *J. Math. Comp. Modeling*, 23:37–54, 1996.

[16] Harald Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods.* S.I.A.M., Philadelphia, PA, 1992.

[17] S. Ninomiya and S. Tezuka. Toward real-time pricing of complex financial derivatives. *Appl. Math. Finance*, 3:1–20, 1996.

[18] D.M. O'Brien. Accelerated quasi Monte Carlo integration of the radiative transfer equation. *J. Quant. Spectrosc. Radiat. Transfer*, 48(1):41–59, 1992.

[19] A. B. Owen. Orthogonal arrays for computer experiments, integration and visualization. *Statistica Sinica*, 2:439–452, 1992.

[20] S.H. Paskov. New methodolgies for valuing derivatives. In S. Pliska and M. Dempster, editors, *Mathematics of Derivative Securities.* Isaac Newton Inst., Cambridge U. Press, 1996.

[21] S.H. Paskov and J.F. Traub. Faster valuation of financial derivatives. *J. Portfolio Manag.*, pages 113–120, 1995.

[22] W.H. Press and S.A. Teukolsky. Quasi- (that is, sub-) random numbers. *Computers in Physics*, pages 76–79, Nov/Dec 1988.

[23] B.V. Shuhman. Application of quasirandom points for simulation of gamma radiation transfer. *Progress in Nuclear Energy*, 24:89–95, 1990.

[24] I.M. Sobol′. On the distribution of points in a cube and the approximate evaluation of integrals. *Zh. vŷchisl. Mat. mat. Fiz.*, 7(4):784–802, 1967.

[25] O. A. Vasicek. An equilibrium characterization of the term structure. *Journal of Financial Economics*, 5:177–88, 1977.