Path Generation for Quasi-Monte Carlo Simulation of Mortgage Backed Securities

Fredrik Åkesson[*] John P. Lehoczky[†]

First version: December 1998
This version: September 29, 2000

This is an extended version of a paper that appeared
in the September 2000 issue of Management Science.

Abstract

Monte Carlo simulation is playing an increasingly important role in the pricing and hedging of complex, path dependent financial instruments. Low discrepancy simulation methods offer the potential to provide faster rates of convergence than those of standard Monte Carlo methods, however in high dimensional problems special methods are required to ensure that the faster convergence rates hold. Indeed, Ninomiya and Tezuka (1996) have shown high-dimensional examples, in which low discrepancy methods perform worse than Monte Carlo methods. The principal component construction introduced by Acworth et al. (1998) provides one solution to this problem. However, the computational effort required to generate each path grows quadratically with the dimension of the problem. This article presents two new methods that offer accuracy equivalent, in terms of explained variability, to the principal components construction with computational requirements that are linearly related to the problem dimension. One method is to take into account knowledge about the payoff function, which makes it more flexible than the Brownian Bridge construction. Numerical results are presented that show the benefits of such adjustments. The different methods are compared for the case of pricing mortgage backed securities using the Hull-White term structure model.

1 Introduction

Monte Carlo simulation methodology has become an important tool in the pricing and hedging of complex financial instruments. This is because one can often express the price of a contingent claim as the discounted expected value of its payoff with respect to a martingale measure, a probability measure with respect to which the discounted asset price process is a martingale[1]. Hedging portfolios can often be constructed from the derivatives of the instrument's price as a function of some parameter. To estimate a price using simulation one needs only to generate a set of independent random price paths, compute the payoff for each of them, average the results, then discount to present value. The law of large numbers guarantees the consistency of such an estimate, and the central limit theorem permits the construction of confidence intervals. If n represents the number of independent paths used to construct the estimate, then the rate of convergence is $O(n^{-1/2})$. Variance reduction methods such as importance sampling, control variates or antithetic variates can substantially reduce the constant term but do not change the $O(n^{-1/2})$ rate of convergence. This rate is often too slow for the industry practice on complex pricing and hedging problems, and a method producing a faster rate of convergence must be found.

[*] Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213 and Department of Mathematics, Royal Institute of Technology, 100 44 Stockholm, Sweden. fakesson@andrew.cmu.edu

[†] Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213. jpl@stat.cmu.edu

[1] The reader might consult standard reference on derivative security pricing such as Hull (1997) or the more advanced Duffie (1996) to background on arbitrage-based pricing theory.

One can also compute the discounted expected value representing a derivative security price as an integral involving the payoff function and the joint density function of the asset prices. For problems involving only a low dimensional integral direct numerical integration methods such as using a trapezoid rule with a grid of evaluation points can offer a much faster convergence rate than $O(n^{-1/2})$. However, as the dimension increases, these methods become impractical. A promising alternative approach is to carry out a simulation replacing the independent paths with paths generated from low discrepancy sequences. This approach can provide consistent price estimates with a much faster convergence rate than $O(n^{-1/2})$.

Quasi-Monte Carlo (also known as low discrepancy) methods have been widely studied in numerical analysis (see Niederreiter (1992) for a comprehensive introduction), but their use in computational finance problems was popularized by Paskov and Traub (1995) and Paskov (1996). A survey of the use of Monte Carlo and low-discrepancy simulation methods in finance was given by Boyle, Broadie and Glasserman (1997). Low discrepancy methods exhibit excellent results for low dimensional problems; however, recent interest has focused on pricing mortgage-backed securities, a particularly challenging problem. These securities, which are discussed in detail later in this paper, involve monthly cashflows over a 30 year period (resulting in a 360-dimensional problem) with the added complexity of mortgage prepayments which depend on earlier interest rates. As mentioned earlier, derivative security prices can often be computed as an expected value. This expectation is a d-dimensional integral with respect to the probability density derived from the discretized version of the asset price process. Using the probability integral transformation, one can transform this integral into an integral with respect to a uniform distribution over the d-dimensional unit cube where the continuous price process has been discretized into a d-dimensional vector (see for example Joy, Boyle and Tan 1996). Consequently, pricing contingent claims becomes equivalent to integrating functions over the d-dimensional unit cube. These are problems for which Monte Carlo and quasi-Monte Carlo methods can be quite attractive if d is large. Quasi-Monte Carlo methods involve the generation of a sequence of d-dimensional vectors, treating them in the same way as one would treat a d-dimensional pseudo-random sequence in standard Monte Carlo methods. The difference is that quasi-Monte Carlo methods are designed to fill the d-dimensional space much more uniformly than an ordinary random sample. An important difficulty arises when low discrepancy sequences are used in high dimensional problems such as the 360-dimensional mortgage-backed security problem described above. The difficulty is that low-discrepancy sequences are no more uniform than random sequences in high dimension, unless one uses a very large number of points. It is hard to assess the uniformity of a sequence in a high-dimensional space. A necessary but not sufficient condition for uniformity is uniformity of low-dimensional coordinate projections of the sequence. For example, consider Figure 1 in which the first and the third coordinate of 1024 points taken from a Sobol sequence are plotted. This very regular figure should be compared with Figure 2 which plots pairs of points from a pseudo-random generator. One can see the relative regularity of the Sobol points compared with the pseudo-random points which exhibit both clustering and relative sparsity over the unit square. However, in higher dimensions, lower dimensional projections of the Sobol sequences can exhibit very poor uniform regularity. Consider Figure 3 which gives the projections of coordinates 30 and 31 for 1024 Sobol numbers. The picture is quite regular, but is far from the desired uniformity.

Ninomiya and Tezuka (1996) report that some low-discrepancy sequences can not only lose their advantage but can even perform worse than Monte Carlo methods in high dimensional financial problems. Recently, important works by Acworth et al. (1998), Morokoff (1998) and Caflisch et al. (1997) began to develop effective methods to achieve accurate price with convergence rates much better than $O(n^{-1/2})$. These methods introduce new approaches to construct price paths so that their shapes are mostly determined by the first few dimensions of a low-discrepancy sequence. This paper introduces two new methods, the partial principal components and the subsequence methods, which are computationally efficient and obtain improvements in the rates of convergence. The current and the new methods are discussed in detail in later sections of the paper.

This paper is organized as follows. In Section 2 we present some background on various price path generation methods and introduce our new methods. In Section 3 we introduce the one-factor Hull-White term structure model under which we will do our pricing. This model is particularly convenient as it can be written in terms of an Ornstein-Uhlenbeck process which is Gaussian and Markov. This allows us to

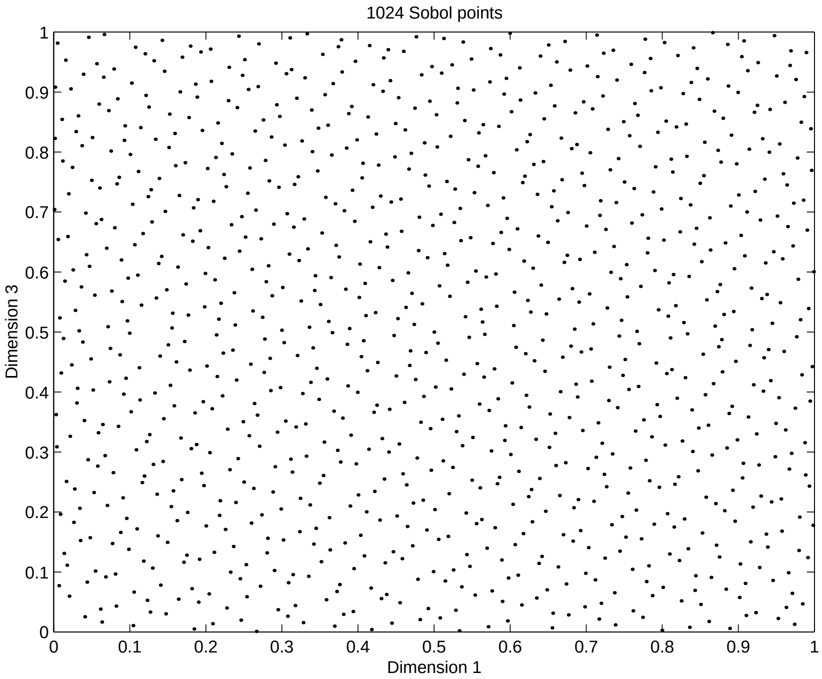


Figure 1: Two-dimensional projection of a Sobol sequence.

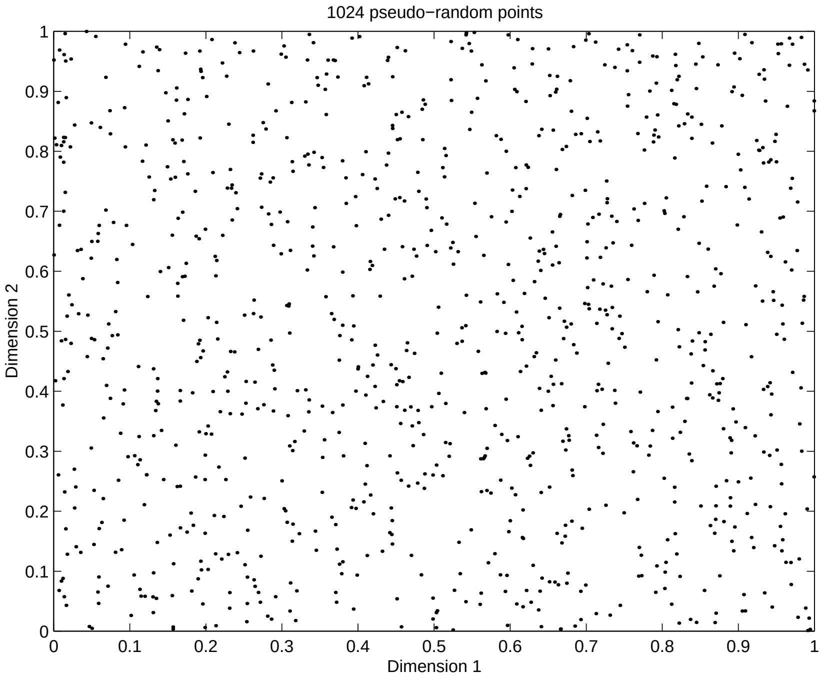


Figure 2: Two-dimensional projection of a pseudo-random sequence.

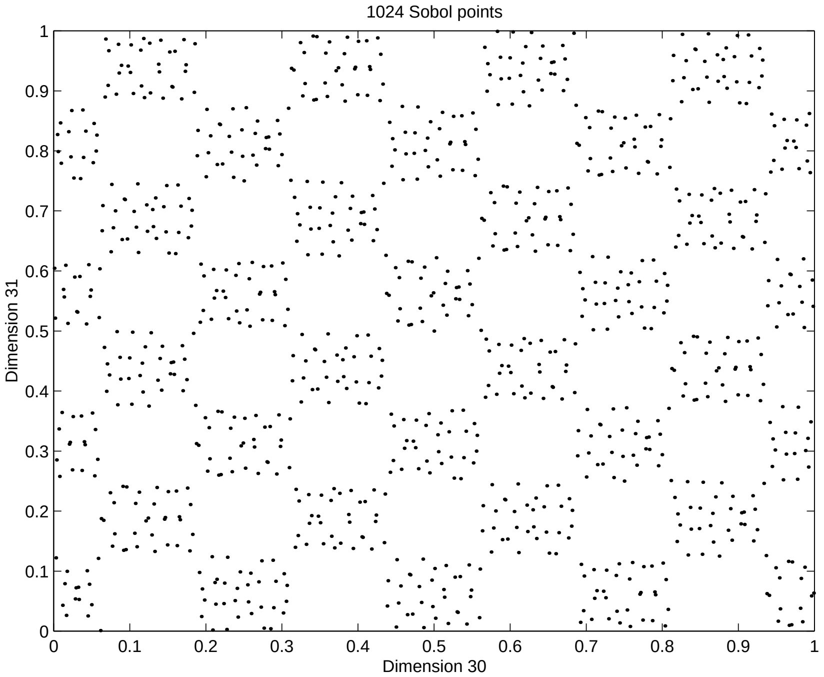


Figure 3: Two-dimensional projection of a Sobol sequence.

simulate prices and discount factor exactly, without any discretization error. The extension to other term structure models is also discussed. Section 4 presents five different methods of path generation including the two new methods: *partial principal components* and *subsequence principal components*. Section 5.3 applies the methods developed in Section 4 to price mortgage-backed securities with the Hull-White term structure model and a commonly used prepayment model by Richard and Roll (1989).

2 Background on Path Generation Methods

The worst case error bounds that exists for quasi-Monte Carlo methods are $O(n^{-1}(\log n)^d)$, asymptotically superior to $O(n^{-1/2})$. These bounds do not explain the success of low-discrepancy methods in high-dimensional problems in finance since the bounds are increasing for $n < e^d$. The concept of **effective dimension** introduced by Caflisch et al. (1997) offers one explanation. An integrand is said to have a low effective dimension if it can be written as a sum of lower dimensional integrands plus an integrand with small variance. An integrand f has effective dimension $\leq t$ in the **truncated sense** if there exists a function g depending only on the first t coordinates such that $\mathrm{Var}(f-g)$ is much smaller than $\mathrm{Var}(f)$, or equivalently

$$\mathrm{E}\left(f - \mathrm{E}[f|z_1, \ldots, z_k]\right)^2 \tag{1}$$

is much smaller than $\mathrm{Var}(f)$. An integrand f has effective dimension $\leq s$ in the **superposition sense** if there exists s-dimensional functions $\{g_i\}$ such that $\mathrm{Var}(f - \sum_i g_i)$ is much smaller than $\mathrm{Var}(f)$. The idea is that the lower dimensional integrands will have a convergence rate faster than $O(n^{-1/2})$ and the high dimensional integrand with small variance will have a negligible effect for sample sizes used in practice (see Owen (1998) for a precise discussion).

We will assume that our integrand is a function of a normally distributed vector x with mean zero and covariance matrix Σ. The vector x can be generated by taking a linear transformation of a vector z

i.i.d. standard normals, i.e. $x = Az$ for some square matrix A which creates the appropriate covariance matrix Σ. Thus we require $AA' = \Sigma$. The goal of the methods in this article is to use the freedom in the choice of A to reduce the effective dimension in the truncated sense of the integrand.

Caflisch and Moskowitz (1995) departed from the standard "forward" method of constructing paths of Brownian motion, in which points are generated in increasing temporal order. Instead, they introduced the Brownian bridge construction method in which one generates the end points first then fills in the midpoint, quarter-points and so forth of the Brownian motion, in each case sampling the new point from its conditional distribution given the existing points. By using the ith coordinate of a quasi-random vector z to construct the ith step, the early parts of z determine the major features of the path. Caflisch et al. (1997) use the Brownian bridge construction to price mortgage backed securities.

Many integrands f that appear in derivative pricing have the feature that two paths that look similar also generally give similar values. This can be expressed as

$$|f(x) - f(y)|^2 \leq C\|x - y\|_2^2, \tag{2}$$

where C is a positive constant. Finding an A that minimizes (1) is a very hard problem. We will instead minimize an upper bound to (1):

$$\begin{aligned} \mathrm{E}\left(f(x) - \mathrm{E}\big[f(x)|z_1, \ldots, z_k\big]\right)^2 &\leq \mathrm{E}\left(f(x) - f\big(A(z_1, \ldots, z_k, 0, \ldots, 0)'\big)\right)^2 \\ &\leq C\ \mathrm{E}\,\|x - A(z_1, \ldots, z_k, 0, \ldots, 0)'\|_2^2 \\ &= C\ \mathrm{tr}\left[\mathrm{Cov}\big(x - A(z_1, \ldots, z_k, 0, \ldots, 0)'\big)\right]. \end{aligned} \tag{3}$$

We recall that $g(X) = \mathrm{E}[Y|X]$ minimizes $\mathrm{E}(Y - g(X))^2$, where X and Y are random variables. This gives the first inequality, which should be reasonably tight when the last $d - k$ columns of A are small. It is well known from the statistics literature (see Seber 1984) that (3) is minimized for all k when the columns of A are the principal components of Σ, i.e.

$$A = (\sqrt{\lambda_1}v_1, \ldots, \sqrt{\lambda_d}v_d),$$

where $\lambda_1 \geq \ldots \geq \lambda_d$ are the eigenvalues in decreasing order, and $v_1, \ldots, v_d$ are the corresponding unit-length column eigenvectors of Σ. Okamoto and Kanazawa (1968) show that this choice of A gives the optimal approximation of x by $A(z_1, \ldots, z_k, 0, \ldots, 0)'$ in a rather general sense, which includes the case when the trace in (3) is replaced by the Frobenius norm. It is possible to show that

$$\mathrm{E}\,\|x - A(z_1, \ldots, z_k, 0, \ldots, 0)'\|_2^2 = \mathrm{tr}(\Sigma) - \sum_{i=1}^{d}\sum_{j=1}^{k} {a_{ij}}^2, \tag{4}$$

where $\{a_{ij}\}$ are the elements of A. This motivates the next definition. The **variability explained** by the first k normals alone is defined as the sum of the squared norms of the first k columns of A. The principal component path generation was first introduced by Acworth, Broadie and Glasserman (1998) with motivation that it maximizes the explained variability. The connection to effective dimension given here is new and it will be discussed further in Section 4.6.

It is important to note that the principal components approach offers the possibility of generalization to account for the effect of the payoff function through the use of a weighted norm in (2). For example, with mortgage-backed securities, experience shows that most parts will be paid off in a relatively short number of years, thus the most important parts of the path are the first few years.

In spite of these optimality properties, the computational cost to carry out the principal components construction can be large, as it grows with the square of the dimension of the problem. This cost is of great concern in high dimensions. For example, mortgage-backed securities may require a 360-dimensional construction. On the other hand, the forward construction or the Brownian bridge construction are linear in the dimension. Hence, it would seem at first glance that the principal components method should be dismissed. However, in this paper we develop two methods based on the principal component construction which offer accuracy comparable (in terms of explained variability) to the full principal components construction, but have computational costs which are linear in the dimension.

The idea underlying the new method is very straightforward. In statistics, principal component methodology is often used to identify a coordinate representation in which a small number of coordinates (factors) can explain most of the variance in the data. We later show in Table 1 (see also Table 3 in Acworth et al. 1998) that the first 5 principal components explain 96% of the variance for a particular 256 dimensional problem. Consequently, only a small number of principal components are needed. If one can find an efficient method to generate a random variate that has the distribution of the residual components, then the computation can be reduced to just using a small number of components. This construction, called the *partial principal components (PPC) method*, is introduced in Section 4.4.

A second construction method adopts a hierarchical approach. First, a subset of the path is generated on a regular subsequence of size k of the evaluation points. For example, for a mortgage-backed security over a 30 year time interval one might generate asset prices for the first month of each year, $W_{12}, W_{24}, \ldots, W_{360}$. The remaining points can be filled in using Brownian bridge techniques. This approach is far more efficient than the full principal components approach but offers comparable accuracy. This approach, which we call the *subsequence principal components* method, is discussed in Section 4.5.

3 One-Factor Hull-White Model

The Hull-White (1990) term structure model[2] is a one-factor mean-reverting term structure model in which the spot interest rate under the martingale measure is given by the stochastic differential equation

$$dr(t) = (\theta(t) - ar(t))dt + \sigma dB(t), \tag{5}$$

where $\{B(t), t \geq 0\}$ is a Brownian motion, a and σ are constants, $\theta(t)$ is chosen to fit the initial term structure,

$$\theta(t) = \frac{\partial f(0,t)}{\partial t} + af(0,t) + \frac{\sigma^2}{2a}\left(1 - e^{-2at}\right),$$

where $f(0,t)$ is the instantaneous forward rate. In the special case with $\theta(t) = ab$ for some constant b, this model becomes the Vasicek (1977) model. It is convenient to introduce the reparameterization

$$dx(t) = -ax(t)dt + \sigma dB(t), \text{ with } x(0) = 0, \tag{6}$$

where

$$\alpha(t) = r(t) - x(t) = f(0,t) + \frac{\sigma^2}{2a^2}(1 - e^{-at})^2. \tag{7}$$

Hull and White (1994) develop tree implementations using (6). The process defined by (6) is called the Ornstein-Uhlenbeck process, and its solution is given by

$$x(t) = \sigma e^{-at} \int_0^t e^{au} dB(u), \tag{8}$$

a Gaussian Markov process, which can also be represented as

$$x(t) = \sigma e^{-at} W\left(\frac{e^{2at} - 1}{2a}\right), \tag{9}$$

where $\{W(t), t \geq 0\}$ is a new Brownian motion, related to $B(t)$ in a non-trivial way (see Karatzas and Shreve 1991). The representation given by (9) will be used extensively throughout this paper.

[2] The reader can consult Chapter 17 of the text by Hull (1997) for an introduction to the term structure models discussed in this section and bond pricing formulas.

The representation given by (6) is also useful in representing discount factors. According to arbitrage pricing theory, the price of a security is equal to the expected value of its discounted cash flows. The discounting factor is equal to $e^{-\int_0^t r(u)du}$ and

$$\int_0^t r(u)du = \int_0^t \alpha(u)du + \int_0^t x(u)du,$$

with

$$\int_0^t \alpha(u)du = -\ln P(0,t) + \frac{\sigma^2}{4a^3}\left(2at - 3 + 4e^{-at} - e^{-2at}\right),$$

where $P(0,t)$ denotes the current price of bonds maturing at time t. The trapezoid rule can be used to estimate $\int_0^t x(u)du$ from $\{x(t_i)\}$.

The path constructions that we will introduce later are not unique for the Hull-White model. We will derive the path constructions for all interest rate models for which $r(t)$ can be represented in the following way:

$$r(t) = F(\alpha(t) + g(t)W_{h(t)}) \tag{10}$$

where $\alpha, g : \mathbf{R}_+ \to \mathbf{R}$ are continuous functions, and the functions $F : \mathbf{R} \to \mathbf{R}$ and $h : \mathbf{R}_+ \to \mathbf{R}_+$ are strictly increasing and continuous. From (7) and (9) we see that the Hull-White model is represented as: $F(x) = x$, $\alpha(t) = f(0,t) + \frac{\sigma^2}{2a^2}(1-e^{-at})^2$, $g(t) = \sigma e^{-at}$ and $h(t) = \frac{e^{2at}-1}{2a}$. The representation for the Black-Karasinski (1991) model and other interest rate models are given in Schmidt (1997). To simulate $r(t)$ given by (10), we will first simulate

$$x(t) = g(t)W_{h(t)}, \tag{11}$$

which is a Gaussian Markov process, and then compute the interest rate by $r(t) = F(\alpha(t) + x(t))$.

It is not sufficient to know the path of $r(t)$ to evaluate the payoff function of an interest rate dependent contingent claim. One also needs to know the bond prices $P(r(t), t, T)$ for the dates at which the claim has cash flows. For some models, including the Hull-White model, it is fortunate that there exists a closed-form solution for the bond prices. However, for the Black-Karasinski model and other models without closed-form solutions we must use numerical methods to compute the bond prices. It would be very inefficient to use Monte Carlo simulations to compute these bond prices. Instead one should use a tree implementation (see Hull 1994 or Schmidt 1997) or a finite difference implementation. Since a tree or a finite difference method gives bond prices for only a discrete number of interest rates, some interpolation scheme is needed to obtain bond prices for other interest rates.

4 Path Generation Methods

In this section, we discuss different ways to generate paths of $x(t)$ without discretization errors at $t = t_1, \ldots, t_d$, where $0 = t_0 < \ldots < t_d = T$ and T is constant. We first derive formulas for $x(t)$ for the general form (11) and later specialize them to (9). In all cases we have $x(0) = 0$.

4.1 The Standard Path Generation

The standard (forward) path generation method is given by

$$\begin{aligned} x(t_{i+1}) &= \frac{g(t_{i+1})}{g(t_i)} g(t_i) W(h(t_i)) + g(t_{i+1})\left[W(h(t_{i+1})) - W(h(t_i))\right] \\ &= \frac{g(t_{i+1})}{g(t_i)} x(t_i) + g(t_{i+1})\sqrt{h(t_{i+1}) - h(t_i)}\, z_{i+1} \end{aligned} \tag{12}$$

Construction	Cumulative Explained Variability (%)					Mult.
Standard	0.0078	0.0155	0.0233	0.0309	0.0386	256
Brownian bridge	0.6680	0.8340	0.8755	0.9170	0.9274	512
Prin. comp.	0.8106	0.9006	0.9331	0.9496	0.9596	65536
PPC ($k = 8$)	0.8106	0.9006	0.9331	0.9496	0.9596	6656
Subseq. ($k = 16$)	0.8093	0.8987	0.9306	0.9465	0.9560	736
Subseq. ($k = 32$)	0.8103	0.9002	0.9324	0.9488	0.9587	1472

Table 1: Cumulative explained variability from the first five dimensions using the standard construction, the Brownian bridge construction, the principal component construction, the partial principal component construction with $k = 8$ and the subsequence construction with $k = 16, 32$, based on dimension $d = 256$, and the mean-reverting coefficient $a = 0$, i.e. for an ordinary Brownian motion. The number of multiplications for the methods are: d, $2d$, d^2, $d(3k+1)$ and $k^2 + 2(d-k)$ for the last two rows. Since d is a power of two and a is equal to zero, fewer multiplications are needed compared with the general case.

where $\{z_i\}$ are independent standard normals. In the special case where $x(t)$ is from the Hull-White model we obtain

$$x(t_{i+1}) = e^{-a\Delta t_i} x(t_i) + \sigma \sqrt{\frac{1 - e^{-2a\Delta t_i}}{2a}} z_{i+1} \qquad (13)$$

where $\Delta t_i = t_{i+1} - t_i$. The matrix A corresponding to this method is the Cholesky factorization of Σ. We note that $2d$ multiplications are needed to generate a path with d time-steps.

4.2 Principal Component Path Generation

The covariance matrix $\Sigma = (\sigma_{ij})$ needed to compute the principal components is given by

$$\sigma_{ij} = g(t_i) g(t_j) h(t_i \wedge t_j), \qquad (14)$$

where $t_i \wedge t_j$ denotes the minimum of t_i and t_j. In the special case that $x(t)$ is from Hull-White model we obtain

$$\sigma_{ij} = \sigma^2 \frac{e^{-a|t_i - t_j|} - e^{-a(t_i + t_j)}}{2a}. \qquad (15)$$

The computation time required to find the eigenvectors and eigenvalues of a covariance matrix of dimension d by standard numerical procedures is proportional to d^3. In the special case when $t_i = \frac{i}{n} T$, $i = 1, \ldots, d$, there exists an algorithm (see Åkesson and Lehoczky 1998) for computing the eigenvalues and eigenvectors of (σ_{ij}) in $O(d)$ operations, when σ_{ij} is given by (15). For these t_i's there exists a closed-form solution for the covariance matrix of the Brownian motion, i.e. when $a = 0$.

4.3 Brownian Bridge Path Generation

The Brownian bridge construction introduced by Caflisch and Moskowitz (1995) offers a method to generate paths that concentrate their variance on the first few components of the normals with computational cost equivalent to that of the standard path generation method. The method is based on the fact that the distribution of a Markov process $x(s)$ at $s = s_2$ conditional on $\{x(u), 0 \le u \le s_1,\ s_3 \le u\}$ is the same as the conditional distribution of $x(s_2)$ given $x(s_1), x(s_3)$. This means that we can generate paths in a recursive way, where we only need to take into account the closest known values of x.

Morokoff (1998) presents a rather complicated algorithm for generating the conditional distribution of an Ornstein-Uhlenbeck process. We will now derive a closed-form solution for $x(t)$ based on (11). Given

Construction	Cumulative Explained Variability (%)					Mult.
Standard	0.0033	0.0066	0.0100	0.0133	0.0166	720
Brownian bridge	0.1958	0.4754	0.5934	0.7113	0.7464	1080
Prin. comp.	0.4785	0.6763	0.7707	0.8238	0.8573	129600
PPC ($k = 8$)	0.4785	0.6763	0.7707	0.8238	0.8573	9360
Subseq. ($k = 15$)	0.4754	0.6701	0.7616	0.8121	0.8432	1260
Subseq. ($k = 30$)	0.4777	0.6747	0.7683	0.8207	0.8536	1890

Table 2: Cumulative explained variability from the first five dimensions using the standard construction, the Brownian bridge construction, and the principal component construction, the partial principal component construction with $k = 8$ and the subsequence construction with $k = 15, 30$, based on dimension $d = 360$, $T = 30$, and the mean-reverting coefficient $a = 0.1$. The number of multiplications for the methods are: $2d$, $3d$, d^2, $d(3k + 2)$ and $k^2 + 3(d - k)$ for the last two rows.

the values of $x(s)$ at s_1 and s_3, we have that

$$W(h(s_i)) = \frac{x(s_i)}{g(s_i)} \quad \text{for } i = 1, 3.$$

Standard results for the Brownian bridge give the conditional distribution of $W(t_2)$ given $W(t_1)$ and $W(t_3)$, for $t_1 < t_2 < t_3$, to be normally distributed with mean $\frac{(t_3-t_2)W(t_1)+(t_2-t_1)W(t_3)}{t_3-t_1}$ and variance $\frac{(t_3-t_2)(t_2-t_1)}{t_3-t_1}$. Consequently, the conditional distribution of $x(s_2) = g(s_2)W(h(s_2))$ is normally distributed with mean

$$g(s_2)\frac{(h(s_3) - h(s_2))\,\frac{x(s_1)}{g(s_1)} + (h(s_2) - h(s_1))\,\frac{x(s_3)}{g(s_3)}}{h(s_3) - h(s_1)}$$

and variance

$$g^2(s_2)\frac{(h(s_3) - h(s_2))\,(h(s_2) - h(s_1))}{h(s_3) - h(s_1)}.$$

In the special case when $x(t)$ comes from the Hull-White model with midpoints, i.e. $\Delta s_2 = \Delta s_3 = t/2$ where $\Delta s_i = s_{i+1} - s_i$, the mean simplifies to $\frac{x(s_1)+x(s_3)}{2\cosh(at/2)}$ and the variance to $\frac{\sigma^2 \tanh(at/2)}{2a}$. We note that as $a \to 0$, this converges to the distribution of the standard Brownian bridge.

The Brownian bridge construction begins by generating the path value at the final time point using the formula:

$$x(T) = g(T)\sqrt{h(T)}z_1.$$

In the Hull-White model, this formula becomes

$$x(T) = \sigma\sqrt{\frac{1 - e^{-2aT}}{2a}}z_1$$

The intermediate time points are filled in by subdividing the remaining time intervals into halved subintervals. For example, one constructs $x(T), x(T/2), x(T/4), x(3T/4), x(T/8)$, etc. If T is not a power of 2, one can choose the nearest smaller point, instead of the middle point. It is also possible to fill in the points in any order that is desired.

4.4 Partial Principal Component Path Generation

The usefulness of the principal component construction over the other path constructions is that it concentrates most of the path variability on the first few normals. It seems reasonable that we would obtain equivalent numerical accuracy with much less computational effort if we could first generate

$$\sum_{i=1}^{k} \sqrt{\lambda_i} v_i \, z_i$$

using a low discrepancy sequence that benefits from the variance concentration and then add one more random variable with the same distribution as

$$\sum_{i=k+1}^{d} \sqrt{\lambda_i} v_i \, z_i. \tag{16}$$

Consider the algorithms that generate multivariate normal vectors with covariance matrix Σ given a vector of d independent standard normals. There is no known algorithm that requires less than $O(d^2)$ multiplications for a general Σ. There exist more efficient algorithms for some covariance matrices. For example, all Gaussian Markov processes can be generated by $O(d)$ multiplications, and all stationary Gaussian processes can be generated by $O(d \log d)$ multiplications (see Wood and Chan 1994). In the presence of a more efficient algorithm it is possible to generate random variables with distribution (16) using less multiplications than $d(d-k)$ when d is large and k is small.

Theorem 1. *Let A and $B = (b_1, \ldots, b_d)$ be two matrices such that $AA' = BB' = \Sigma$ and let $\{z_i\}_{i=1}^{d+k}$ be independent standard normals, then*

$$\sum_{i=1}^{k} b_i z_i + A(I_d - UU') \begin{pmatrix} z_{k+1} \\ \ldots \\ z_{k+d} \end{pmatrix} \sim N(0, \Sigma), \tag{17}$$

where $U = A^{-1}(b_1, \ldots, b_k)$.

Proof. Let C be the $d \times k$ matrix containing the first k columns of $d \times d$ a unit matrix, then $U = A^{-1}BC$. We note that $U'U = C'B'(AA')^{-1}BC = C'B'(BB')^{-1}BC = C'B'(B')^{-1}B^{-1}BC = C'C = I_k$. It follows that $(I_d - U'U)(I_d - U'U)' = I_d - U'U$, hence the covariance matrix of $A(I_d - UU')(z_{k+1}, \ldots, z_{k+d})'$ is equal to $A(I_d - UU')(I_d - UU')A' = A(I_d - UU')A' = \Sigma - (AU)(AU)' = \Sigma - (b_1, \ldots, b_k)(b_1, \ldots, b_k)$. □

With $B = (\sqrt{\lambda_1} v_1, \ldots, \sqrt{\lambda_d} v_d)$ we can use the above theorem for the partial principal component construction. In the case that A corresponds to the standard path generation method, we can see from (17) that the partial principal component construction requires $3dk + 2d$ multiplications. We also have to generate $d + k$ normals instead of d. In the next section we will show that good accuracy can be obtained with relatively small values of k. This construction was inspired by a related construction by Glasserman et al. (1999) for the special case in which Σ is the identity matrix.

4.5 Subsequence Path Generation

The subsequence method is designed to generate paths that are similar to the paths generated by the principal component construction, but with a much smaller computational requirement. Given that one is going to generate a path with d time steps, the subsequence method selects a subset of k time points, $t_{i_1}, \ldots t_{i_k}$, where $0 < i_1 < \ldots < i_k = d$, and generates $x(t)$ at those time steps using the principal components construction. Once we are given the value of $x(t)$ at $t = t_{i_m}, 1 \leq m \leq k$, the remaining points are filled in using the Brownian bridge construction. The most natural choice for the subsequence would be to distribute the k points evenly, so if $d = 360$ and $k = 30$, we would use the subsequence $(x(t_{12}), x(t_{24}), x(t_{36}), \ldots, x(t_{360}))$. Table 1 and Table 2 show the explained variability in four different cases where the k initial points are evenly distributed. We notice that this construction gives almost the same explained variability as the principal component construction while using far fewer multiplications. Fishman et al. (1997) described a related path construction method, where one first generates the path at selected time points using a low discrepancy sequence and then fills in the remaining points. Although their paper lacks sufficient details to determine the precise method they propose, it appears that the standard path generation method was used to generate the initial subsequence of path values.

4.6 Weighted principal components

In Section 2 it was shown that using the lower dimensional components of the z vector on the parts of the path with the greatest variability minimizes an upper bound on the effective dimension in the truncated sense. This upper bound was based on the following inequality

$$|f(x) - f(y)|^2 \leq C \sum_{i=1}^{d} \left(x(t_i) - y(t_i)\right)^2. \tag{18}$$

One important limitation of this approach to path construction is that it treats all parts of x as being equally important. For many financial instruments certain parts of a path are more important than other parts. An extreme example is a European call or put option, where the payoff function depends only on the stock price at the last time step. For mortgage-backed securities, we expect that the earlier parts of $x(t)$ will be most important, because the value of one dollar in 30 years is worth much less than one dollar today, and mortgage prepayments depend on the previous interest rates. A generalization of (18), designed to weight the parts of a path according to their importance in the payoff, is

$$|f(x) - f(y)|^2 \leq \sum_{i=1}^{d} w_i^2 \left(x(t_i) - y(t_i)\right)^2, \tag{19}$$

where w_i are weights giving the relative importance of $x(t_i)$. By rescaling x it possible to use the previous results. Let $\tilde{x} = Dx$, where $D = \operatorname{diag}(w_1, \dots, w_d)$. Since $f(\tilde{x})$ satisfies (18), the optimal way of generating $\tilde{x}$ is given by the principal components of the covariance matrix of $\tilde{x}$. Given $\tilde{x}$ we scale it back to get x, so the optimal A is equal to

$$A_{WPC} = D^{-1}\left(\sqrt{\lambda_1}v_1, \dots, \sqrt{\lambda_d}v_d\right), \tag{20}$$

where $\lambda_1 \geq \ldots \geq \lambda_d$ are the eigenvalues in decreasing order, and $v_1, \dots, v_d$ are the corresponding unit-length column eigenvectors of $D\Sigma D'$. We will refer to this method as the *weighted principal component path construction.* Using $B = A_{WPC}$ in Theorem 1, we obtain the *weighted partial principal component construction.* If $x = Az$, we have that $\tilde{x} = DAz$ and (4) gives

$$\mathrm{E}\,\|\tilde{x} - DA(z_1, \dots, z_k, 0, \dots, 0)'\|_2^2 = \operatorname{tr}(D\Sigma D') - \sum_{i=1}^{d}\sum_{j=1}^{k} w_i {a_{ij}}^2. \tag{21}$$

The double sum in (21) is defined to be the *weighted explained variability* of the first k normals. Table 3 compares the cumulative weighted variability explained by the first seven normals. A further generalization of (18) would be to use

$$|f(x) - f(y)|^2 \leq (x-y)'C(x-y), \tag{22}$$

where C is a symmetric positive definite matrix. The same argument as above shows that the optimal A is given by (20) if D now denotes a matrix such that $D'D = C$. However, in most cases it would be difficult to specify $d(d+1)/2$ different weights, at least without some numerical procedure.

In order for (18), (19) or (22) to provide a useful guideline in lowering the effective dimension, they need to be reasonably tight for most pairs (x, y) that have a high probability of occurring. Examples when these bounds don't work include integrands that depend on the local rather than the global properties of the underlying price path. One such example, mentioned by Dupire and Savine (1998), is the volatility swap. A slightly modified version of it is

$$f(x) = \frac{1}{d}\sqrt{\sum_{i=0}^{d-1}\left(e^{x(t_{i+1}) - x(t_i)} - 1\right)^2}.$$

Construction	Cumulative Explained Weighted Variability (%)						
STD	0.0150	0.0298	0.0444	0.0587	0.0729	0.0868	0.1005
BB	0.0281	0.2459	0.5339	0.5533	0.7113	0.7523	0.7629
PC	0.2871	0.5167	0.6470	0.7251	0.7760	0.8113	0.8372
WPC	0.5374	0.7027	0.7822	0.8288	0.8593	0.8808	0.8967

Table 3: Cumulative explained weighted variability from the first seven dimensions using the standard construction, the Brownian bridge construction, the principal component construction and the weighted principal component construction, based on dimension $d = 360$, $T = 30$, and the mean-reverting coefficient $a = 0.1$. The weights are equal to $e^{-0.09t}$.

One possible remedy is to represent f as a function of a different discrete process. More generally, assume that we have an integrand of the following form

$$f(x) = \phi\left(\sum_{i=1}^{d} \psi_i \left(\sum_{j=1}^{d} c_{ij} x(t_j)\right)\right),$$

where $\phi, \psi_1, \ldots, \psi_d$ are smooth non-linear functions and $\{c_{ij}\}$ are constants. We recommend replacing x by Cx, where $C = (c_{ij})$, before using any principal component methods. This makes the integrand more dependent on the global features of underlying process.

The bounds (18), (19) and (22) all imply that f is continuous. However, this is not true for many types of options, including barrier options. If f only has small discontinuities, it is possible to add a small positive constant to the right of the bounds. However, when discontinuities dominate the behavior of f, the above analysis doesn't offer a convincing argument why a principal components construction should give a lower effective dimension than the Brownian bridge construction, for example.

A heuristic method to find weights

For some problems it is straightforward to specify reasonable weights, but in other cases we will need to use some numerical procedure. We want to find a fast heuristic procedure that is applicable even in situations when only a few thousands payoff function evaluations can be afforded, including those used to find the weights. A first idea might be to generate many sample paths and find a set of weights that minimize $\sum_{i=1}^{d} {w_i}^2$ under the constraint that (19) holds for all pairs of the sample paths. This leads to a large linear programming problem that is more complicated to solve than the original Monte Carlo simulation.

We will modify the above idea in two ways in order to make it feasible: 1) we generate paths in such a way that most of the terms in (19) are equal to zero, and 2) we requiring (19) to hold only for a subset of all possible combinations of simulated paths. The following offers a reasonable heuristic:

1. Divide $t_1, \ldots, t_d$ into L intervals, so that each interval contains approximately the same number of time points. Let $I_1, \ldots, I_L$ denote the set of indices for the different intervals. The weights are assumed to be constant in each interval. Let $w(I_k)$ denote the value of the weights in interval I_k.

2. Simulate M paths of $x(t)$. The paths should be uniformly distributed, so it is preferable to generate the paths with a low discrepancy sequence combined with the principal component construction. Let $\{x^j\}_{j=1,\ldots,M}$ be the results of this simulation.

3. Let Δ^k be a vector in which the ith element is equal to $\frac{1}{2}\sqrt{\mathrm{Var}\, x(t_i)}$ if $i \in I_k$ and zero otherwise.

4. Evaluate $f(x^j + \Delta^k)$ and $f(x^j - \Delta^k)$ for $j = 1, \ldots, M$ and $k = 1, \ldots, L$.

5. Find the smallest weights under which the following constraints are satisfied

$$\left|f(x^j \pm \Delta^k) - f(x^j)\right|^2 \leq \sum_{i=1}^{d} w_i^2 ({\Delta^k}_i)^2 = w(I_k)^2 \|\Delta^k\|_2^2,$$

for all $j = 1, \ldots, M$ and $k = 1, \ldots, L$. It follows that

$$w(I_k) = \frac{1}{\|\Delta^k\|_2} \max_{j=1,\ldots,M} \left|f(x^j \pm \Delta^k) - f(x^j)\right|. \tag{23}$$

The size of the elements in Δ^k are chosen to satisfy two criteria. First, $\|\Delta^k\|_2$ needs to be large enough to make (23) robust against small discontinuities in f. Second, we don't want (18) to be dominated by some extremely unlikely path, so $\|\Delta^k\|_2$ needs to be small enough so that $\{x^j \pm \Delta^k\}$ all are paths with not too small probability.

This weighting method requires $M(2L+1)$ evaluations of the discounted payoff. A reasonable choice of M and L might be $M = 32$ and $L = 10$, which results in 672 evaluations of the discounted payoff function. This choice of (M, L) was somewhat robust for the mortgage-backed security problem discussed in the next section. If one intends to price several instruments with the same paths, it is possible to use an average of the weights derived for these instruments. One could also develop a good set of weights for one problem and use them for related problems. It is possible that the above algorithm returns zero weights. One way to avoid division by zero in (20), it to first generate only the time-steps with non-zero weight and then later fill in the remaining points using their conditional distribution. Another simpler solution is to change the zero weights to a value that is small compared to the non-zero weights.

The integrand, f, in the heuristic above should be the resulting integrand after eventual control variates and/or antithetic variates have been applied.

Other related approaches

A different approach to generate paths is to approximate the integrand with a second order Taylor approximation and use a matrix A that minimizes the effective dimension of the approximation. Morokoff (1998) finds a matrix A that diagonalizes the quadratic term, such that the elements in the resulting diagonal matrix appear in order of decreasing absolute value. This gives an effective dimension of one in the superposition sense. This choice of A also gives a minimal effective dimension in the truncated sense when antithetic variates are used to remove the linear term. We recall that $\mathrm{Var}(f) = \mathrm{Var}(\mathrm{E}[f|z_1, \ldots, z_k]) + \mathrm{E}(f - \mathrm{E}[f|z_1, \ldots, z_k])^2$, so minimizing the effective dimension in the truncated sense is equivalent to maximizing $\mathrm{Var}(\mathrm{E}[f|z_1, \ldots, z_k])$.

Theorem 2. *If Q is an orthogonal matrix, $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$ a diagonal matrix with $|\lambda_1| \geq \ldots \geq |\lambda_d|$ and $\{z_i\}_{i=1}^d$ independent standard normals, then*

$$\mathrm{Var}\big(\mathrm{E}\big[(Qz)'\Lambda(Qz)|z_1, \ldots, z_k\big]\big) \leq \mathrm{Var}\big(\mathrm{E}\big[z'\Lambda z|z_1, \ldots, z_k\big]\big) = 2\sum_{i=1}^{k} \lambda_i^2$$

Proof. Let $Z_1 = (z_1, \ldots, z_k)'$ and $Z_2 = (z_{k+1}, \ldots, z_d)'$ be a partition of z and

$$\begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix}$$

a partition of $G = Q'\Lambda Q$ were G_{11} is a $k \times k$ matrix. We now have that $\mathrm{E}\big[(Qz)'\Lambda(Qz)|z_1, \ldots, z_k\big] = Z_1' G_{11} Z_1 + \mathrm{tr}(G_{22})$, which implies that

$$\begin{aligned}\mathrm{Var}\big(\mathrm{E}\big[(Qz)'\Lambda(Qz)|z_1, \ldots, z_k\big]\big) &= \mathrm{Var}(Z_1' G_{11} Z_1) \\ &= \mathrm{Var}\big((UZ_1)' \,\mathrm{diag}(\mu_1, \ldots, \mu_k)(UZ_1)\big) \\ &= 2\sum_{i=1}^{k} \mu_i^2,\end{aligned}$$

where $U' \operatorname{diag}(\mu_1, \ldots, \mu_k)U$ is a spectral decomposition of G_{11} with the eigenvalues in order of decreasing absolute value. The last equality uses the fact that UZ_1 is a vector of independent standard normals. Assume that $S = \{\mu_1, \ldots, \mu_i\}$ consists of p positive numbers and n non-positive numbers. It follows from Rayleigh Theorem (see Lancaster and Tismenetsky 1985) that there exist p lambdas larger than the smallest positive number of S and n lambdas smaller than the largest non-positive number of S. Hence there exist i lambdas with absolute values larger than $|\mu_i|$ which implies that $|\lambda_i| \geq |\mu_i|$ for $i = 1, \ldots, k$. □

This theorem was also proved independently by Fox (1999). Glasserman et al. (1998) prove the corresponding result when the logarithm of integrand is approximated by a second order Taylor approximation. The performance of this approach depends on how well the integrand can be approximated by a quadratic function.

5 Comparison of Methods for MBS

5.1 Introduction to MBS securities

Mortgage-backed securities(MBS) and *collateralized mortgage obligations* (CMO) are an increasingly popular and important class of financial instruments. Each is created by a financial institution which pools a set of mortgages and sells units to investors. Ownership of a unit entitles the owner to a cash flow from the principal and/or interest of the mortgage payments. All of the mortgages have some given payment schedule and often permit prepayment without penalty. Thus the owners of the pool units are subject to interest rate risk and prepayment risk. In a pass-through MBS, the owners are treated equally and bear equivalent risks. In CMOs the total pool is divided into classes or *tranches*, each having different prepayment and interest rate risk characteristics. For example, one tranche may be paid off completely before any other tranche receives any prepayments. MBS and CMOs are a particularly relevant instruments to use to study the effectiveness of new simulation methodology because they are high dimensional (monthly payments over a 30 year period results in a 360 dimensional problem) and their pricing is sensitive to the term structure of interest rates, thus the simulation must include term structure models. These characteristics make simulation the only viable approach for pricing and hedging MBS. Davidsson and Herskovitz (1993) give a good introduction to mortgage-backed securities.

In this paper we study MBS simulation pricing using the Hull-White term structure model and a widely used prepayment model described by Richard and Roll (1989). More advanced models could be used, however this will suffice for our study of simulation methodology.

Richard and Roll identify four factors that must be included in a prepayment model:

1. *Refinancing incentives*: measured by the mortgage coupon rate divided by the mortgage refinancing rate, C/R.

2. *Seasoning or age of the mortgage*: newer loans tend to prepay slower than older loans.

3. *Seasonality*: home-owners are more likely to move during certain months than others.

4. *Premium burnout*: the tendency for prepayment to diminish over time, even when refinancing incentives are favorable. Richard and Roll note that the more the prepayment option has been deep in the money, the more burned out the pool must be.

The prepayment rate, *CPR*, is computed as the product of these four factors. We take the parameters from an example on Numerix's homepage, `www.numerix.com`, as of August 1, 2000. It is thought to be a representative MBS pricing problem. The details can be found in the Appendix.

Instrument	Mean	Std. Dev.
Tranche A	197.6356719	0.0000097
Tranche B	36.9147492	0.0000045
Tranche C	100.176674	0.000013
Tranche D	78.911907	0.000014
IO	155.95061	0.00015
PO	257.68840	0.00014
pass-through	413.639002	0.000030

Table 4: Mean value and standard deviation for the different instruments (in millions of dollars).

5.2 Specific problem simulated

For our test problem, the investors are divided into four tranches, A through D. All principal repayments (both scheduled and prepayments) are used to payoff tranche A completely. Principal repayments are then used to payoff tranche B and so forth. All four tranches receive interest on their outstanding balance during the entire time period. The collateral underlying the CMO is a fixed-rate pass-through MBS with a total par value of 400 million dollars, a net coupon of 7.5%, a weighted average coupon (WAC) of 8.125%, and a weighted average remaining maturity (WAM) of 357 months. The difference in the net coupon and the weighted average coupon comes from the fees that are subtracted from the households' coupon payments until they reach the investor. Formulas describing exactly how the cashflow is distributed are presented in the Appendix.

To specify the Hull-White interest rate model, we used a flat initial term structure, $f(0,t) = 0.065$ for all t, mean-reverting coefficient $a = 0.1$ and volatility $\sigma = 0.01$.

5.3 Numerical results

The low discrepancy sequence used is a Sobol sequence generated by FINDER obtained from Columbia University (see Traub and Papageorgiou 1996). The algorithm by Moro (1995) was used to transform the Sobol numbers into standard normals. The function `randn` in MATLAB® version 5 was used to generate pseudo-random numbers having a normal distribution.

We estimated the true value of the different instruments being priced by averaging 200 independent random shifts of 2^{19} fixed points from a Sobol sequence, whose paths were generated with the weighted partial principal component construction, $k = 8$, (see Table 4). These random shifts are due to Cranley and Patterson (1976) and Tuffin (1996).

Next we present the results of the simulation study. Seven different instruments are considered: ownership of any of the tranches A, B, C or D; interest only; principal only; and pass-through (both interest and principal combined). Each instrument is priced with large number of different path generation methods.

In all figures in Section 8 the x-axis gives the number of paths, n, (which range from 256 to 16,384) plotted on a log scale. In the methods that use antithetic variates n denotes the number of evaluations of the payoff function, since this is the most expensive operation (see Table 6). The y-axis gives the relative root-mean-squared-error (RMSE) obtained from 50 independent computations, where the error is measured as the difference from the value in Table 4. For the Sobol sequence calculations the 50 independent values were obtained first by dividing the Sobol sequence into 50,000 blocks of size n, then randomly selecting 50 without replacement. (These 50 blocks are not independent in a statistical sense.) Each path is used to calculate the value of all seven instruments. The pairs of figures compare the results without and with the use of antithetic variables. The estimated slopes of the linear fits obtained using ordinary least-squares. A selection of them are presented in Table 5. The estimated convergence rates for standard Monte Carlo vary from 0.47 to 0.54. However, the theoretical convergence rate for Monte Carlo based on pseudo-random numbers is 0.5, so the estimates in Table 5 are rather imprecise.

Method	Tran. A	Tran. B	Tran. C	Tran. D	IO	PO	PT
STD	0.52	0.54	0.53	0.52	0.49	0.50	0.52
PC	0.77	0.70	0.77	0.89	0.68	0.75	0.83
BB	0.77	0.72	0.81	0.88	0.77	0.81	0.85
WPC	0.85	0.86	0.91	1.00	0.71	0.80	0.97
WPC2	0.93	0.88	0.91	0.98	0.72	0.83	0.91
STD-anti	0.54	0.51	0.50	0.47	0.53	0.48	0.51
PC-anti	0.65	0.64	0.78	0.68	0.54	0.54	0.71
BB-anti	0.64	0.68	0.66	0.71	0.61	0.58	0.72
WPC-anti	0.77	0.72	0.73	0.76	0.64	0.64	0.71
WPC2-anti	0.81	0.83	0.78	0.78	0.66	0.63	0.79

Table 5: Estimated convergence rates.

The weighted principal component construction

Next, we turn to the figures in Section 8.1, where the results for 5 different path generation methods (Standard Monte Carlo - STD, Brownian bridge - BB, principal component - PC, and two version of weighted principal component - WPC and WPC2) are presented.

The weights used in WPC are based on the following observation: A fixed-rate mortgage is designed to make total cashflow constant over time if no prepayments occur. Since a dollar at time t is worth $P(0,t) = e^{-\int_0^t f(0,s)ds}$ at time 0, a reasonable weighting scheme might be $e^{-0.065t}$. However, since prepayments are dependent on the previous interest rate, and we have more cashflow at earlier dates due to prepayments, a good weighting scheme should decrease faster. Exactly how much faster it should decrease is hard to say. The WPC method uses a weighting matrix D with entries of the form $e^{-0.09t}$ for $0 \leq t \leq T$. This was our first guess. It turns out that this weighting matrix works well for all seven instruments. The second set of weights, used in WPC2, is calculated using the heuristic described in Section 4.6. This method uses a different weighting matrix D for each individual instrument. Obviously, there is a large extra cost to generate different paths for each instrument.

Two features hold for all instruments with or without antithetic variables: the STD method has the largest RMSE, while the WPC and WPC2 have the smallest. In fact, the RMSE for WPC and WPC2 with $n = 256$ is roughly equivalent to the RMSE for STD with $n = 16,384$. This is a reduction of a factor of 64. Neither PC or BB perform as well as WPC and WPC2. The WPC2 method performs better than WPC in some cases, and not significantly worse than WPC in any cases. The improvements over WPC are most noticeable for Tranche A, since the heuristic correctly finds that the payoff only depends on the early time-steps. Tranche D is the instrument where the difference is smallest between the two weighted principal component methods and the other methods. PC and BB are roughly equivalent for all instruments. PC is slightly better than BB for Tranche D, but the opposite holds for PO. The low-discrepancy methods here, PC, BB, WPC and WPC2, are far superior to the standard method, and this is true throughout all the instruments.

The antithetic variables generally move the curves down but also reduce the slopes. Antithetic sampling substantially reduces the error for Tranche D, which indicates that it is almost a linear function.

The partial principal component construction

In the figures in Section 8.1 we observed the superiority of WPC and WPC2 over the other methods. We must recall, however, that the principal components method requires substantially more computation than BB or STD. To overcome this computational problem we developed the PPC and WPPC methods which use only a small number of the principal components. The figures in Section 8.2 addresses this issue. The first seven pairs of figures compare the performance of PPC (using $k = 8$ or 16) with PC, the full principal components method. The second seven pairs of figures compare the relative performance

Method	Part 1	Part 2	Part 3	Total
STD	0.041	0.112	3.619	3.772
PC/WPC	0.092	3.665	3.619	7.376
BB	0.092	0.158	3.619	3.869
PPC/WPPC($k = 8$)	0.094	0.439	3.619	4.150
PPC/WPPC($k = 16$)	0.096	0.538	3.619	4.249
STD-anti	0.020	0.056	3.619	3.695
PC/WPC-anti	0.046	1.833	3.619	5.498
BB-anti	0.046	0.080	3.619	3.745
PPC/WPPC($k = 8$)-anti	0.047	0.220	3.619	3.885
PPC/WPPC($k = 16$)-anti	0.048	0.269	3.619	3.934

Table 6: Computation time for $n = 512$ (in seconds).

of WPC with WPPC ($k = 8$ or 16). We hoped that WPPC would be comparable to WPC for some small value of k. The results for all cases are very encouraging. PC/WPC is superior to PPC8/WPPC8 for most instruments. However, in all cases PC/WPC and PPC16/WPPC16 are essentially equivalent. Consequently, WPPC16 appears to be very effective method which are also computationally efficient. While a much more comprehensive study will be needed to ensure that these results apply across a wide range of MBS-pricing problems, these initial results are very encouraging. Furthermore, the study did include a variety of different MBS instrument types.

We also tried the subsequence path method with $k = 40$; however, the results were significantly worse than those for PC. We conjecture that the reason for such behavior is that the earliest part of the path is most important in this problem. With $k = 40$, the time points that are first filled in are $9, 18, 27, \ldots$. This means that the entire path between 0 and 9 is left to be filled in with only ordinary pseudo random numbers. Since the subsequence method has almost the same explained variability as PC (see Table 1-2), this indicates that explained variability is not a very good measure of the path generation method's performance, at least not for the problems where certain parts of the underlying asset's path are much more important than the other.

In Table 6 we present the computation times for the methods listed, once they have been initialized. The times are based on our implementation in MATLAB® and C. Part 1 gives the time it takes to generate the vectors of standard normals. Part 2 gives the time it takes to generate the paths of $x(t)$ from the standard normals. Note the large times associated with PC and PC-anti. Part 3 gives the time it takes to generate the value of the four different tranches given the bond prices and discounting factors computed from the paths of $x(t)$. The time needed for the weighted versions is the same as for the unweighted versions. Clearly, the payoff function evaluation is the dominant part of the computation (except for PC). Consequently, STD, PPC and BB all require approximately the same total time. The additional multiplications needed for PC make this method almost twice as expensive as the other methods. For a two factor model with the same number of time steps PC would be approximately 3 to 4 times more expensive than the other methods, since the work required to multiply a $d \times d$ matrix by a vector is $O(d^2)$. Taking into account both the accuracy obtained and the computational effort, it appears that WPPC is a very attractive procedure.

Quadratic approximation methods

The figure in section 8.3 compares PC and WPC2 with three methods based on quadratic approximation of the payoff function. The three methods are:

1. QA - this method generates the paths by the matrix that diagonalizes the quadratic approximation of the payoff function, such that the diagonal elements appear in decreasing absolute value (see Section 4.6).

2. ISS - this method by Glasserman et al. (1998) uses a combination of importance sampling and stratification based on a quadratic approximation of the logarithm of the payoff function. The importance sampling changes the mean of z so that the linear term is removed from the logarithm of the payoff. The importance sampling is motived by large deviation analysis. They stratify $u'z$, where u is chosen to be optimal for the log-quadratic approximation. We use 64 equiprobable strata with a equal number of samples in each stratum.
3. ISS-Sobol - this a version of ISS uses Sobol numbers instead of stratification. The paths are generated by the matrix that minimizes the effective dimension of the log-quadratic approximation of the payoff function.

It requires $O(d^2)$ evaluations of f to numerically calculate the Hessian of f (or $\log f$). This is much more evaluations that the ones used for the rest of the simulation, even in the case when $n = 16,384$. In order to reduce the computational effort, we looked for approximation of the eigenvectors of the Hessian in a subspace spanned by the first 40 columns of the matrix A that corresponds to the Brownian bridge path generation. This approximation is described in detail on page 135 in Glasserman et al. (1998).

Without antithetic variates, we have that ISS-Sobol is best method except for IO and PO. The ISS method has a slower convergence rate than the other methods since it uses pseudo random numbers; it is the worst method then n is larger than 4,096, but the second best method for $n = 256$, except for IO and PO. ISS has a larger error than standard Monte Carlo for instrument PO, which means that the importance sampling in this case increased the variance instead of decreasing it. The QA method is slightly worse than WPC2 without antithetic variates

With antithetic variates, we have that WPC2 is the best method followed by ISS-Sobol and QA that are roughly equivalent. Both ISS and ISS-Sobol don't benefit from the antithetic variates (except for PO). This is excepted since the importance sampling removes the variance from the linear term of the log-payoff.

Generalized Faure

We also compared the performance of generalized Faure (GFaure) points with Sobol points. Both sequences are generated by the FINDER software. The results in shown in Section 8.4. Both STD-Sobol and STD-GFaure perform better than STD-random but worse than BB-Sobol, PC-Sobol, WPC-Sobol. STD-Sobol and STD-GFaure have roughly the same errors, except for Tranche A, where STD-Sobol is slightly better, and tranche D, where STD-GFaure is slightly better. GFaure performs better than Sobol then antithetic variates are used together with standard path generation. STD-Sobol-anti performs worse than STD-random-anti, where this does not happen for STD-GFaure-anti. Our results are consistent with the results of Acworth et al. (1998), but are somewhat different from the results of Papageorgiou and Traub (1996), who reported that generalized Faure sequences "usually achieve the same accuracy as the Sobol points 2.5 to 6.5 times faster". A possible explanation could be that they measure errors in a different way. We only measure RMSE when the number of points is equal to powers of 2, which is likely to benefit the Sobol numbers since they are constructed in base 2. BB-GFaure, PC-GFaure, WPC-GFaure with and without antithetic variables all perform much worse than the corresponding versions with Sobol numbers, except that WPC-GFaure-anti has approximately the same error as WPC-Sobol-anti for IO and PO. We concluded that Sobol points are superior to GFaure points when the paths are generated by a Brownian bridge or a principal component method that concentrates the variability to the first few dimensions. We believe that the reason is the very uniform one- and two–dimensional projections of the first few coordinates of the Sobol sequence.

6 Conclusions

This paper presented new methods for simulating the prices of financial instruments based on low-discrepancy sequences. The new methods, called partial principal components and subsequence path generation, extend the work of Caflisch et al. (1997) and Acworth et al. (1998) based on the Brownian bridge and principal components constructions. The methods are designed to reduce the effective dimension of the integrand. The new methods coupled with a weighted version of the partial principal components method generate paths using the low dimensional components of the sequence for the path components having the highest variability. The methods were applied to the pricing of a variety of types of mortgage-backed securities assuming a Hull-White term structure model. The weighted principal components method was shown to outperform the other methods, often by a substantial amount. More importantly, the partial principal components and weighted partial principal components were shown to nearly achieve the performance of the corresponding full procedures, sometimes using as few as 8 principal components and always when using 16 principal components. This is an important finding, since the full principal components method requires substantial extra computation time, while the partial principal components versions are comparable to the standard or Brownian bridge path generation methods in terms of computational requirements. The experiments clearly point out the need to use weighting procedures, especially in the MBS problem where early prepayment causes much greater importance for the early parts of the path. A more comprehensive study is needed to confirm the benefits of these new procedures, however the results presented in this paper are very promising.

Acknowledgments

We thank Anargyros Papageorgiou and Joseph Traub for providing us with their quasi-random number generation package FINDER. We also thank Paul Glasserman, Philip Heidelberger and the anonymous referees for their helpful suggestions and comments. Part of the research was conducted by Fredrik Åkesson during his internship at IBM Watson Research, Yorktown Heights, NY.

7 Appendix

In this Appendix we summarize the details of the mortgage-backed security model that was used to conduct the simulation experiments.

Formulas for the prepayment model

The index t denotes the month for all variables except for the bond prices. Let $B(t)$ be the mortgage balance in the beginning of month t. The initial mortgage balance, $B(0)$, is equal to $400 \cdot 10^6$. The burnout multiplier, $BM(t)$, is equal to $BM(t) = 0.3 + 0.7\frac{B(t)}{B(0)}$ The monthly multiplier, $MM(t)$, for the ith month is assumed to be given by

$$MM(i) = (.94, .76, .74, .95, .98, .92, .98, 1.1, 1.18, 1.22, 1.23, .98)$$

where $MM(0) = 0.94$. The seasoning multiplier is given by $AGE(t) = \min\left(1, \frac{t}{30}\right)$. We model the refinancing coupon using the 10-year zero-coupon rate:

$$RC(t) = -\frac{\log(P(t/12, t/12 + 10))}{10}$$

where $P(t,T)$ denotes the bond price at time t with maturity T in years. The refinancing incentive, $RI(t)$, is equal to $RI(t) = 0.28 + 0.14 \arctan(-8.571 + 430 \cdot (WAC - RC(t)))$ and the annualized prepayment rate, $CPR(t)$, is equal to

$$CPR(t) = RI(t) \cdot AGE(t) \cdot MM(t) \cdot BM(t).$$

The single month mortality, $SMM(t)$, is the proportion of the outstanding balance that is prepaid during month t. It is given by $SMM(t) = 1 - \sqrt[12]{1 - CPR(t)}$.

Cashflow from the pool of underlying mortgages

The cashflow with index t is the cashflow during month t. The mortgage payment, $MP(t)$, is equal to

$$MP(t) = B(t)\frac{WAC/12}{1-(1+WAC/12)^{-WAC+t}}.$$

The scheduled principal payment, $SP(t) = MP(t) - \frac{WAC}{12}B(t)$, is chosen to make the sum of the interest rate and the scheduled principal payments constant for all months. The principal prepayment, $PP(t)$, is equal to $PP(t) = SMM(t)\ (B(t) - SP(t))$. The total principal payment, $TPP(t)$, is equal to the scheduled principal payment plus the principal prepayment, $TPP(t) = SP(t) + PP(t)$. The reduction in the mortgage balance for each month is given by $B(t+1) - B(t) = TPP(t)$.

Principal disbursements to tranches

Let $PP_a(t)$, $PP_b(t)$, $PP_c(t)$ and $PP_d(t)$ denote the principal payments to tranches A-D during month t. Let $B_a(t)$, $B_b(t)$, $B_c(t)$ and $B_d(t)$ denote the balances for tranches A-D at the beginning of month t. The initial balances are $B_a(0) = 194.5 \cdot 10^6$, $B_b(0) = 36 \cdot 10^6$, $B_c(0) = 96.5 \cdot 10^6$ and $B_d(0) = 73 \cdot 10^6$. The total prepayments, $TPP(t)$, are used to reduce B_a until it is completely paid off. When B_a is paid off, the prepayments are then used to reduce B_b. This continues until B_b, then B_c and finally B_d are all paid off. Let $INT_a(t)$, $INT_b(t)$, $INT_c(t)$, $INT_d(t)$, denote the interest payments to tranches A-D in month t. We have $INT_a(t) = B_a(t)\frac{C_a}{12}$ with similar formulas holding for tranches B-D. The net-coupon is equal from all four tranches, $C_a = C_b = C_c = C_d = 0.075$. The cashflow occurs at the end of month t. For a given interest rate path the discounting factor, $Discount(t)$, for cashflow during month t is

$$Discount(t) = e^{-\int_0^{\frac{t+1}{12}} r(s)ds}$$

The present discounted value is equal to the sum of all discounted cashflows,

$$PV_a = \sum_{t=0}^{WAM-1} Discount(t) \cdot (PP_a(t) + INT_a(t)).$$

The same formula holds for tranches B-D. The price of tranches A-D are the expected value of their present value.

References

[1] Acworth, P., M. Broadie, and P. Glasserman, "A Comparison of Some Monte Carlo and Quasi Monte Carlo Techniques for Option Pricing" in [25], 1-18.

[2] Åkesson, F. and J.P. Lehoczky, "Discrete Eigenfunction Expansion of Multi-Dimensional Brownian Motion and the Ornstein-Uhlenbeck Process," Working Paper, Department of Mathematical Sciences, Carnegie Mellon University, 1998.

[3] Black, F. and P. Karasinski, "Bond and Option Pricing when Short Rates are Lognormal," *Finan. Analysts J.*, 47, 4 (1991), 52-59.

[4] Boyle, P., M. Broadie and P. Glasserman, "Monte Carlo Methods for Security Pricing," *J. Econom. Dynam. Control*, 21, 8-9 (1997), 1267-1321.

[5] Caflisch, R.E., and B. Moskowitz, "Modified Monte Carlo Methods Using Quasi-Random Sequences," in H. Niederreiter and P.J. Shiue (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, Springer, 1995, 1-16.

[6] Caflisch, R.E., W. Morokoff, and A. Owen, "Valuation of Mortgage Backed Securities Using Brownian Bridges to Reduce Effective Dimension," *J. Comp. Finance*, 1, 1 (1997), 27-46.

[7] Cranley, R., and T.N.L. Patterson, "Randomization of Number Theoretic Methods for Multiple Integration," *SIAM J. Numer. Anal.*, 13, 6 (1976), 904-914.

[8] Davidson, A.S. and M.D. Herskovitz, *Mortgage-Backed Securities: Investment Analysis & Advanced Valuation Techniques*, Irwin Professional Pub., 1993.

[9] Duffie, D., "Dynamic Asset Pricing Theory," Second Ed., Princeton University Press, 1996.

[10] Dupire, B. and A. Savine, "Dimension Reduction and Other Ways of Speeding Monte Carlo Simulation" in B. Dupire (Ed.), *Monte Carlo: Methodologies and Applications for Pricing and Risk Management*, Risk Publications, 1998, 51-63.

[11] Fishman, V., P. Fitton, and Y. Galperin, "Hybrid Low-Discrepancy Sequences: Effective Path Reduction for Yield Curve Scenario Generation," *J. Fixed Income*, 7, 1 (1997), 75-84.

[12] Fox, B., "Generating Random Normal Vectors for Quasi-Monte Carlo", Working Paper, SIM-OPT Consulting, 872 Timber Lane, Boulder, CO 80304, 1999.

[13] Glasserman, P., P. Heidelberger and P. Shahabuddin, "Asymptotically Optimal Importance Sampling and Stratification for Pricing Path-Dependent Options," *Math. Finance*, 9, 2 (1999), 117-152.

[14] Hull, J. and A. White, "Pricing Interest Rate Derivative Securities," *Review of Financial Studies*, 3, 4 (1990), 573-593.

[15] Hull, J. and A. White, "Numerical Procedures for Implementing Term Structure Models I: Single-Factor Models," *J. Derivatives*, 2, 1 (1994), 7-16.

[16] Hull, J., *Options, Futures and Other Derivatives*, Third Ed., Prentice-Hall, 1997.

[17] Joy, C., Boyle, P.P. and Tan, K.S., "Quasi-Monte Carlo Methods in Numerical Finance," *Management Sci.*, 42, 6 (1996), 926-938.

[18] Karatzas, I. and S.E. Shreve, *Brownian Motion and Stochastic Calculus*, Second Ed., Springer, 1991.

[19] Lancaster P. and M. Tismenetsky, *The Theory of Matrices*, Second Ed., Academic Press, 1985.

[20] Moro, B., "The Full Monte," *Risk*, February (1995), 57-58.

[21] Morokoff, W.J. and R.E. Caflisch, "Quasi-Random Sequences and Their Discrepancies," *SIAM J. Sci. Comput.*, 15, 6 (1994), 1251-1279.

[22] Morokoff, W.J. and R.E. Caflisch, "Quasi-Monte Carlo Simulation of Random Walks in Finance," in [25], 340-352.

[23] Morokoff, W.J., "Generating Quasi-Random Paths for Stochastic Processes," *SIAM Rev.*, 40, 4 (1998), 1765-1788.

[24] Niederreiter, H., *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, 1992.

[25] Niederreiter, H., P. Hellekalek, G. Larcher, and P. Zinterhof (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods 1996*, Springer, 1998.

[26] Ninomiya, S. and S. Tezuka, "Toward Real-Time Pricing of Complex-Financial Derivatives," *Appl. Math. Finance*, 3, 1 (1996), 1-20.

[27] Okamoto M. and M. Kanazawa, "Minimization of Eigenvalues of a Matrix and Optimality of Principal Components,", *Ann. Math. Stat.*, 39, 3 (1968), 859-863.

[28] Owen, A.B., "Latin Supercube Sampling for Very High-Dimensional Simulations," *ACM Trans. Model. Comput. Simul.*, 8, 1 (1998), 71-102.

[29] Papageorgiou, A. and J.F. Traub, "Beating Monte Carlo," *Risk*, June (1996), 63-65.

[30] Paskov, S.H. and J.F. Traub, "Faster Valuation of Financial Derivatives," *J. Portfolio Management*, 22, 1 (1995), 113-120.

[31] Paskov, S.H., "New methodologies for Valuing Derivatives," in S. Pliska and M. Dempster (Eds.), *Mathematics of Derivative Securities*, Cambridge University Press, 1996, 545-582.

[32] Richard, S.F. and R. Roll, "Prepayments on Fixed-Rate Mortgage Backed Securities," *J. Portfolio Management*, 15, 3 (1989), 73-82.

[33] Schmidt, W.M., "On a General Class of One-Factor Models for the Term Structure of Interest Rates," *Finance & Stochastics*, 1, 1 (1997), 3-24.

[34] Seber, G.A.F., *Multivariate Observations*, John Wiley & Sons, 1984.

[35] Tuffin, B., "On the Use of Low-Discrepancy Sequences in Monte Carlo Methods," Technical Report Publication Interne 1060, IRISA, 1996.

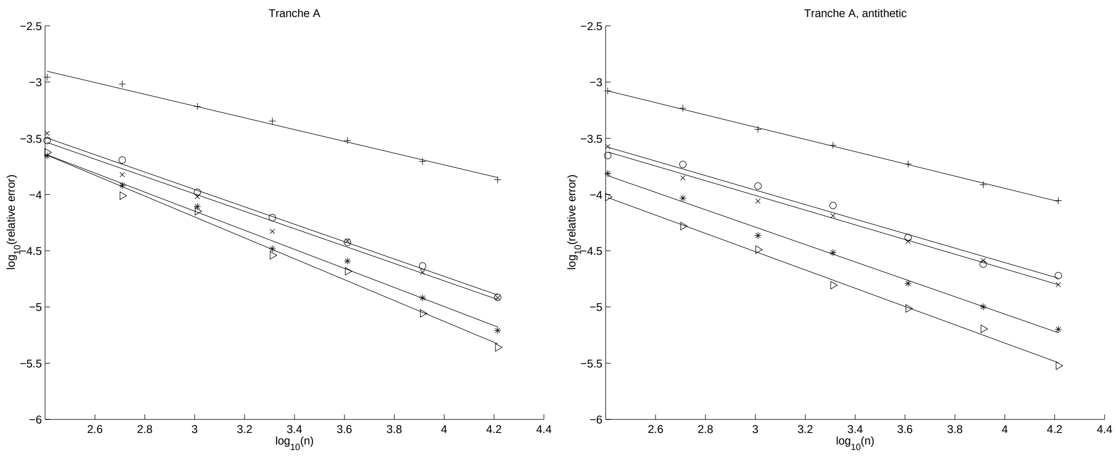
[36] Vasicek, O.A., "An Equilibrium Characterization of the Term Structure," *J. Financial Econ.*, 5 (1977), 177-188.

[37] Wood, A., G. Chan, "Simulation of Stationary Gaussian Processes in $[0,1]^d$," *J. Comput. Graph. Statist.*, 3, 4 (1994), 409-432.

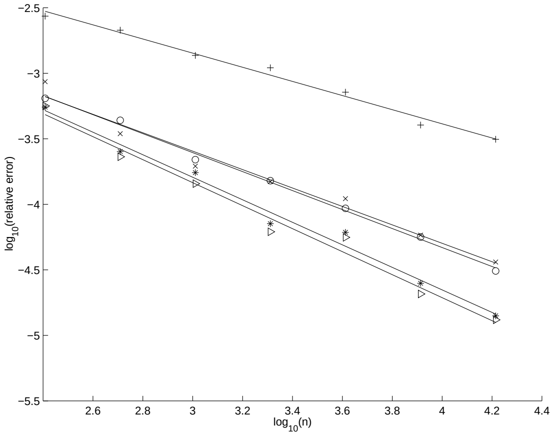
8 Figures

8.1 The weighted principal component construction

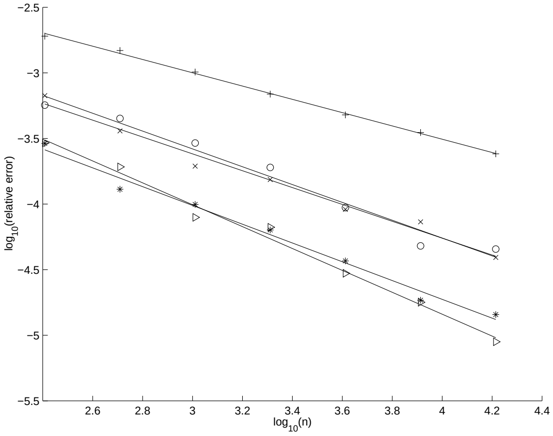
Method	Symbol
Standard path generation with pseudo random numbers	$+$
Principal component construction with Sobol numbers	$\times$
Brownian Bridge construction with Sobol numbers	o
Weighted principal component construction with Sobol numbers and $e^{-0.09t}$ weights	$*$
Weighted principal component construction with Sobol numbers and weights from heuristic	$\triangleright$



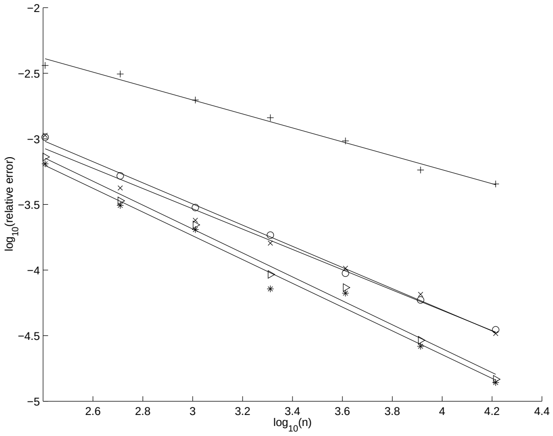
Tranche B



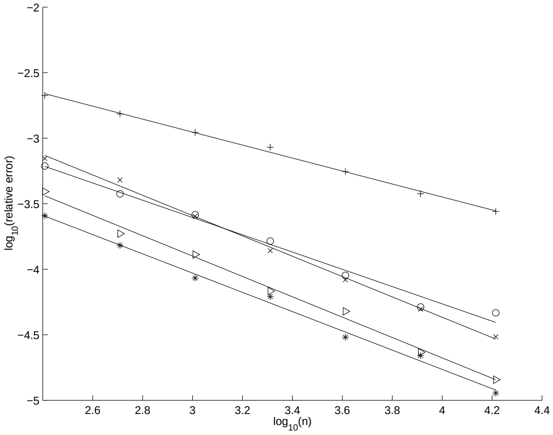
Tranche B, antithetic



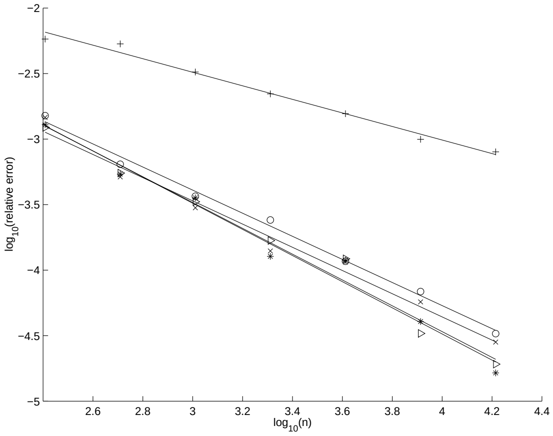
Tranche C



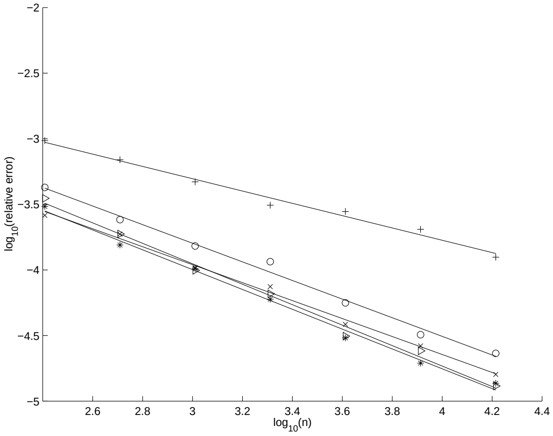
Tranche C, antithetic

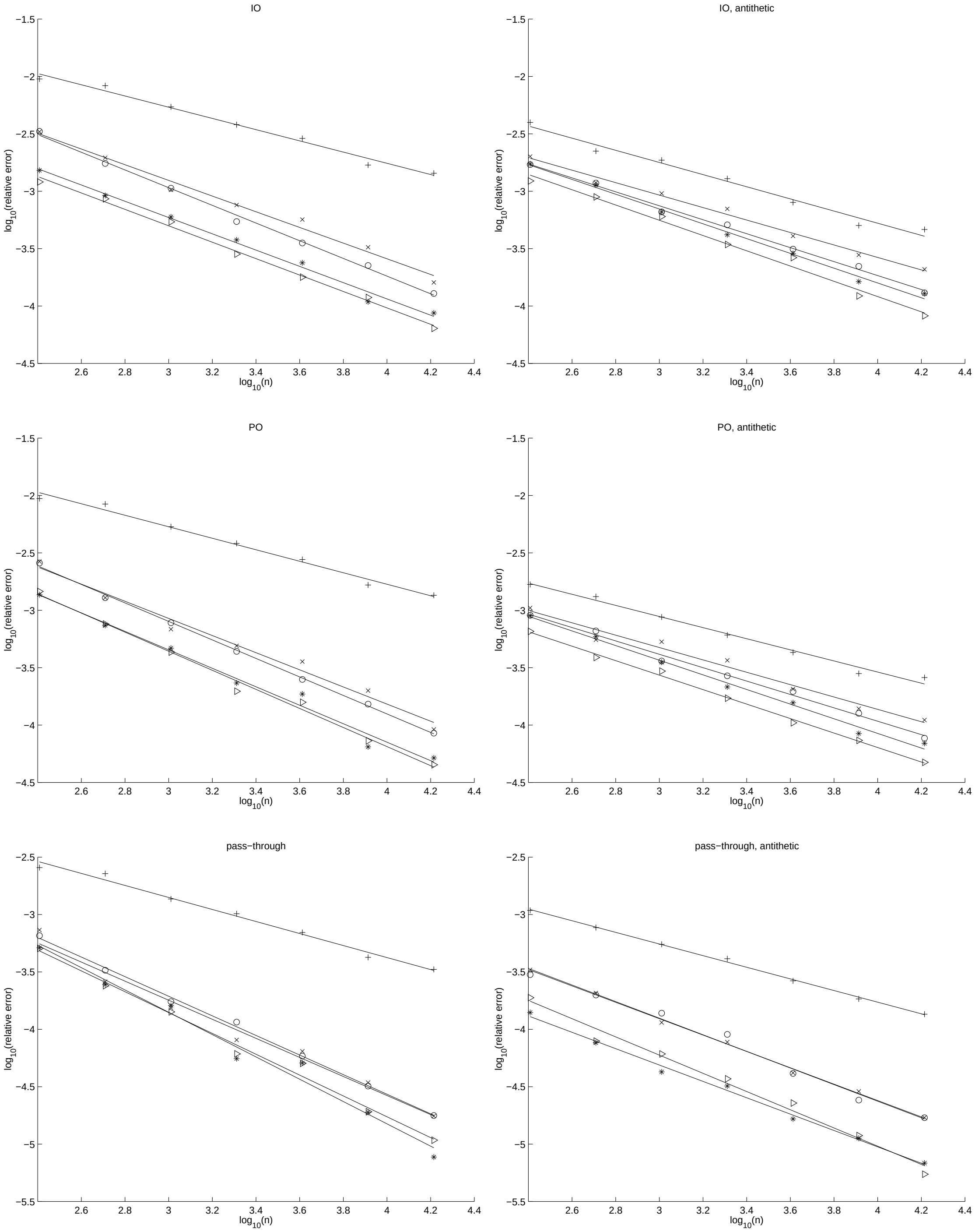


Tranche D



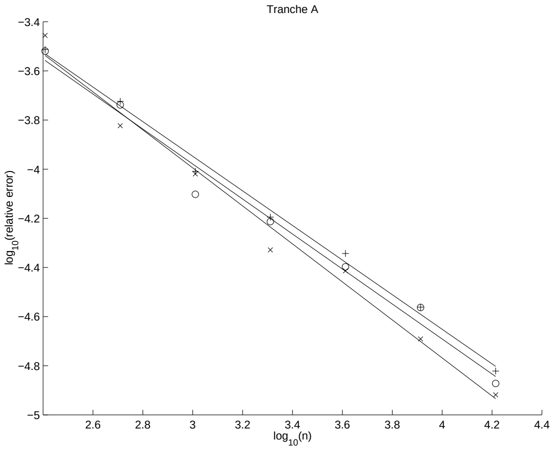
Tranche D, antithetic



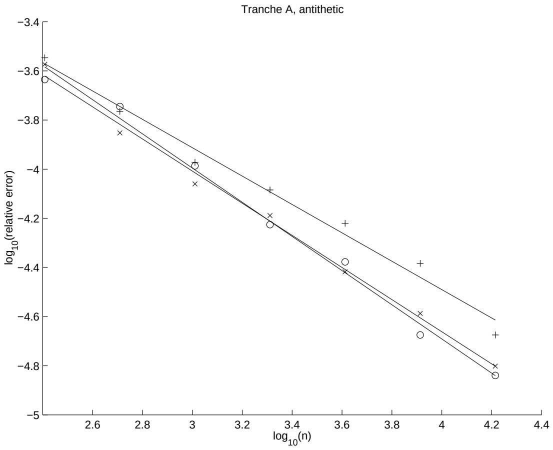


8.2 The partial principal component construction

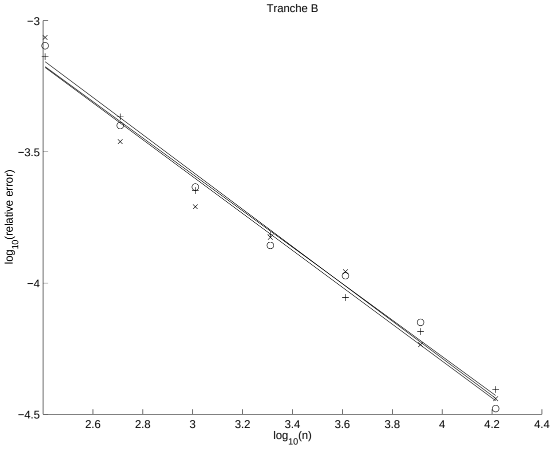
Method	Symbol
Principal component construction with Sobol numbers	×
Partial principal component construction ($k = 8$) with Sobol numbers	+
Partial principal component construction ($k = 16$) with Sobol numbers	o



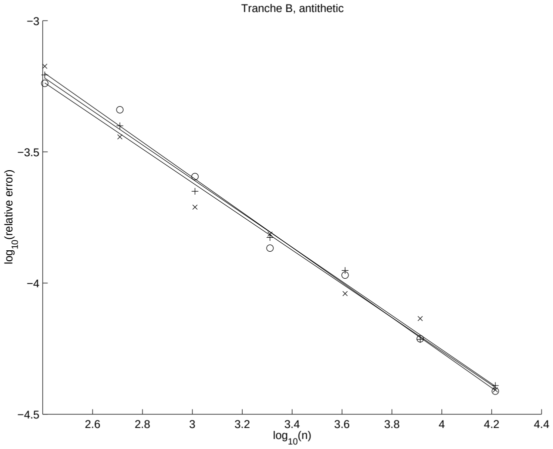
Tranche A



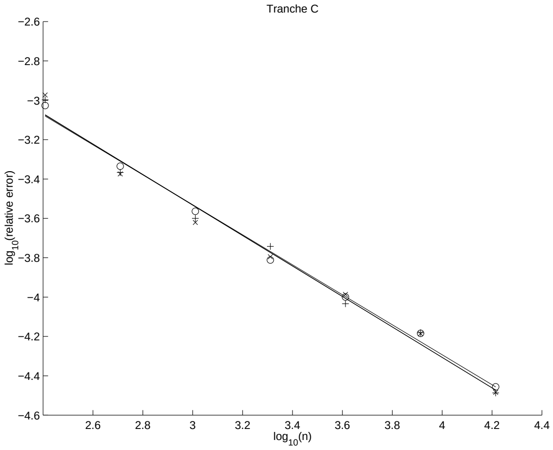
Tranche A, antithetic



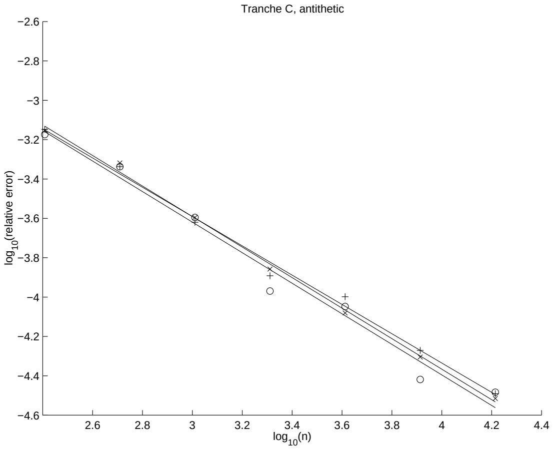
Tranche B



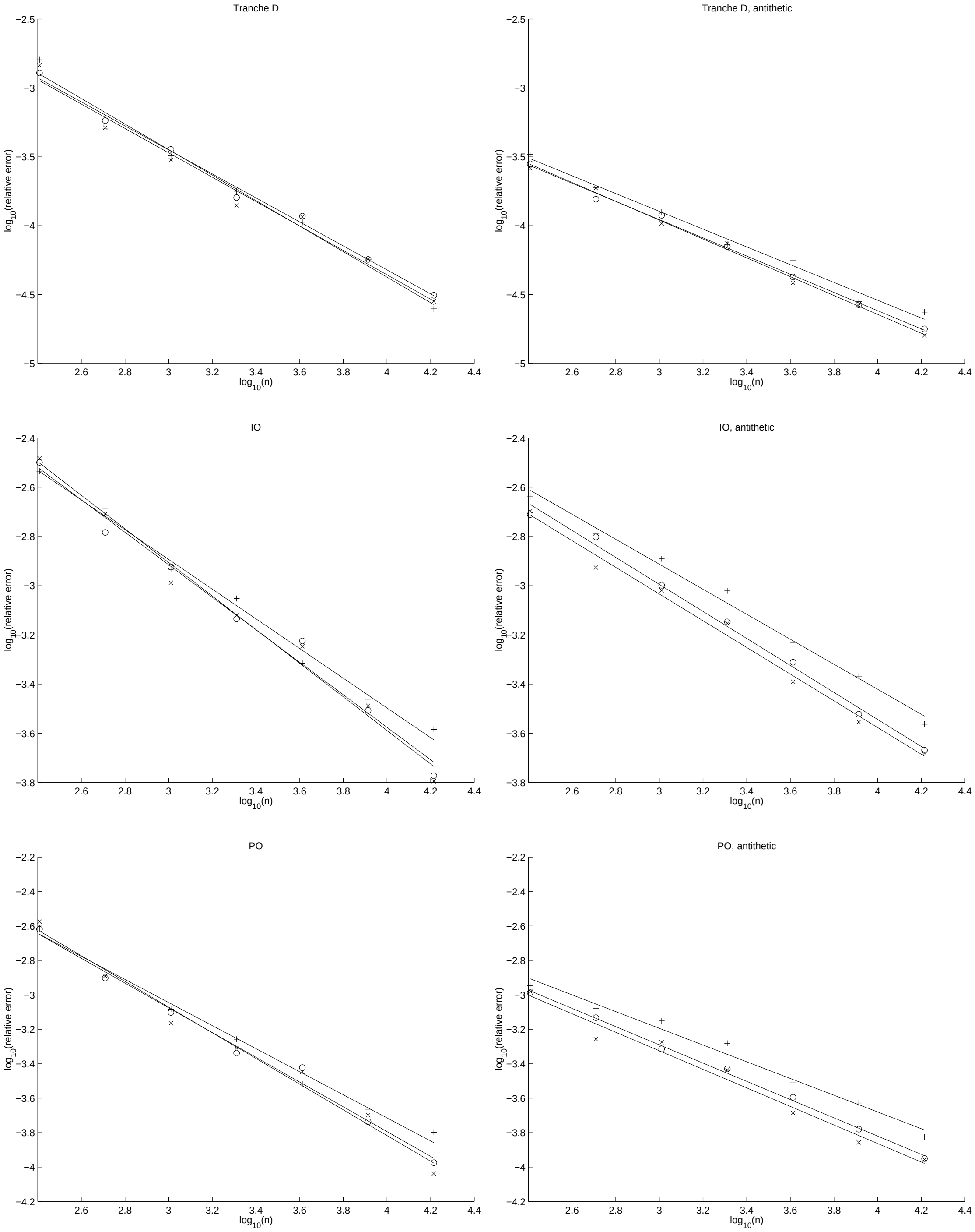
Tranche B, antithetic

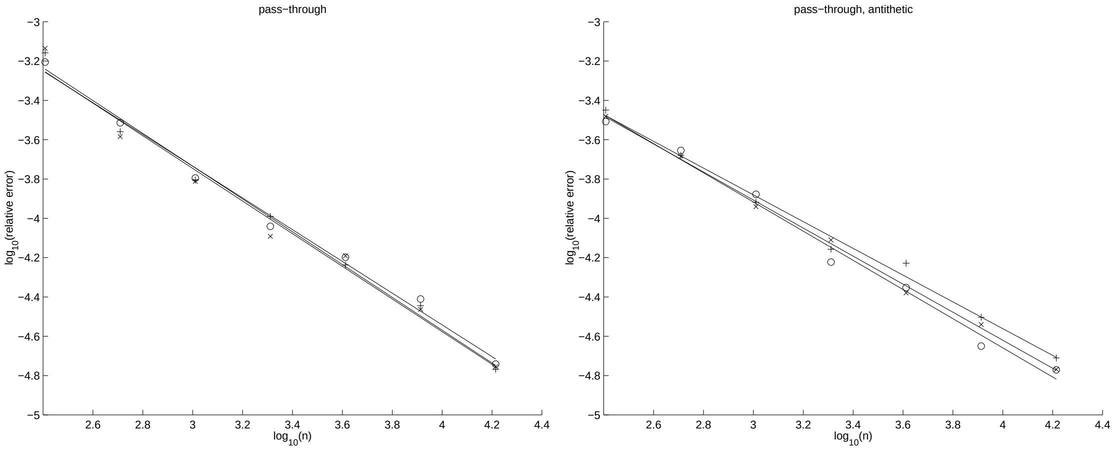


Tranche C

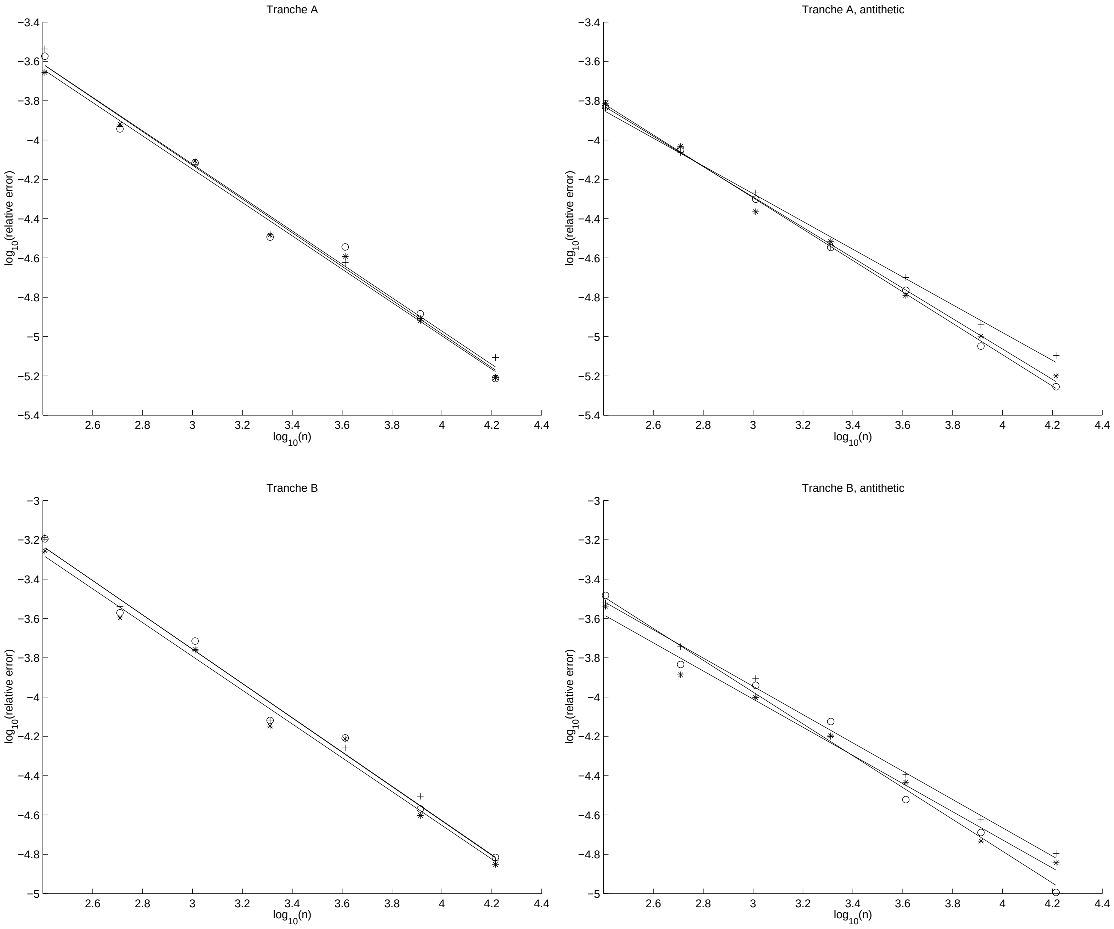


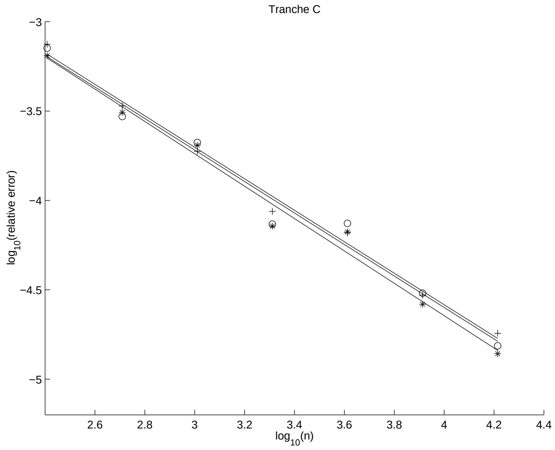
Tranche C, antithetic

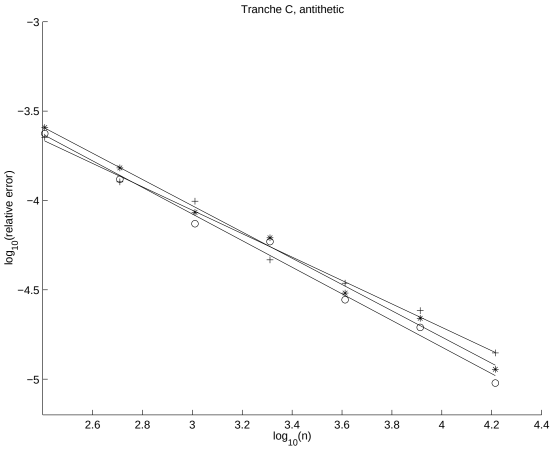

Tranche D
Tranche D, antithetic
IO
IO, antithetic
PO
PO, antithetic
log$_{10}$(relative error)
log$_{10}$(n)
−2.5
−3
−3.5
−4
−4.5
−5
−2.2
−2.4
−2.6
−2.8
−3.2
−3.4
−3.6
−3.8
−4.2
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4

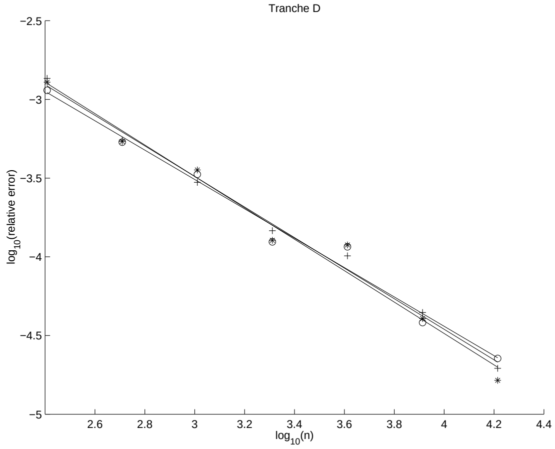


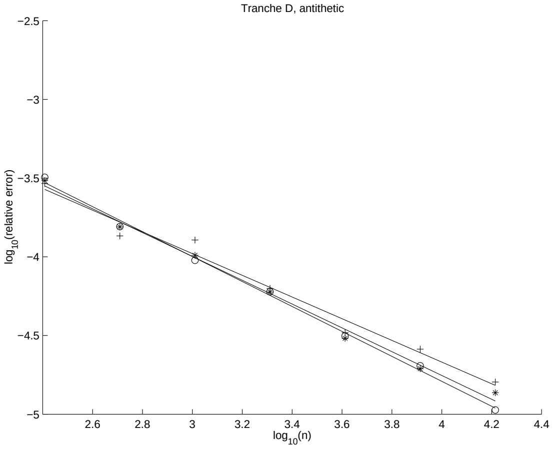
Method	Symbol
Weighted principal component construction with Sobol numbers and $e^{-0.09t}$ weights	$*$
Weighted partial principal component construction ($k = 8$) with Sobol numbers and $e^{-0.09t}$ weights	$+$
Weighted partial principal component construction ($k = 16$) with Sobol numbers and $e^{-0.09t}$ weights	o

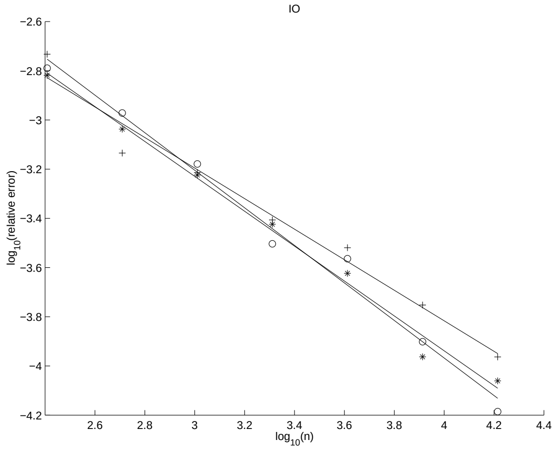


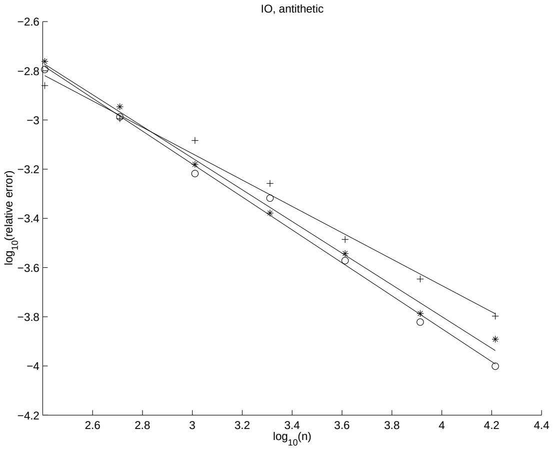

Tranche C
−3
−3.5
−4
−4.5
−5
$\log_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
$\log_{10}$(n)

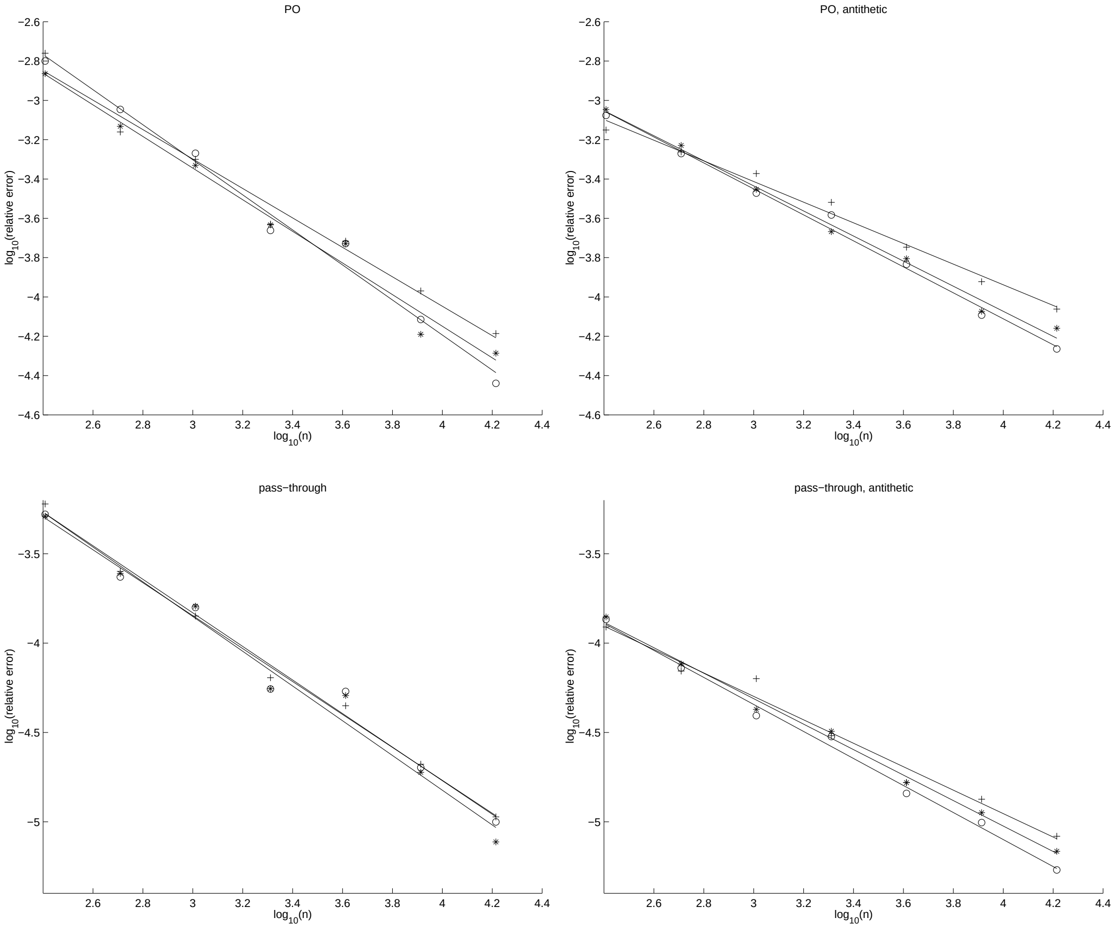

Tranche C, antithetic
−3
−3.5
−4
−4.5
−5
$\log_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
$\log_{10}$(n)


Tranche D
−2.5
−3
−3.5
−4
−4.5
−5
$\log_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
$\log_{10}$(n)


Tranche D, antithetic
−2.5
−3
−3.5
−4
−4.5
−5
$\log_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
$\log_{10}$(n)

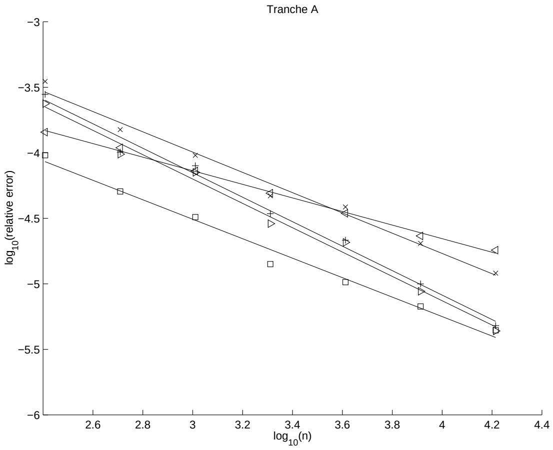

IO
−2.6
−2.8
−3
−3.2
−3.4
−3.6
−3.8
−4
−4.2
$\log_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
$\log_{10}$(n)

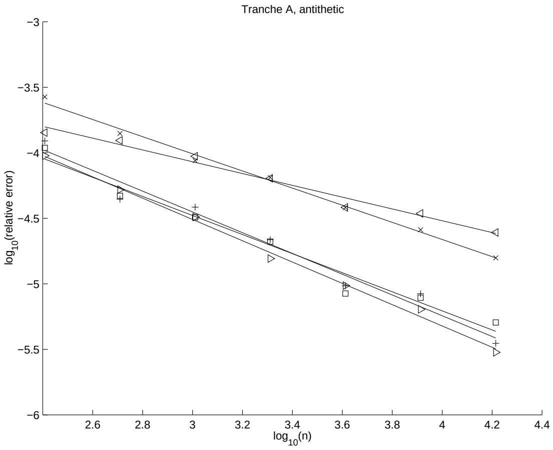

IO, antithetic
−2.6
−2.8
−3
−3.2
−3.4
−3.6
−3.8
−4
−4.2
$\log_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
$\log_{10}$(n)

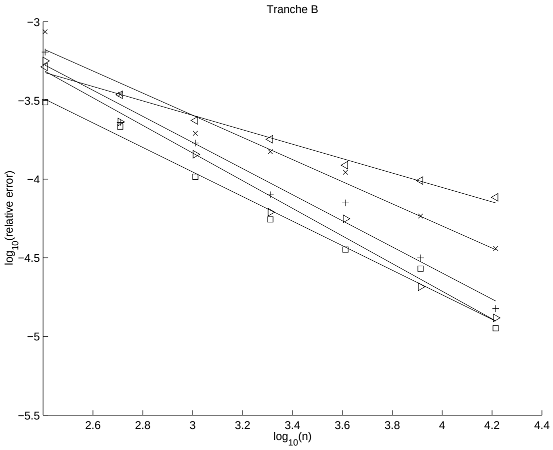


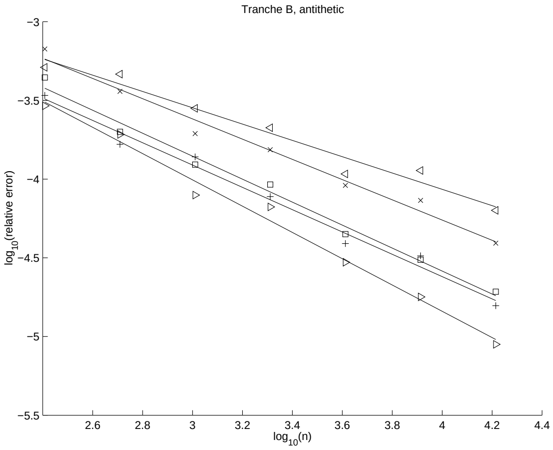
8.3 The quadratic approximation methods

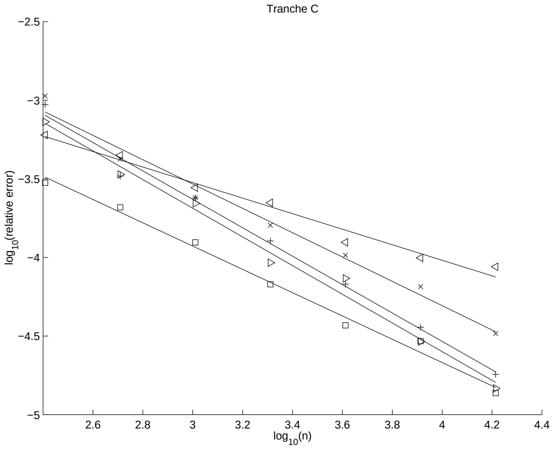
Method	Symbol
Principal component construction with Sobol numbers	×
Weighted principal component construction with Sobol numbers and weights from heuristic	▷
Quadratic approximation with Sobol numbers	+
Importance sampling and stratification based on a log-quadratic approximation using pseudo random numbers	◁
Importance sampling and path generation based on a log-quadratic approximation using Sobol numbers	□

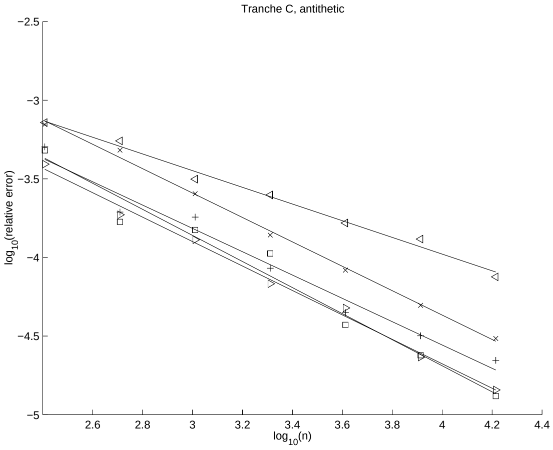

Tranche A
−3
−3.5
−4
−4.5
−5
−5.5
−6
log$_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
log$_{10}$(n)

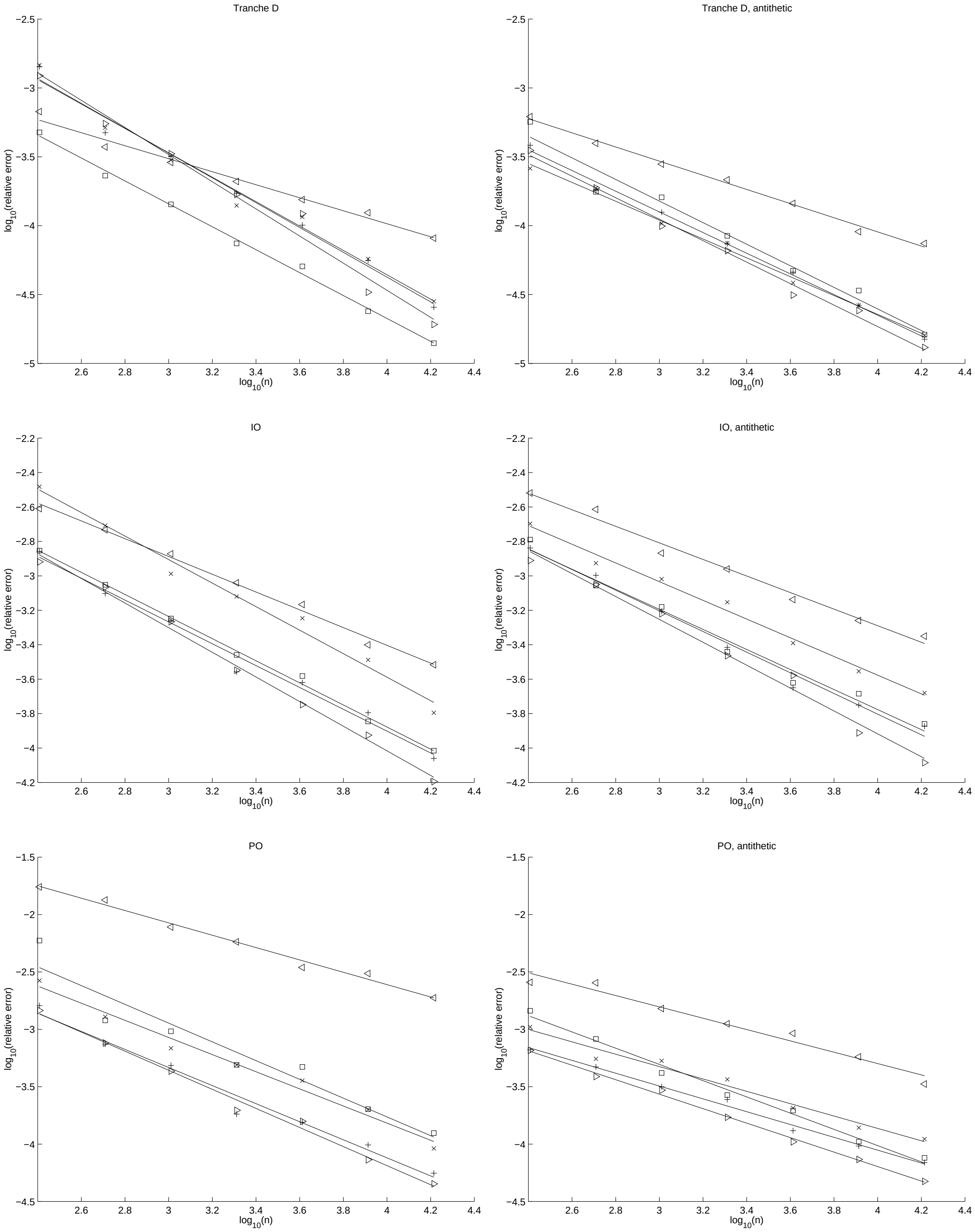

Tranche A, antithetic
−3
−3.5
−4
−4.5
−5
−5.5
−6
log$_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
log$_{10}$(n)

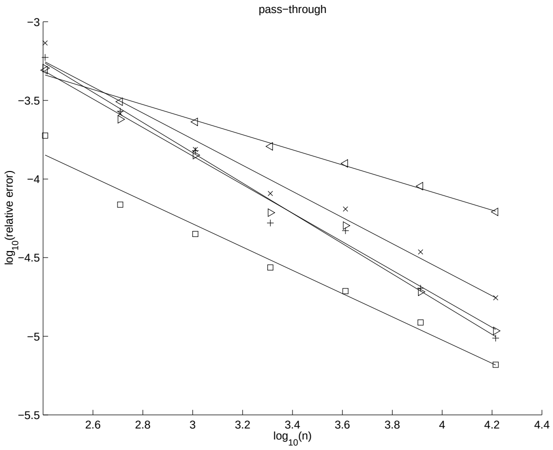

Tranche B
−3
−3.5
−4
−4.5
−5
−5.5
log$_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
log$_{10}$(n)

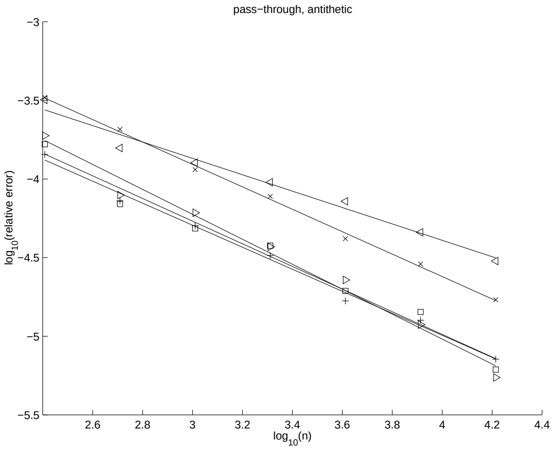

Tranche B, antithetic
−3
−3.5
−4
−4.5
−5
−5.5
log$_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
log$_{10}$(n)


Tranche C
−2.5
−3
−3.5
−4
−4.5
−5
log$_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
log$_{10}$(n)


Tranche C, antithetic
−2.5
−3
−3.5
−4
−4.5
−5
log$_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
log$_{10}$(n)

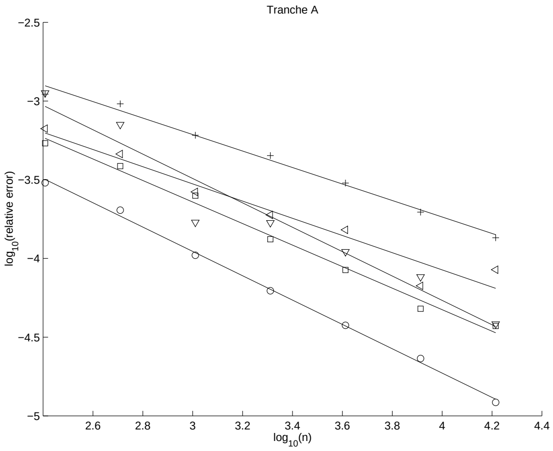

Tranche D
Tranche D, antithetic
IO
IO, antithetic
PO
PO, antithetic
log$_{10}$(relative error)
log$_{10}$(n)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
−2.5
−3
−3.5
−4
−4.5
−5
−2.2
−2.4
−2.6
−2.8
−3
−3.2
−3.4
−3.6
−3.8
−4
−4.2
−1.5
−2
−2.5
−3
−3.5
−4
−4.5

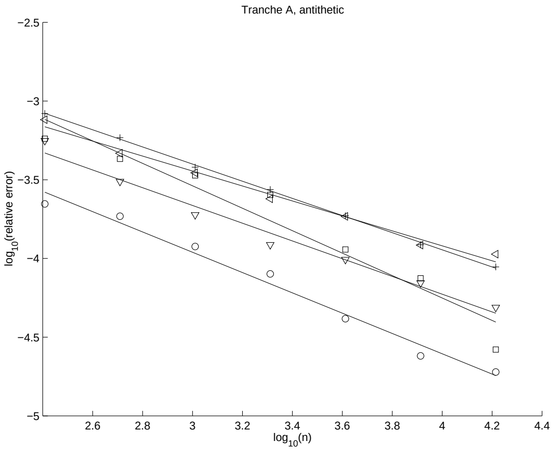


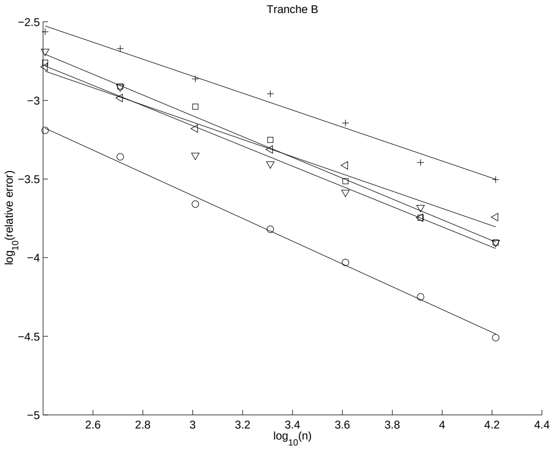


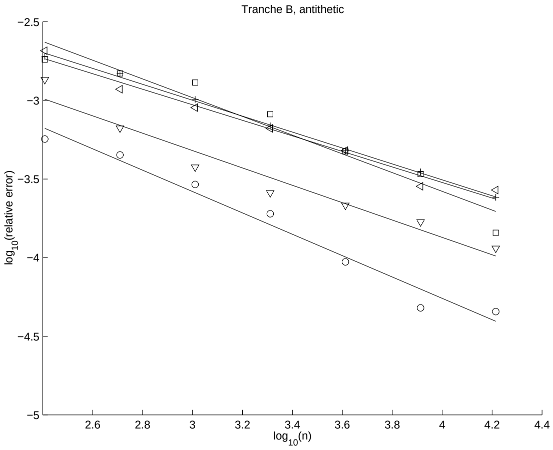
8.4 Generalized Faure

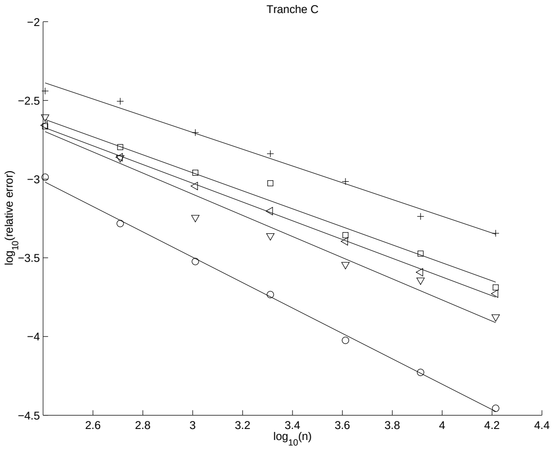
Method	Symbol
Standard path generation with pseudo random numbers	+
Standard path generation with Sobol numbers	□
Standard path generation with GFaure numbers	◁
Brownian Bridge construction with Sobol numbers	o
Brownian Bridge construction with GFaure numbers	▽

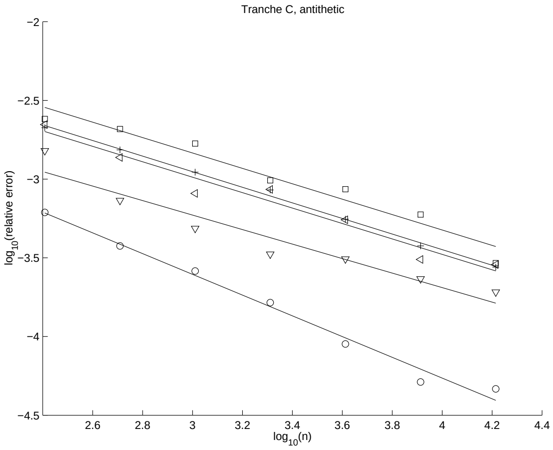


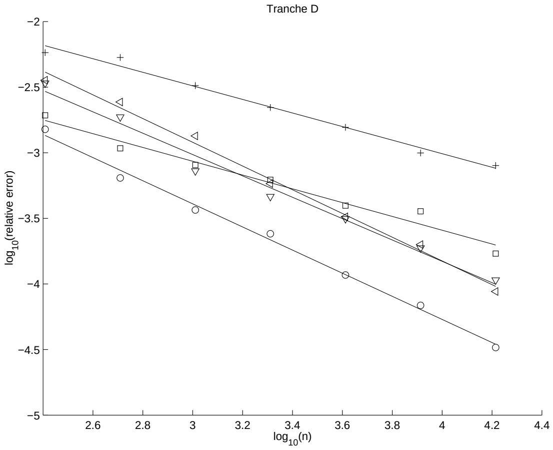


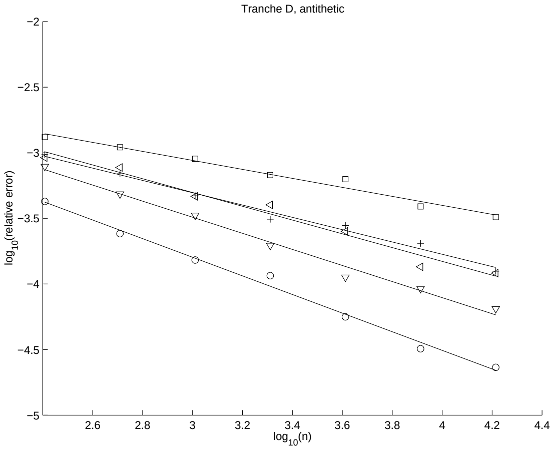

Tranche B
−2.5
−3
−3.5
−4
−4.5
−5
$\log_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
$\log_{10}$(n)

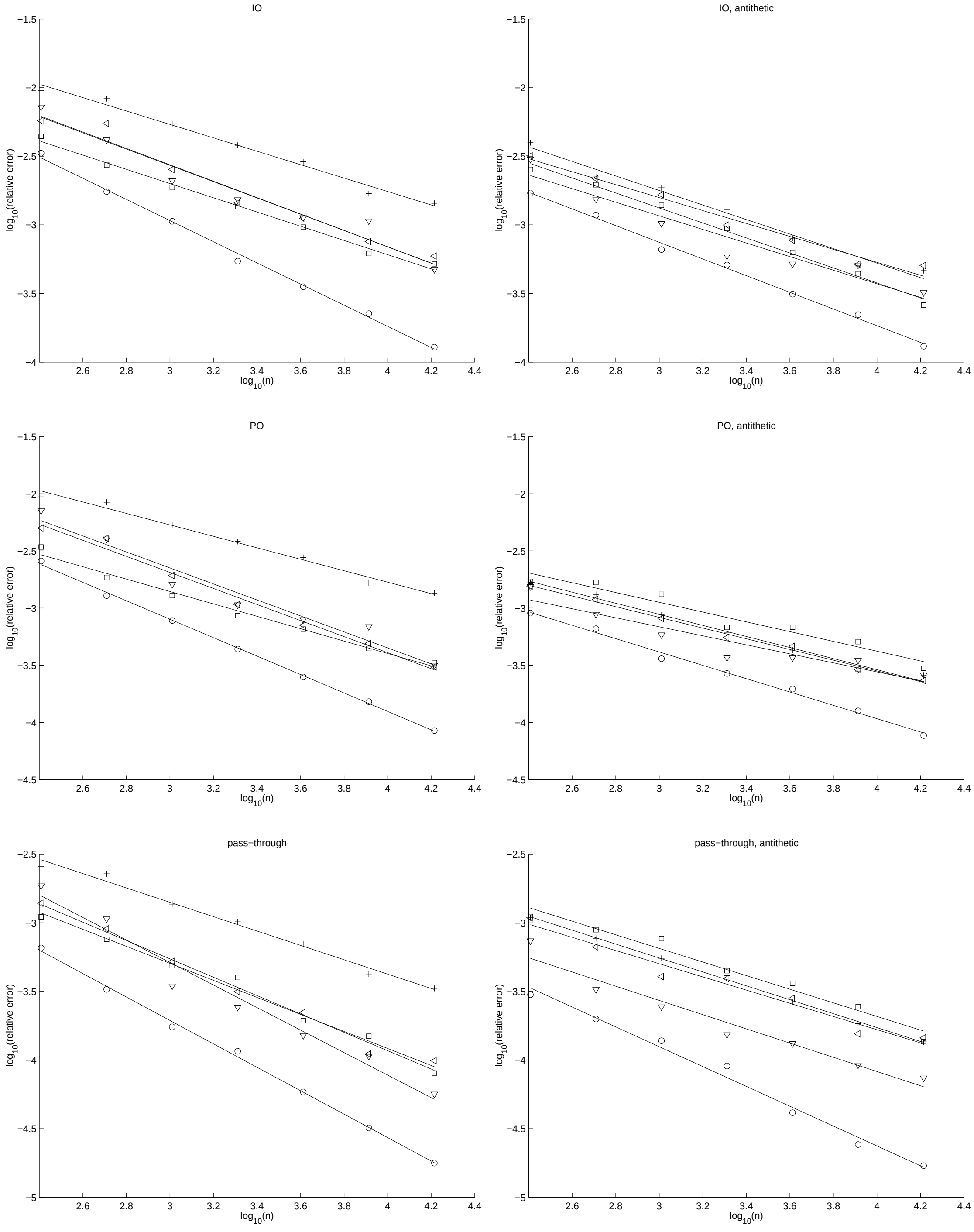

Tranche B, antithetic
−2.5
−3
−3.5
−4
−4.5
−5
$\log_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
$\log_{10}$(n)


Tranche C
−2
−2.5
−3
−3.5
−4
−4.5
$\log_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
$\log_{10}$(n)

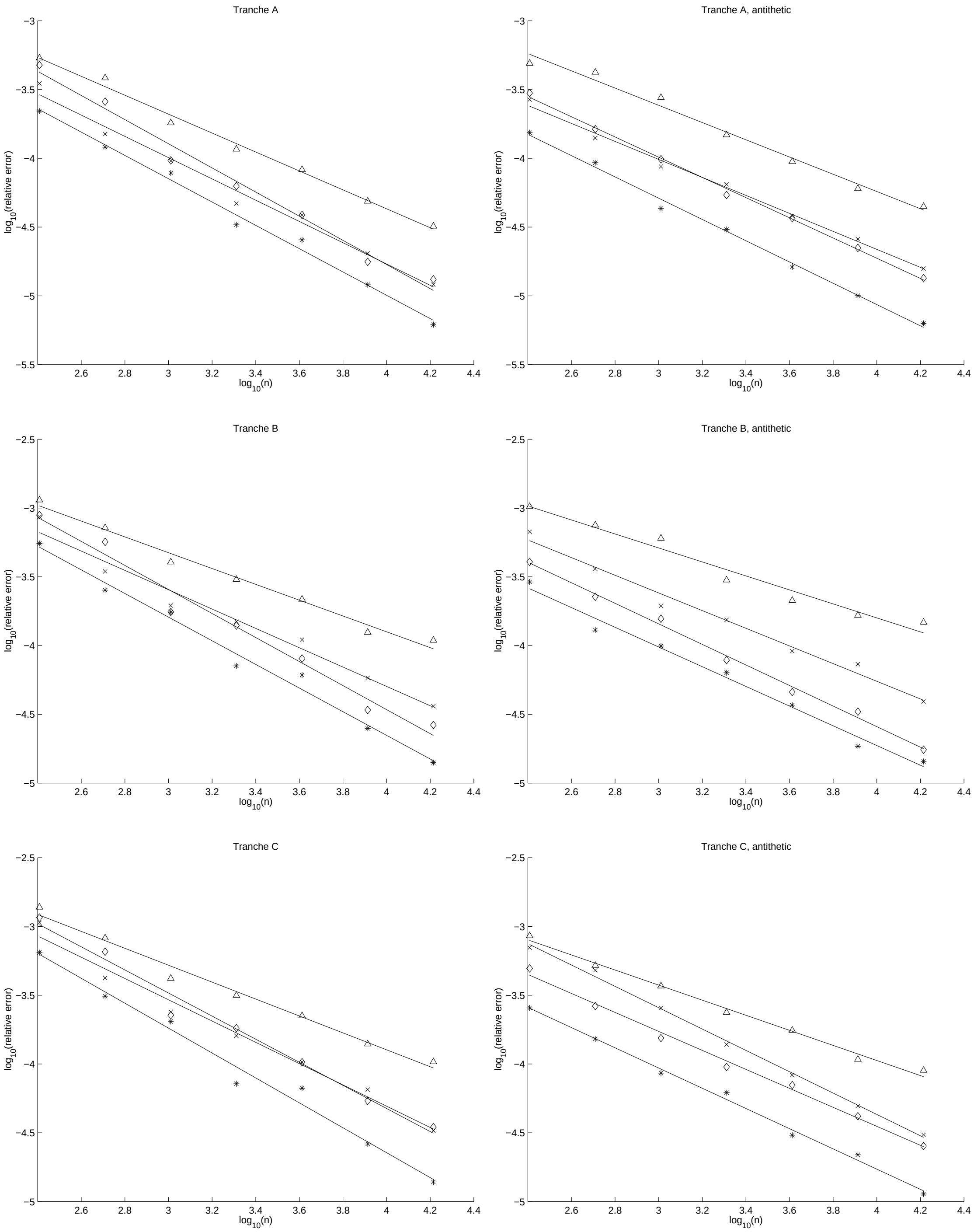

Tranche C, antithetic
−2
−2.5
−3
−3.5
−4
−4.5
$\log_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
$\log_{10}$(n)

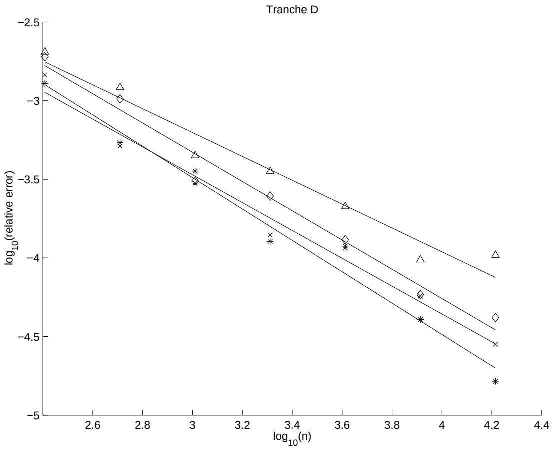

Tranche D
−2
−2.5
−3
−3.5
−4
−4.5
−5
$\log_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
$\log_{10}$(n)

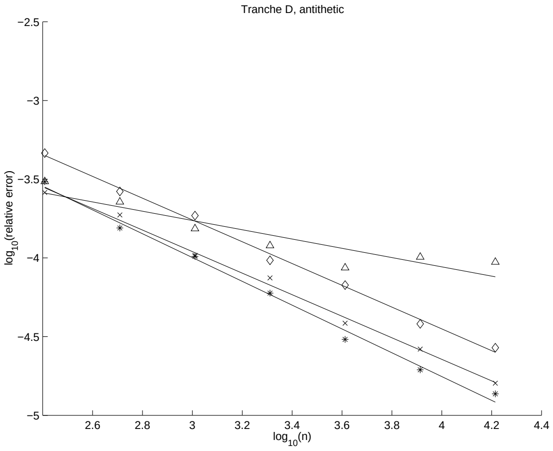

Tranche D, antithetic
−2
−2.5
−3
−3.5
−4
−4.5
−5
$\log_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
$\log_{10}$(n)

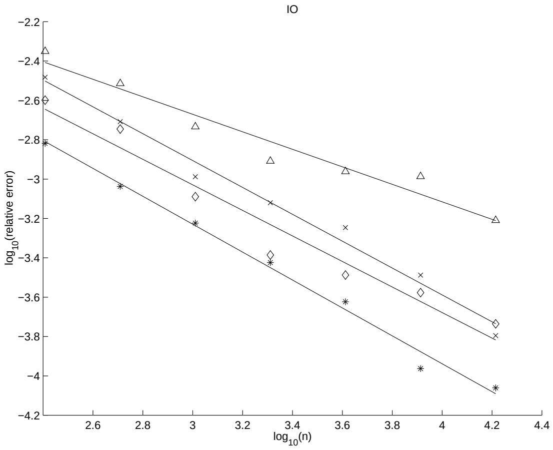


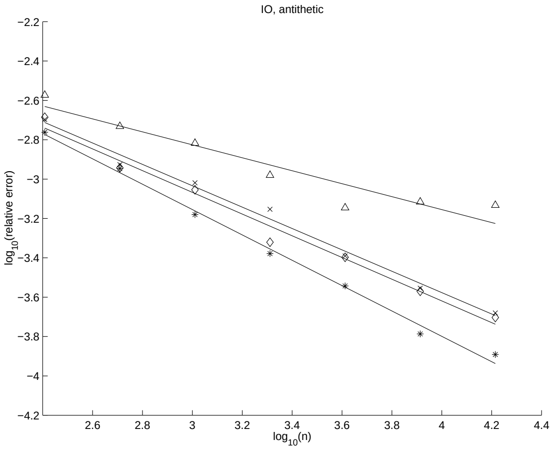
Method	Symbol
Principal component construction with Sobol numbers	×
Principal component construction with GFaure numbers	△
Weighted principal component construction with Sobol numbers and $e^{-0.09t}$ weights	∗
Weighted principal component construction with GFaure numbers and $e^{-0.09t}$ weights	⋄

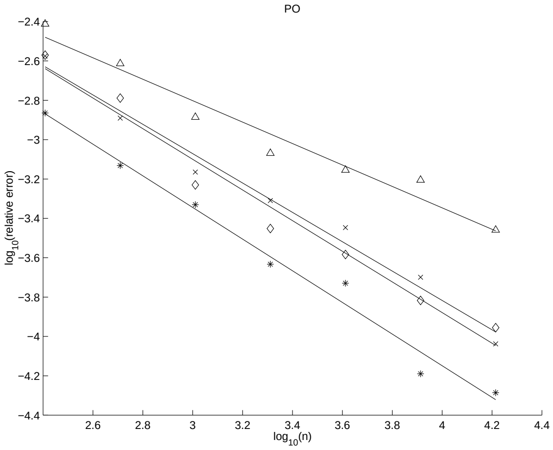

Tranche A
Tranche A, antithetic
Tranche B
Tranche B, antithetic
Tranche C
Tranche C, antithetic
$\log_{10}$(relative error)
$\log_{10}(n)$
−3
−3.5
−4
−4.5
−5
−5.5
−2.5
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4

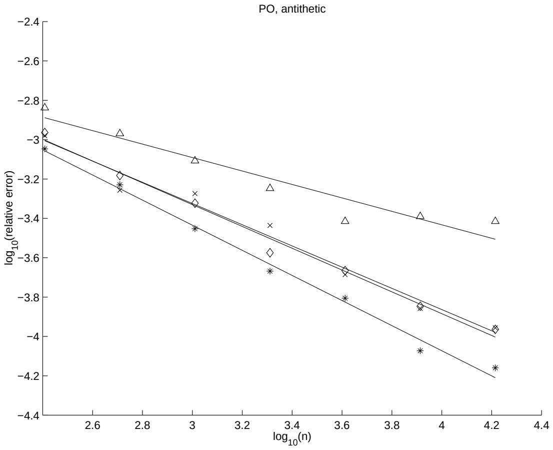

Tranche D

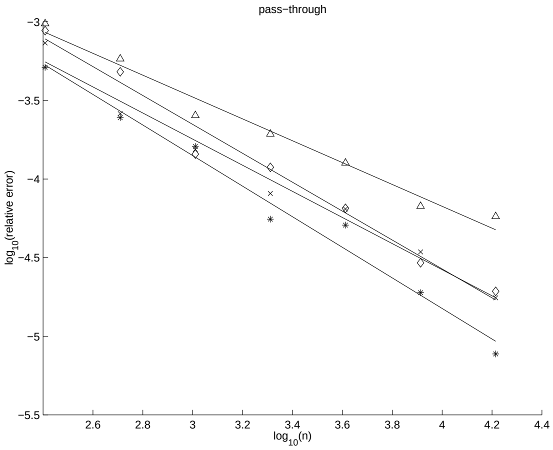

Tranche D, antithetic

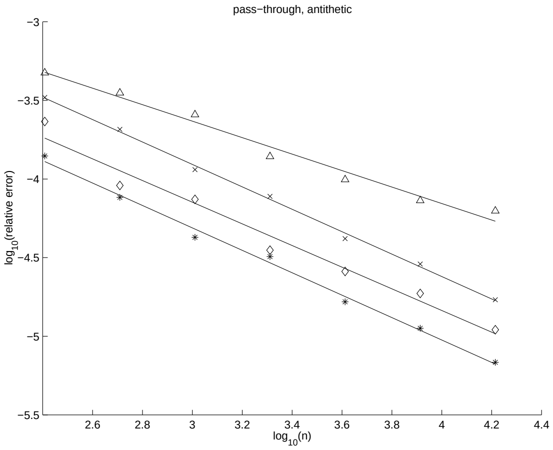

IO


IO, antithetic


PO


PO, antithetic


pass−through
−3
−3.5
−4
−4.5
−5
−5.5
log$_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
log$_{10}$(n)


pass−through, antithetic
−3
−3.5
−4
−4.5
−5
−5.5
log$_{10}$(relative error)
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
log$_{10}$(n)