# FAST ORTHOGONAL TRANSFORMS FOR PRICING DERIVATIVES
# WITH QUASI-MONTE CARLO

Christian Irrgeher
Gunther Leobacher

University of Linz (JKU)
Altenbergerstraße 69
A-4040 Linz, AUSTRIA

## ABSTRACT

There are a number of situations where, when computing prices of financial derivatives using quasi-Monte Carlo (QMC), it turns out to be beneficial to apply an orthogonal transform to the standard normal input variables. Sometimes those transforms can be computed in time $O(n \log(n))$ for problems depending on $n$ input variables. Among those are classical methods like the Brownian bridge construction and principal component analysis (PCA) construction for Brownian paths.

Building on preliminary work by Imai and Tan (2007) as well as Wang and Sloan (2011), where the authors try to find optimal orthogonal transform for given problems, we present how those transforms can be approximated by others that are fast to compute. We further present a new regression-based method for finding a Householder reflection which turns out to be very efficient for a wide range of problems. We apply these methods to several very high-dimensional examples from finance.

## 1 INTRODUCTION

Many simulation problems in finance and other applied fields can be written in the form $E(f(X))$, where $f$ is a measurable function on $\mathbb{R}^n$ and $X$ is a standard normal vector, that is, $X = (X_1, \ldots, X_n)$ is jointly normal with $E(X_j) = 0$ and $E(X_j X_k) = \delta_{jk}$. It is a trivial observation of surprisingly big consequences that

$$E(f(X)) = E(f(UX)) \tag{1}$$

for every orthogonal transform $U : \mathbb{R}^n \longrightarrow \mathbb{R}^n$. While this reformulation does not change the simulation problem from the probabilistic point of view, it does sometimes make a big difference when quasi-Monte Carlo simulation is applied to generate the realizations of $X$.

Examples are supplied by the well-known Brownian bridge and PCA constructions of Brownian paths which will be detailed in the following paragraphs. Assume that one wants to know $E(g(B))$ where $B$ is a Brownian motion with index set $[0, T]$. In most applications this can be reasonably approximated by $E(\tilde{g}(B_{\frac{T}{n}}, \ldots, B_{\frac{Tn}{n}}))$, where $\tilde{g}$ is a function on the set of discrete Brownian paths.

There are three classical methods for sampling from $(B_{\frac{T}{n}}, \ldots, B_{\frac{nT}{n}})$ given a standard normal vector $X$, namely the forward method, the Brownian bridge construction and the principal component analysis construction (PCA). All of these constructions may be written in the form $(B_{\frac{T}{n}}, \ldots, B_{\frac{nT}{n}}) = AX$, where $A$ is an $n \times n$ real matrix with

$$AA^\top = \Sigma := \left( \frac{T}{n} \min(j,k) \right)_{j,k=1}^n = \frac{T}{n} \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & 2 & 2 & \ldots & 2 \\ 1 & 2 & 3 & \ldots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & 3 & \ldots & n \end{pmatrix}.$$

For example, the matrix $A$ corresponding to the forward method is

$$A = S := \sqrt{\frac{T}{n}} \begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix}, \tag{2}$$

while PCA corresponds to $A = VD$, where $\Sigma = VD^2V^\top$ is the singular value decomposition of $\Sigma$. A decomposition for the Brownian bridge algorithm is given, for example, by Larcher, Leobacher, and Scheicher (2003).

It has been observed by Papageorgiou (2002) that $AA^\top = \Sigma$ if and only if $A = SU$ for some orthogonal matrix $U$, so that every linear construction of $(B_{\frac{T}{n}}, \dots, B_{\frac{nT}{n}})$ corresponds to an orthogonal transform of $\mathbb{R}^n$. In that sense the forward method corresponds to the identity, PCA corresponds to $S^{-1}VD$ and Brownian bridge corresponds to the inverse Haar transform, see Leobacher (2012).

Thus our original simulation problem can be written, as

$$\mathrm{E}(\tilde{g}(B_{\frac{T}{n}}, \dots, B_{\frac{Tn}{n}})) = \mathrm{E}(\tilde{g}(SX)) = \mathrm{E}(f(X))$$

with $f = \tilde{g} \circ S$, and we interpret this as using the forward method. Consequently, the same problem using the Brownian bridge takes on the form $\mathrm{E}(f(H^{-1}X))$, where $H$ is the matrix of the inverse Haar transform, and has the form $\mathrm{E}(f(S^{-1}VDX))$, with $S$, $V$, $D$ as above, when PCA is used.

As an application one can generalize the classical constructions of discrete Brownian paths to discrete Lévy paths. See Leobacher (2006), L'Ecuyer, Parent-Chartier, and Dion (2008), Imai and Tan (2009).

There are some theories as to why an orthogonal transform might have the effect to make the problem more suitable for QMC. Caflisch, Morokoff, and Owen (1997) introduce the concept of effective dimension of a function: consider a function $g : \mathbb{R}^n \longrightarrow \mathbb{R}$ with finite variance w.r.t. normal distribution, that is $\mathrm{E}(g(X)^2) < \infty$ where $X = (X_1, \dots, X_n)$ is a vector of independent standard normal random variables. Then $g$ may be written uniquely as the sum of functions $g_u : \mathbb{R}^n \to \mathbb{R}$, $u \subseteq \{1, \dots, n\}$, where $g_u$ depends on the $i$-th coordinate only if $i \in u$ and where $\mathrm{E}(g_u(X)) = 0$ for all $u \neq \emptyset$ and $\mathrm{E}(g_u(X)g_v(X)) = 0$ for $u \neq v$, using the so-called ANOVA decomposition of $g$. Furthermore it holds

$$\mathrm{V}(g(X)) = \sum_{\emptyset \neq u \subseteq \{1, \dots, n\}} \mathrm{V}(g_u(X)).$$

The effective dimension in the truncation sense at level $\alpha \in (0, 1)$ is then the smallest integer $k$ such that

$$\mathrm{V}(g(X))(1 - \alpha) < \sum_{\emptyset \neq u \subseteq \{1, \dots, k\}} \mathrm{V}(g_u(X)),$$

see Caflisch, Morokoff, and Owen (1997). Typically $\alpha$ is chosen as 0.01. Therefore, a function with effective dimension $k$ is one that, in this sense, almost exclusively depends on the first $k$ variables and which therefore is more suitable for QMC. This is confirmed by empirical evidence. Building on the concept of effective dimension of a function, Owen (2012) gives definitions of effective dimensions of function spaces, thus connecting the concepts of effective dimension with that of tractability.

Now one can turn this around and try to put as much variance as possible to the first few coordinates, by concatenating $g$ with a suitable orthogonal transform. This is what has been done by Imai and Tan (2007) and what we will do here, using a different approach. We shall see in Section 4 that empirical evidence also supports the conjectured efficiency of our method.

However, there is also a disadvantage of that approach: the computation of the orthogonal transform has a cost, which is in general of the order $O(n^2)$. For large $n$ this cost is likely to swallow the potential

gains from the transform. We therefore concentrate on orthogonal transforms which have cost of the order $O(n\log(n))$.

Examples include discrete sine and cosine transform, Walsh and Haar transform as well as the orthogonal matrix corresponding to the PCA, see Scheicher (2007), Leobacher (2012).

Imai and Tan (2007) propose an algorithm to find a good orthogonal transform in the sense that it puts as much variance as possible to the first few dimensions. They propose to take the first order Taylor expansion at some point $\widetilde{X}$, i.e.

$$g(X) \approx g(\widetilde{X}) + \sum_{i=1}^{n} \frac{\partial g(X)}{\partial X_i}\big|_{X=\widetilde{X}}(X_i - \widetilde{X}_i).$$

Then the contribution of the $i$-th component of $X$ to $\mathrm{V}(g(X))$ is given by $(\frac{\partial g(X)}{\partial X_i}\big|_{X=\widetilde{X}})^2$. The columns of the orthogonal transform are chosen by solving optimization problems of the form

$$A^*_{\cdot i} = \max_{A_{\cdot i} \in \mathbb{R}^n} \left( \frac{\partial g(AX)}{\partial X_i}\big|_{X=\widetilde{X}_i} \right)^2$$

$$\text{with } ||A_{\cdot i}|| = 1 \text{ and } A^\top_{\cdot j}A_{\cdot i} = 0, \ j = 1,\ldots,i-1$$

with $\widetilde{X}_i = (\widetilde{X}_i^1,\ldots,\widetilde{X}_i^i,0,\ldots,0)$. They suggest to perform this optimization only for the first few columns of the matrix $A$. In this paper we improve on their algorithm in various directions. In particular we find a good orthogonal transform that is fast in that it can be computed even in linear time.

The remainder of the paper is organized as follows. Section 2 reviews basic properties of Householder reflections and shows how they can be used to find fast versions of orthogonal transforms which put most variance on the first $k$ variables. The main part of our article, Section 3, describes algorithms for finding fast orthogonal transforms using again Householder reflections. In contrast to the method of Imai and Tan (2007) we do not rely on differentiability. This makes the algorithm useful for barrier-type options. We further provide some theoretical results which indicate why the method serves to reduce the effective dimension.

Section 4 gives some numerical examples where the methods described earlier are applied to examples from finance. We will see that the new methods described in Section 3 are among the best, both with regard to speed and accuracy.

We provide an appendix where we compute certain expectations depending on the maximum of a Brownian path. This is useful for some of the numerical examples.

## 2 HOUSEHOLDER REFLECTIONS

We recall the definition and basic properties of Householder reflections from Golub and Van Loan (1996).

**Definition 1** A matrix of the form

$$U = I - 2\frac{vv^\top}{v^\top v},$$

where $v \in \mathbb{R}^n$, is called a *Householder reflection*. The vector $v$ is called the defining *Householder vector*.

In the following proposition, $e_1$ denotes the first canonical basis vector in $\mathbb{R}^n$, $e_1 = (1,0,\ldots,0)$.

**Proposition 1** A Householder reflection have the following properties:

1. If $x \in \mathbb{R}^n$ is a vector then $Ux$ is the reflection of $x$ in the hyperplane $\mathrm{span}\{v\}^\perp$. In particular, $U$ is orthogonal and symmetric, i.e. $U^{-1} = U$.
2. Given any vector $a \in \mathbb{R}^n$ we can find $v \in \mathbb{R}^n$ such that for the corresponding Householder reflection $U$ we have $Ua = ||a||e_1$. The computation of the Householder vector uses $3n$ floating point operations.

3. The computation of the matrix-vector multiplication $Ux$ uses at most $4n$ floating point operations.

*Proof.* See Chapter 5.1 of Golub and Van Loan (1996). □

Our main application of Householder reflections is the following: suppose we know that for a given integration problem $E(f(X))$ some orthogonal transform $\hat{U}$ reduces the effective dimension in the truncation sense to $k$, that is, almost all of the variance of $f(\hat{U}X)$ is captured by $X_1, \ldots, X_k$, $k \ll n$.

Let $\hat{U} = (\hat{u}_1, \ldots, \hat{u}_n)$, that is, $\hat{u}_j$ is the $j$-th column of $\hat{U}$. We can find Householder reflections $U_1, \ldots, U_k$ such that $U_1 \ldots U_k e_\ell = \hat{u}_\ell$, $\ell = 1, \ldots, k$ as follows:

- Let $U_1$ be a Householder reflection that maps $e_1$ to $\hat{u}_1$. $U_1$ also maps $\hat{u}_1$ to $e_1$. Since the vectors $\hat{u}_j$ are orthogonal we have $e_1^\top(U_1\hat{u}_2) = (U_1\hat{u}_1)^\top(U_1\hat{u}_2) = \hat{u}_1^\top\hat{u}_2 = 0$.
- Therefore there exists a Householder reflection $U_2$ operating on the last $n-1$ coordinates which maps $e_2$ to $U_1\hat{u}_2$. Thus $U_1U_2e_1 = U_1e_1 = \hat{u}_1$, $U_1U_2e_2 = U_1U_1\hat{u}_2 = \hat{u}_2$.
- Suppose Householder reflections $U_1, \ldots, U_j$ have been constructed such that $U_1 \ldots U_j e_\ell = \hat{u}_\ell$, $\ell = 1, \ldots, j$.
- Then there exists a Householder reflection $U_{j+1}$ operating on the last $n-j$ coordinates which maps $e_{j+1}$ to $U_j \ldots U_1\hat{u}_{j+1}$. Then $U_1 \ldots U_{j+1}e_\ell = \hat{u}_\ell$, $\ell = 1, \ldots, j+1$.

Write $U = U_1 \ldots U_k$. By construction the first $k$ columns of $U$ coincide with those of $\hat{U}$. Since, by assumption, $X_1, \ldots, X_k$ capture almost all of the variance of $f(\hat{U}X)$, the same is true for $f(UX)$. But for small $k$ the computational cost for computing $UX$ is of the order $nk$, as compared to general matrix-vector multiplication which occurs a cost of order $O(n^2)$.

Imai and Tan (2007) and Wang and Sloan (2011) give examples for which they find good orthogonal transforms $\hat{U}$ that reduce the effective dimension. However they do not specify how those transforms are applied. We propose to approximate them using the above method.

However, the main topic of this paper is to present transforms that use only one Householder reflection. This will by detailed in the next section.

## 3  REGRESSION ALGORITHM

Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be a measurable function with $E(f(X)^2) < \infty$ for a standard normal vector $X$.

We want to approximate $f$ by a linear function:

$$f(x) \approx a^\top x + b$$

where $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$. This can be done in different ways. For example, Imai and Tan (2007) take the first order Taylor expansion of $f$.

In contrast, we take a "linear regression" approach, i.e. we minimize

$$E\left((f(X) - a^\top X - b)^2\right) \to \min. \tag{3}$$

First order conditions give

$$a_j = E(f(X)X_j), \quad j = 1, \ldots, n \quad \text{and} \quad b = E(f(X)).$$

Therefore, (3) minimizes the variance of the difference between $f$ and the linear approximation. So

$$\begin{aligned}
V(f(X)) &= E\left((f(X) - b)^2\right) \\
&= E\left((a^\top X)^2 + (f(X) - b - a^\top X)^2\right) \\
&= \|a\|^2 + V\left(f(X) - a^\top X\right).
\end{aligned}$$

That is, $\|a\|^2/V(f(X))$ measures the proportion of variance captured by the linear approximation. Now there exists a unique Householder reflection $U$ that maps $e_1$ to $a/\|a\|$. With this transform we have $a^\top U X = \|a\| e_1^\top X = \|a\| X_1$ and therefore

$$\mathrm{E}(f(X)) = \mathrm{E}(f(UX)) = \mathrm{E}\left(a^\top UX + \left(f(UX) - a^\top UX\right)\right) = \mathrm{E}\left(\|a\| X_1 + (f(UX) - \|a\| X_1)\right).$$

Therefore the linear part of the integration problem depends on the parameter $X_1$ alone. Now, if the linear part constitutes a large part of the integration problem then we have succeeded in putting a large fraction of the variance into the first coordinate by composing $f$ with $U$.

**Algorithm 2** Let $X_1, \ldots, X_n$ be independent standard normal variables. Let $f$ be a function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$.

1. $a_j := \mathrm{E}(X_j f(X))$ for $j = 1, \ldots, n$;
2. if $\|a\| = 0$ define $U = I$ and go to 4.;
3. else let $U$ be a Householder reflection that maps $e_1$ to $a/\|a\|$;
4. Compute $\mathrm{E}(f(UX))$ using QMC.

A drawback of the algorithm is that in general the computation of the expectations in step 1 is no easier than the original problem. In some cases the expectation can be computed explicitly, though usually in that case also the original problem has an explicit solution.

**Example 3** $f(X) = \sum_{k=1}^m w_k \exp\left(\sum_{j=1}^n (c_{k,j} X_j + d_{k,j})\right)$. It is easily verified that, with $\phi$ denoting the standard normal density, $\phi(x) = \exp(-\frac{x^2}{2})/\sqrt{2\pi}$,

$$\int_{\mathbb{R}} \exp(cx+d)\phi(x)dx = \exp\left(c^2/2+d\right), \qquad \int_{\mathbb{R}} x\exp(cx+d)\phi(x)dx = c\exp\left(c^2/2+d\right).$$

Therefore it holds that

$$a_i = \mathrm{E}(f(X)X_i) = \int_{-\infty}^{\infty} f(x)x_i\phi(x_1)\ldots\phi(x_n)dx_1\ldots dx_n = \sum_{k=1}^m c_{k,i} w_k \exp\left(\sum_{j=1}^n \left(c_{k,j}^2/2 + d_{k,j}\right)\right).$$

Let us find out how much of the variance of $f(UX)$ is captured by $\|a\| X_1$:

We write $\bar{w}_k := w_k \exp(\sum_{j=1}^n (c_{k,j}^2/2 + d_{k,j}))$. Then

$$\|a\|^2 = \sum_{i=1}^n \left(\sum_{k=1}^m \bar{w}_k c_{k,i}\right)^2 = \sum_{k_1=1}^m \sum_{k_2=1}^m \bar{w}_{k_1} \bar{w}_{k_2} \sum_{i=1}^n c_{k_1,i} c_{k_2,i} = \sum_{k_1=1}^m \sum_{k_2=1}^m \bar{w}_{k_1} \bar{w}_{k_2} \bar{c}_{k_1,k_2}, \qquad (4)$$

where $\bar{c}_{k_1,k_2} := \sum_{i=1}^n c_{k_1,i} c_{k_2,i}$.

On the other hand, it is easy to see that $\mathrm{E}(f(X)) = \sum_{k=1}^m \bar{w}_k$ and $\mathrm{E}(f(X)^2) = \sum_{k_1=1}^m \sum_{k_2=1}^m \bar{w}_{k_1} \bar{w}_{k_2} e^{\bar{c}_{k_1,k_2}}$. Therefore we get for the variance of $f(UX)$

$$V(f(UX)) = V(f(X)) = \mathrm{E}(f(X)^2) - \mathrm{E}(f(X))^2$$

$$= \sum_{k_1=1}^m \sum_{k_2=1}^m \bar{w}_{k_1} \bar{w}_{k_2}\left(e^{\bar{c}_{k_1,k_2}} - 1\right). \qquad (5)$$

Let us try some special values that are related to Asian options:

$$m = n, \; w_k = \frac{1}{n}, \; c_{k,j} = \sigma\sqrt{\Delta t}\,1_{j\le k}, \; d_{k,j} = \left(r - \frac{\sigma^2}{2}\right)\Delta t\,1_{j\le k}$$

with $r, \sigma, T > 0$, $\Delta t = \frac{T}{n}$. For this choice we get $\bar{w}_k = \frac{1}{n}e^{rTk/n}$, and $\bar{c}_{k_1,k_2} = \sigma^2 T\frac{\min(k_1,k_2)}{n}$.

For large $n$ the sums in equations (4) and (5) can be approximated by corresponding integrals such that

$$\|a\|^2 \approx \sigma^2 T \int_0^1 \int_0^1 e^{rTx} e^{rTy} \min(x,y) dx dy$$

$$= \sigma^2 \frac{4e^{rT} + 2e^{2rT} rT - (3e^{2rT} + 1)}{2r^3 T^2}$$

$$V(f(X))^2 \approx \int_0^1 \int_0^1 e^{rTx} e^{rTy} (e^{\sigma^2 T \min(x,y)} - 1) dx dy$$

$$= \frac{4e^{rT} r\sigma^2 + 2e^{T(2r+\sigma^2)} r^2 + 2e^{rT} \sigma^4 - (2e^{2rT} r^2 + r\sigma^2 + 3e^{2rT} r\sigma^2 + \sigma^4 + e^{2rT} \sigma^4)}{r^2 T^2 (r+\sigma^2)(2r+\sigma^2)}$$

Table 1 shows the fraction $\frac{V(f(X)) - \|a\|^2}{V(f(X))}$ for a few values of $r$, $\sigma$ and $T = 1$.

It can be concluded that in this example almost all of the variance of $f(UX)$ is captured by $X_1$. □

Table 1: $\frac{V(f(X)) - \|a\|^2}{V(f(X))}$ for $T = 1$ and different values for $r$, $\sigma^2$.

| $r \backslash \sigma^2$ | 0.01 | 0.02 | 0.03 | 0.04 |
|---|---|---|---|---|
| 0.1 | 0.0025 | 0.0051 | 0.0076 | 0.0101 |
| 0.2 | 0.0026 | 0.0051 | 0.0077 | 0.0103 |
| 0.3 | 0.0026 | 0.0052 | 0.0078 | 0.0104 |

In general we cannot expect that $E(f(X)X_i)$ can be computed explicitly. Of course it is an option to compute $E(f(X)X_i)$ using (quasi-)Monte Carlo, though it is unlikely that this will lead to small overall computing times. But quite frequently, especially in financial applications, a problem can be written in the form, $f(X) = g(h(X))$, where $E(h(X)X_i)$ can be computed and $h$ is some relatively simple function $h : \mathbb{R} \longrightarrow \mathbb{R}$.

**Algorithm 4** Let $X_1, \ldots, X_n$ be independent standard normal variables. Let $f$ be a function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, which is of the form $f = g \circ h$ where $h : \mathbb{R}^n \longrightarrow \mathbb{R}$ and $g : \mathbb{R} \longrightarrow \mathbb{R}$.

1. $a_j := E(X_j h(X))$ for $j = 1, \ldots, n$;
2. if $\|a\| = 0$ define $U = I$ and go to 4.;
3. else let $U$ be a Householder reflection that maps $e_1$ to $a/\|a\|$;
4. Compute $E(f(UX))$ using QMC.

Without additional assumptions on the functions $h$ and $g$ there is no guarantee that $U$ gives better convergence. Nevertheless there are practical examples where this algorithm gives excellent results.

**Example 5** Consider an arithmetic average value option written on some underlying $S$,

$$f(X) = e^{-rT} \max \left( \frac{1}{n} \sum_{k=1}^n S_{\frac{k}{n}T}(X) - K, 0 \right),$$

and

$$S_{\frac{k}{n}T}(X) = S_0 \exp \left( \sum_{j=1}^k \sigma \sqrt{\frac{T}{n}} X_j + \left( r - \frac{\sigma^2}{2} \right) \frac{k}{n} T \right).$$

Here we have $f(X) = g(h(X))$, where $g(s) = e^{-rT} \max(s - K, 0)$ and $h$ is like in Example 3 with $m = n$, $w_k = \frac{1}{n} S_0$, $c_{k,j} = \sqrt{\frac{T}{n}} \sigma 1_{j \leq k}$, $d_{k,j} = \frac{T}{n} (r - \frac{\sigma^2}{2}) 1_{j \leq k}$. □

Write $Y := \|a\|X_1$, $Z := h(UX) - \|a\|X_1$. Then $Y, Z$ are uncorrelated,

$$\begin{aligned}
E(YZ) &= E(h(UX)\|a\|e_1^\top X) - \|a\|^2 = E(h(UX)\|a\|(Ue_1)^\top UX) - \|a\|^2 \\
&= E(h(UX)a^\top UX) - \|a\|^2 = a^\top E(h(UX)UX) - \|a\|^2 \\
&= a^\top E(h(X)X) - \|a\|^2 = a^\top a - \|a\|^2 = 0.
\end{aligned}$$

Further, $E(Y) = 0$, such that $E(Y)E(Z) = 0$ as well, and therefore $Cov(Y,Z) = 0$.

**Theorem 6** Let $f, g, h, U, X_1, \ldots, X_n$ be like in Algorithm 4. Write again $Y := \|a\|X_1$, $Z := h(UX) - \|a\|X_1$. Then $V(f(UX)) = V(E(g(Y+Z)|Y)) + V(g(Y+Z) - E(g(Y+Z)|Y))$.

*Proof.* We write $\bar{Y} = E(g(Y+Z)|Y)$ and $\bar{Z} = g(Y+Z) - E(g(Y+Z)|Y)$, so that we have to show $V(\bar{Y} + \bar{Z}) = V(\bar{Y}) + V(\bar{Z})$. To that end it is sufficient to prove that $\bar{Y}$ and $\bar{Z}$ are uncorrelated:

$$\begin{aligned}
E(\bar{Y}\bar{Z}) &= E(E(g(Y+Z)|Y)g(Y+Z)) - E(E(g(Y+Z)|Y)E(g(Y+Z)|Y)) \\
&= E(E(E(g(Y+Z)|Y)g(Y+Z)|Y)) - E(E(g(Y+Z)|Y)^2) \\
&= E(E(g(Y+Z)|Y)E(g(Y+Z)|Y)) - E(E(g(Y+Z)|Y)^2) = 0.
\end{aligned}$$

Since $E(\bar{Z}) = 0$, we have $E(\bar{Y})E(\bar{Z}) = 0 = E(\bar{Y}\bar{Z})$. □

We consider a special case that will rarely occur in practice but which gives a flavor of the best result possible. Assume that $g$ is Lipschitz continuous with constant $L$. Suppose further that $Y$ and $Z$ are not only uncorrelated, but even independent.

Denote by $F_Y$, $F_Z$ the cumulative probability distribution functions of $Y$ and $Z$, respectively. Using independence we get

$$E(g(Y+Z)|Y) = \int_{\mathbb{R}} g(Y+\zeta)dF_Z(\zeta).$$

Noting that $E(g(Y+Z) - E(g(Y+Z)|Y)) = 0$ we thus get

$$\begin{aligned}
V(g(Y+Z) - E(g(Y+Z)|Y)) &= E\left((g(Y+Z) - E(g(Y+Z)|Y))^2\right) \\
&= E\left(\left(\int_{\mathbb{R}}(g(Y+Z) - g(Y+\zeta))dF_Z(\zeta)\right)^2\right) \\
&\leq E\left(\int_{\mathbb{R}}(g(Y+Z) - g(Y+\zeta))^2 dF_Z(\zeta)\right) \\
&\leq E\left(L^2 \int_{\mathbb{R}}(Z-\zeta)^2 dF_Z(\zeta)\right) \\
&\leq L^2 E\left(Z^2 - 2ZE(Z) + E(Z^2)\right) = 2L^2 V(Z),
\end{aligned}$$

where we also have used the Cauchy-Schwarz inequality. Thus with Theorem 6 we get

$$V(f(UX)) - V(E(g(Y+Z)|Y)) \leq 2L^2 V(Z)$$

that is,

$$V(f(UX)) - V(E(f(UX)|X_1)) \leq 2L^2(V(h(UX)) - \|a\|^2).$$

So in this situation, if $X_1$ captures a large fraction of the variance of $h(UX)$, then $X_1$ also captures a large fraction of the variance of $f(UX)$ provided that the Lipschitz constant $L$ is not too big.

We can also think of a variant of Algorithm 4 for slightly more complicated functions. We have been inspired by Wang and Sloan (2011), where the authors consider functions of the form $f(X) = g(w_1^\top X, \ldots, w_m^\top X)$ and show, that there is an orthogonal transform that makes this function $m$-dimensional. We give a slightly modified version of their argument which guarantees that the orthogonal transform is also fast to compute for small $m$, that is for $m \leq \log(n)$.

Let $f(X) = g(w_1^\top X, \ldots, w_m^\top X)$ for $w_1, \ldots, w_m \in \mathbb{R}^n$. We may assume that $w_1$ is not the zero vector. Let $U_1$ be a Householder reflection which maps $(1, 0, \ldots, 0)$ to $\frac{w_1}{\|w_1\|}$. Then $w_1^\top U_1 X = \|w_1\|(1, 0, \ldots, 0)^\top X = \|w_1\| X_1$ and therefore $f(U_1 X) = g(\|w_1\| X_1, (U_1 w_2)^\top X, \ldots, (U_1 w_m)^\top X)$. Next we write $(U_1 w_k)^\top X = (U_1 w_k)_1^\top X_1 + (U_1 w_k)_{2\ldots n}^\top X_{2\ldots n}$. That is,

$$f(U_1 X) = \bar{g}(X_1, \bar{w}_2^\top X_{2\ldots n}, \ldots, \bar{w}_m^\top X_{2\ldots n})$$

for some $\bar{w}_2, \ldots, \bar{w}_m \in \mathbb{R}^{n-1}$. Assuming that $\bar{w}_2 \neq 0$, let $\bar{U}_2$ be the Householder reflection of $\mathbb{R}^{n-1}$ that maps $(1, 0, \ldots, 0)$ to $\bar{w}_2 / \|\bar{w}_2\|$ and let

$$U_2 = \begin{pmatrix} 1 & 0 \\ 0 & \bar{U}_2 \end{pmatrix}.$$

Then $U_2$ is a Householder reflection of $\mathbb{R}^n$ and

$$f(U_1 U_2 X) = \bar{\bar{g}}(X_1, X_2, \bar{\bar{w}}_3^\top X_{3\ldots n}, \ldots, \bar{\bar{w}}_m^\top X_{3\ldots n}).$$

for some $\bar{\bar{w}}_3, \ldots, \bar{\bar{w}}_n \in \mathbb{R}^{n-2}$. Proceeding that way one arrives at

$$f(U_1 \cdots U_{\hat{m}} X) = \hat{g}(X_1, X_2, \ldots, X_{\hat{m}})$$

for some $\hat{m} \leq m$ (We may have $\hat{m} < m$ if at some stage all remaining $w_k$ are zero).

We propose a similar procedure for an integration problem of the form $f(X) = g(h_1(X), h_2(X), \ldots, h_m(X))$ where $\mathrm{E}(h_j(X) X_k)$ can be computed explicitly (or at least efficiently).

**Algorithm 7** Let $X_1, \ldots, X_n$ be independent standard normal variables. Let $f$ be a function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, which is of the form $f = g \circ h$ where $h : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ and $g : \mathbb{R}^m \longrightarrow \mathbb{R}$.

1. Start with $k = 1, \ell = 1$ and $U = I$;
2. $a_j^{(k)} := \mathrm{E}(X_j h_k(UX))$ for $j = k, \ldots, n$;
3. $a_j^{(k)} := 0$ for $j = 1, \ldots, k-1$;
4. if $\|a^{(k)}\| = 0$ go to 7;
5. else let $U^{(\ell)}$ be a Householder reflection that maps $e_\ell$ to $a^{(k)} / \|a^{(k)}\|$;
6. $U = UU^{(\ell)}$; $\ell = \ell + 1$;
7. $k = k + 1$;
8. while $k \leq m$, go back to 2;
9. Compute $\mathrm{E}(f(UX))$ using QMC.

We will give a numerical example in Section 4.

## 4   NUMERICAL TESTS

In this section we will apply our method to examples from mathematical finance.

### Asian option

The first numerical example we give is the evaluation of an Asian call option with discrete arithmetic average in the Black-Scholes model, which has been discussed previously. Since the payoff function $f$ is

of the form $g \circ h$ with $g$ and $h$ as in Example 5, we apply Algorithm 4 to the integration problem $E(f(X))$ where the vector $a$ follows from Example 3, i.e. for every $i = 1, \ldots, n$

$$a_i = \frac{1}{n} \sum_{k=i}^{n} \sigma \sqrt{\frac{T}{n}} \, e^{rkT/n}.$$

For the quasi-Monte Carlo simulation we use a Sobol sequence of dimension $n = 250$ with a random shift and we have $S_0 = 100, K = 100, r = 0.04, \sigma = 0.2$ as well as $T = 1$. We compute the standard deviation based on 32 batches for $N$ sample paths, where the number of sample paths ranges from $2^1$ to $2^{14}$. Note that the standard deviation is different from the RQMC standard deviation defined in (L'Ecuyer and Munger 2012).

In Figure 1 we compare the regression method with the forward method, the PCA construction and the LT method of Imai and Tan. We see that PCA, LT and the Regression method yield similar results, but all of the three outperform the forward method. Note that the regression method can be applied in $O(n)$. Thus we can achieve the efficiency of the PCA with the regression method with lower computational costs. Moreover, it is interesting that the LT method and regression method yield nearly the same results.
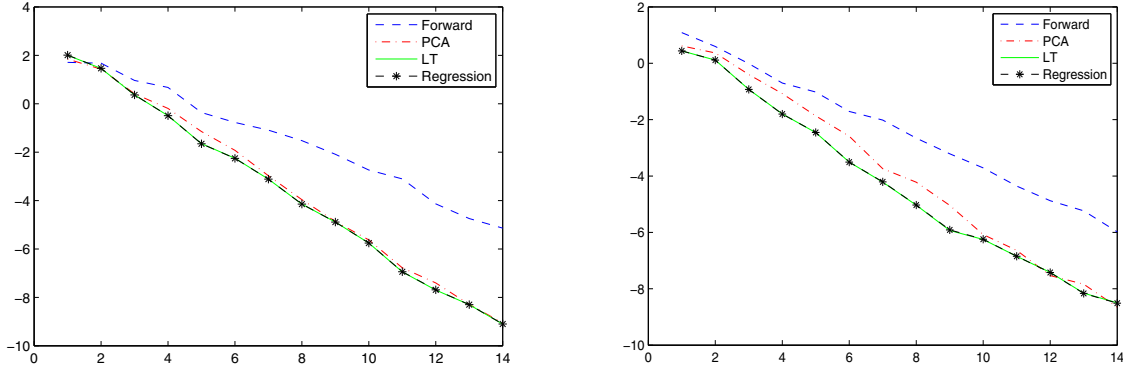


Figure 1: Asian option (left) and Asian Basket option (right): Standard deviation of 32 runs on a $\log_2$-scale.

The computation time required to price the Asian option using quasi-Monte Carlo integration with $2^{14}$ paths is given in Table 2. Note that PCA is implemented using the discrete sine transform as discussed in Leobacher (2012). The LT method is implemented such that only the first 25 columns are optimized and then the orthogonal transform is completed using Householder reflections as we suggested in Section 2.

Table 2: Computation times for pricing the Asian option.

|            | Forward | PCA  | LT   | Regression |
|------------|---------|------|------|------------|
| time (sec) | 0.08    | 0.64 | 1.94 | 0.15       |

Furthermore it should be mentioned that the regression method as well as the LT method produce an overhead caused by determining the orthogonal transform. Nevertheless the overhead time is rather small and is negligible for a large sample size.

The computation times of the subsequent numerical examples are similar to the result regarding the Asian option.

**Asian basket option**

We consider an Asian basket call option with arithmetic average and a basket consisting of $m$ assets, an example taken from Imai and Tan (2007). The $i$-th asset $S^{(i)}$ of the basket ($i = 1, \ldots, m$) is given by

$$S^{(i)}_{k\frac{T}{n}} = S^{(i)}_0 \exp\left(\left(r - \frac{\sigma_i^2}{2}\right) k\frac{T}{n} + \sigma_i B^{(i)}_{k\frac{T}{n}}\right)$$

where $S^{(i)}_0$ is the current price of the $i$-th asset, $r$ is the risk-free interest rate, $\sigma_i$ is the volatility of the $i$-th asset and $B = (B^{(1)}, \ldots, B^{(m)})$ is an $m$-dimensional Brownian motion. The correlation between $B^{(j)}$ and $B^{(k)}$ is denoted by $\rho_{jk}$. The payoff function of the Asian basket option is given by

$$f(X) = \max\left(\frac{1}{nm} \sum_{i=1}^{m} \sum_{k=1}^{n} S^{(i)}_{k\frac{T}{n}}(X) - K, 0\right),$$

where

$$S^{(i)}_{k\frac{T}{n}}(X) = S^{(i)}_0 \exp\left(\sum_{k=1}^{nm} C_{(k-1)m+i,k} X_k + (r - \frac{\sigma_i^2}{2}) k\frac{T}{n}\right),$$

and where $C$ is an $mn \times mn-$matrix with $CC^\top = \tilde{\Sigma} := R \otimes \Sigma$ and $R$ is an $m \times m-$matrix with $R_{ii} = \sqrt{T/n}\,\sigma_i^2$ for all $i$ and $R_{ij} = \sqrt{T/n}\,\rho_{ij}\sigma_i\sigma_j$ for $i \neq j$. Note that the discussion of the previous sections also holds for a discrete Brownian path with covariance matrix $\tilde{\Sigma}$. Since the problem is of the form $f(X) = g(h(X))$, Algorithm 4 can be applied. Since the function $h$ is of the form considered in Example 3, we can compute the corresponding vector $a$ analytically. Furthermore, notice that the PCA construction can be computed in this example efficiently by using the orthogonal transform $V_1 D_1 \otimes V_2 D_2$ where $V_1 D_1^2 V_1^\top = R$ and $V_2 D_2^2 V_2^\top = \Sigma$.

The parameters are $T = 1, r = 0.04, K = 100$ and $\rho_{jk} = 0.05$ for $j \neq k$. Moreover, the volatility of the 10 assets is equally spaced from 0.1 to 0.3 and we assume that $S^{(i)}_0 = 100$ for all $i = 1, \ldots, m$. Since we simulate every asset at 250 time points, we take a Sobol sequence in dimension $n = 2500$ with a random shift. In Figure 1 we can observe the standard deviation based on 32 batches of the forward method, the PCA construction, the LT method and the regression method for $N$ sample paths with $N$ up to $2^{14}$.

**Digital barrier option**

A digital (up-and-in) barrier option is a derivative which pays 1 if the underlying asset breaks through a barrier $u$ on the time interval $[0, T]$ and pays 0 otherwise. We intend to price the option in a discrete Black-Scholes model, where the path of the stock is given by $S = (S_1, \ldots, S_n)$ with

$$S_k(X) = S_0 \exp\left(\left(r - \frac{\sigma^2}{2}\right) k\frac{T}{n} + \sigma B_{k\frac{T}{n}}\right) \tag{6}$$

with current stock price $S_0$, interest rate $r$, volatility $\sigma$, Brownian path $B = (B_{k\frac{T}{n}})_{k=1}^{n}$ where $B_{k\frac{T}{n}} = \sqrt{\frac{T}{n}} \sum_{j=1}^{k} X_j$ and standard normal vector $X = (X_1, \ldots, X_n)$. Hence, the payoff function $h$ of the digital barrier option is

$$h(X) = 1_{\max_{k=1,\ldots,n} S_k(X) \geq u}$$

which leads us to an integration problem of the form $\mathrm{E}(\exp(-rT)h(B))$. We can use Algorithm 4 for solving this problem and therefore, we have to compute $a_i = \mathrm{E}(h(X)X_i)$ for $i = 1, \ldots, n$. In the appendix we

show how to calculate this expectation for a function depending on the maximum of a Brownian motion with drift $v$. We can adjust our problem by

$$\max_{k=1,\ldots,n} S_k \geq u \Longleftrightarrow \max_{k=1,\ldots,n} S_0 \exp\left(\left(r - \frac{\sigma^2}{2}\right) k\frac{T}{n} + \sigma B_{k\frac{T}{n}}\right) \geq u$$

$$\Longleftrightarrow \max_{k=1,\ldots,n} \frac{(r - \frac{\sigma^2}{2})}{\sigma} k\frac{T}{n} + B_{k\frac{T}{n}} \geq \frac{\log(u/S_0)}{\sigma}$$

$$\Longleftrightarrow \max_{k=1,\ldots,n} B^v_{k\frac{T}{n}} \geq \tilde{u}$$

with $B^v_t = vt + B_t$, $v = \frac{(r-\frac{\sigma^2}{2})}{\sigma}$ and $\tilde{u} = \frac{\log(u/S_0)}{\sigma}$. With (7) we get that the vector $a$ in Algorithm 4 can be approximated by

$$a \approx S^{-1}\beta - v\sqrt{T/n}\,\gamma\mathbf{1}$$

where $S$ is given by (2), $\beta = (\beta_1, \ldots, \beta_n)^\top$ with $\beta_i = \mathrm{E}(1_{\max_{0 \leq s \leq T} B^v_s \geq u} B^v_{i\frac{T}{n}})$, $\gamma = \mathrm{E}(1_{\max_{0 \leq s \leq T} B^v_s \geq u})$ and $\mathbf{1} = (1,\ldots,1)^\top$. The computation of $\beta_i$ with $i = 1,\ldots,n$ can be reduced to a 1-dimensional integration problem using (10) with $f = \mathrm{id}_\mathbb{R}$ and $t = i\frac{T}{n}$ and formula (8) with $f \equiv 1$ simplifies $\gamma$. Consequently, we end up with 1-dimensional integrals which can be evaluated efficiently with an adaptive quadrature rule.

For the numerical test we use a Sobol sequence of dimension $n = 2000$ with a random shift and the parameter set is chosen as $S_0 = 100, u = 110, r = 0.04, \sigma = 0.2$ and $T = 1$. The number of sample paths $N$ ranges from $2^1$ to $2^{14}$ and we compute the standard deviation for those $N$ based on 32 batches. Since it is not clear how to apply the LT method of Imai and Tan to barrier options, we compare the regression method with the forward method and the PCA construction only. In Figure 2 we can observe that the difference between the forward method and the PCA is smaller than in the previous examples. Furthermore, we see that the regression method is slightly behind the PCA, but this seems to be the best we can achieve by linear approximation.
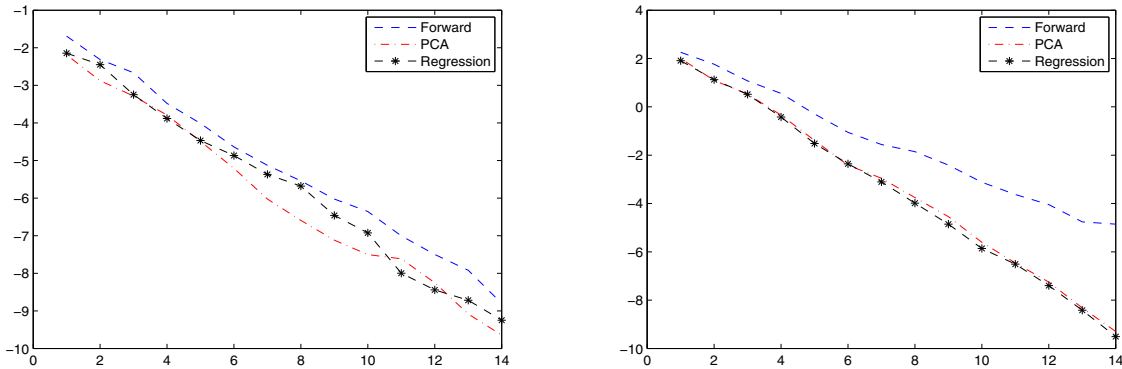


Figure 2: Digital barrier option (left) and Asian barrier option (right): Standard deviation of 32 runs on a $\log_2$-scale.

## Asian barrier option

The last example we provide is an Asian (up-and-in) barrier option by which we mean that the payoff of the option is similar to an Asian option as in the first numerical example, but is paid only if the underlying

asset breaks through an upper barrier $u$. The corresponding function is then given by

$$f(X) = \exp(-rT)1_{\max_{k=1,\ldots,n} S_k(X) \geq u} \max\left(\sum_{k=1}^{n} \frac{1}{n} S_k(X) - K, 0\right)$$

where $S_k(X)$ is as in (6) for $k = 1, \ldots, n$. Since the function $f$ is of the form $f(X) = g(h_1(X), h_2(X))$ with $g(x,y) = \exp(-rT)xy$, $h_1(X) = 1_{\max_{k=1,\ldots,n} S_k(X) \geq u}$ and $h_2(X) = \max(\sum_{k=1}^{n} \frac{1}{n} S_k(X) - K, 0)$, we apply Algorithm 7 with $m = 2$ to the problem. The computation of the vectors $a^{(1)}$ and $a^{(2)}$ is already discussed in the examples above, i.e. $a^{(1)}$ is related to the digital barrier option and $a^{(2)}$ corresponds to the Asian option.

The numerical test is based on 32 batches and we again compare the standard deviation of the forward method, the PCA construction and the regression method for various numbers of sample paths $N$, ranging from $2^1$ to $2^{14}$. Moreover, we use a Sobol sequence in dimension $n = 1000$ with a random shift and the parameters are $S_0 = 100, K = 100, u = 110, r = 0.04, \sigma = 0.2$ and $T = 1$. Figure 2 shows that the regression method yields slightly better results than the PCA and that the forward method is behind the other two approaches.

## APPENDIX: REGRESSION FOR THE MAXIMUM

We give the computations needed for examples of barrier type, that is we want to compute $E(h(X)X_i)$ where $h$ is some function of the maximum of a discrete Brownian path with drift $v$, i.e.

$$h(X) = \tilde{h}\left(\max_k\left(B_{\frac{kT}{n}} + v\frac{kT}{n}\right)\right),$$

and where $B_{\frac{kT}{n}} = \sqrt{\frac{T}{n}} \sum_{j=1}^{k} X_j$. We make the approximation

$$E\left(\tilde{h}\left(\max_k\left(B_{\frac{kT}{n}} + v\frac{kT}{n}\right)\right) X_i\right) \approx E\left(\tilde{h}\left(\max_{0 \leq s \leq T}(B_s + vs)\right)\sqrt{n}\left(B_{\frac{iT}{n}} - B_{\frac{(i-1)T}{n}}\right)\right)$$

$$= E\left(\tilde{h}\left(\max_{0 \leq s \leq T} B_s^v\right)\sqrt{n}\left(B_{\frac{iT}{n}}^v - B_{\frac{(i-1)T}{n}}^v - v\frac{T}{n}\right)\right), \qquad (7)$$

where $B^v$ denotes Brownian motion with drift $v \in \mathbb{R}$, i.e. $B_t^v := B_t + vt$, $t \geq 0$. Moreover, let $M_{t,T}^v := \max_{t \leq s \leq T} B_s^v$ and $M_t^v := M_{0,t}^v$. At first we compute $E(1_{M_T^v \geq u} f(B_t^v))$ for given $u > 0$ and measurable $f$ with $E(|f(B_t^v)|) < \infty$. Then we show how the expectation for more general $\tilde{h}$ can be computed using the first result.

We start with a simple calculation for a Brownian motion $B$ with drift 0 and let $M_t := M_t^0$. For $u \geq 0$ we get, using the reflection principle for Brownian motion,

$$E(1_{M_t \geq u} f(B_t)) = E(1_{M_t \geq u} 1_{B_t \geq u} f(B_t)) + E(1_{M_t \geq u} 1_{B_t < u} f(B_t))$$
$$= E(1_{B_t \geq u} f(B_t)) + E(1_{B_t \geq u} f(2u - B_t)).$$

Next we make a Girsanov-type change of measure such that under the new measure $Q$ the Brownian motion $B^v$ with drift becomes a standard Brownian motion. So with $\frac{dQ}{dP} = e^{-vB_t - \frac{v^2}{2}t}$, that is $\frac{dP}{dQ} = e^{vB_t^v - \frac{v^2}{2}t}$,

$$E\left(1_{M_t^v \geq u} f(B_t^v)\right) = E_Q\left(1_{M_t^v \geq u} f(B_t^v) e^{vB_t^v - \frac{v^2}{2}t}\right)$$
$$= E_Q\left(1_{B_t^v \geq u} f(B_t^v) e^{vB_t^v - \frac{v^2}{2}t}\right) + E_Q\left(1_{B_t^v \geq u} f(2u - B_t^v) e^{v(2u - B_t^v) - \frac{v^2}{2}t}\right)$$
$$= E\left(1_{B_t^v \geq u} f(B_t^v)\right) + E_Q\left(1_{-B_t^v \geq u} f(2u + B_t^v) e^{v(2u + B_t^v) - \frac{v^2}{2}t}\right)$$
$$= E\left(1_{B_t^v \geq u} f(B_t^v)\right) + e^{2uv} E\left(1_{B_t^v \leq -u} f(2u + B_t^v)\right). \qquad (8)$$

The next step is to consider $\mathrm{E}(1_{M_T^v \geq u}f(B_t^v))$ for $t < T$. Let $\{\mathscr{F}_t\}_{0 \leq t \leq T}$ denote the standard filtration of $B$.

$$
\begin{aligned}
\mathrm{E}\big(1_{M_T^v \geq u}f(B_t^v)\big) &= \mathrm{E}\big(\mathrm{E}(1_{M_T^v \geq u}f(B_t^v)|\mathscr{F}_t)\big) = \mathrm{E}\big(f(B_t^v)\mathrm{E}(1_{M_T^v \geq u}|\mathscr{F}_t)\big) \\
&= \mathrm{E}\big(f(B_t^v)\mathrm{E}(1_{M_t^v \geq u} + 1_{M_t^v < u}1_{M_{t,T}^v \geq u}|\mathscr{F}_t)\big) \\
&= \mathrm{E}\big(f(B_t^v)1_{M_t^v \geq u}\big) + \mathrm{E}\big(1_{M_t^v < u}f(B_t^v)\mathrm{E}(1_{M_{t,T}^v \geq u}|\mathscr{F}_t)\big).
\end{aligned} \tag{9}
$$

We have already computed the first term. For the second term we note that by the Markov property of Brownian motion,

$$
\mathrm{E}(1_{M_{t,T}^v \geq u}|\mathscr{F}_t) = \mathrm{E}(1_{M_{t,T}^v \geq u}|B_t^v) = \mathrm{E}(1_{\max_{t \leq s \leq T}(B_s^v - B_t^v) \geq (u - B_t^v)}|B_t^v).
$$

We can use our earlier result (8) with $f(x) \equiv 1$ to obtain

$$
\mathrm{E}(1_{\max_{t \leq s \leq T}(B_s^v - B_t^v) \geq (u - B_t^v)}|B_t^v) = \Phi\Big(\frac{B_t^v - u - v(T-t)}{\sqrt{T-t}}\Big)(1 + e^{2uv}).
$$

Let us write $g(u,x) := \Phi\big(\frac{x-u-v(T-t)}{\sqrt{T-t}}\big)(1 + e^{2uv})$. Then, using (8) and (9) we obtain

$$
\begin{aligned}
\mathrm{E}(1_{M_T^v \geq u}f(B_t^v)) &= \mathrm{E}(f(B_t^v)1_{M_t^v \geq u}) + \mathrm{E}(1_{M_t^v < u}f(B_t^v)g(u,B_t^v)) \\
&= \mathrm{E}(f(B_t^v)g(u,B_t^v)) + \mathrm{E}(f(B_t^v)1_{M_t^v \geq u}) - \mathrm{E}(1_{M_t^v \geq u}f(B_t^v)g(u,B_t^v)) \\
&= \mathrm{E}(f(B_t^v)g(u,B_t^v)) + \mathrm{E}\big(1_{M_t^v \geq u}f(B_t^v)(1 - g(u,B_t^v))\big) \\
&= \mathrm{E}(f(B_t^v)g(u,B_t^v)) + \mathrm{E}\big(1_{B_t^v \geq u}f(B_t^v)(1 - g(u,B_t^v))\big) \\
&\quad + e^{2uv}\mathrm{E}\big(1_{B_t^v \leq -u}f(2u + B_t^v)(1 - g(u,2u + B_t^v))\big).
\end{aligned} \tag{10}
$$

Note that the expectations can be computed explicitly for suitable $f$.

We can also use (10) to compute $\mathrm{E}(h(M_T^v)f(B_t^v))$ for $h$ differentiable and $h(0) = 0$ and such that the expectations all converge absolutely:

$$
\begin{aligned}
\mathrm{E}(h(M_T^v)f(B_t^v)) &= \mathrm{E}(\mathrm{E}(h(M_T^v)|B_t^v)f(B_t^v)) \\
&= \mathrm{E}\Big(\int_0^\infty h'(u)\mathrm{E}(1_{M_T^v \geq u}|B_t^v)du f(B_t^v)\Big) \\
&= \int_0^\infty h'(u)\mathrm{E}(\mathrm{E}(1_{M_T^v \geq u}|B_t^v)f(B_t^v))du \\
&= \int_0^\infty h'(u)\mathrm{E}(1_{M_T^v \geq u}f(B_t^v))du.
\end{aligned}
$$

**REFERENCES**

Caflisch, R. E., W. Morokoff, and A. Owen. 1997. "Valuation of Mortgage-Backed Securities Using Brownian Bridges to Reduce Effective Dimension". *The Journal of Computational Finance* 1 (1): 27–46.

Golub, G. H., and C. F. Van Loan. 1996. *Matrix Computations*. Third ed. Baltimore: The Johns Hopkins University Press.

Imai, J., and K. S. Tan. 2007. "A general dimension reduction technique for derivative pricing". *J. Comput. Finance* 10:129–155.

Imai, J., and K. S. Tan. 2009. "An accelerating quasi-Monte Carlo method for option pricing under the generalized hyperbolic Lévy process". *SIAM J. Sci. Comput.* 31 (3): 2282–2302.

Larcher, G., G. Leobacher, and K. Scheicher. 2003. "On the tractability of the Brownian Bridge algorithm". *J. Complexity* 19:511–528.

L'Ecuyer, P., and D. Munger. 2012. "On Figures of Merit for Randomly-Shifted Lattice Rules". In *Monte Carlo and Quasi Monte Carlo Methods 2010*, edited by H. Woźniakowski and L. Plaskota. Berlin: Springer. to appear.

L'Ecuyer, P., J.-S. Parent-Chartier, and M. Dion. 2008, December. "Simulation of a Lévy process by PCA sampling to reduce the effective dimension". In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Moench, O. Rose, T. Jefferson, and J. W. Fowler, 436–443. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Leobacher, G. 2006. "Stratified sampling and quasi-Monte Carlo simulation of Lévy processes". *Monte-Carlo methods and applications* 12 (3-4): 231–238.

Leobacher, G. 2012. "Fast orthogonal transforms and generation of Brownian paths". *Journal of Complexity* 28 (2): 278–302.

Owen, A. B. 2012. "Effective dimension for weighted function spaces". Technical report, Department of Statistics, Stanford University.

Papageorgiou, A. 2002. "The Brownian bridge does not offer a consistent advantage in quasi-Monte Carlo integration.". *J. Complexity* 18 (1): 171–186.

Scheicher, K. 2007. "Complexity and effective dimension of discrete Lévy areas.". *J. Complexity* 23 (2): 152–168.

Wang, X., and I. H. Sloan. 2011. "Quasi-Monte Carlo Methods in Financial Engineering: An Equivalence Principle and Dimension Reduction". *Operations Research* 59 (1): 80–95.

## AUTHOR BIOGRAPHIES

**CHRISTIAN IRRGEHER** is PhD student in applied mathematics and research assistant at the Institute of Financial Mathematics at the University of Linz, Austria. He has a Master's degree in Industrial Mathematics from the University of Linz. His main research interests are (quasi-)Monte Carlo simulation and financial engineering. He is supported by the Austrian Science Foundation (FWF), Project S9609. His email address is christian.irrgeher@jku.at.

**GUNTHER LEOBACHER** is assistant professor at the Institute of Financial Mathematics at the University of Linz, Austria. He received his PhD degree in Mathematics from the University of Linz. His research interests include (quasi-)Monte Carlo methods, derivative pricing and stochastic optimal control. He is partially supported by the Austrian Science Foundation (FWF), Project P21196. His email address is gunther.leobacher@jku.at.