Quasi-random sampling with black box or acceptance-rejection inputs

Erik Hintz, Marius Hofert and Christiane Lemieux

Abstract We propose randomized quasi-Monte Carlo (RQMC) methods to estimate expectations $\mu = \mathbb{E}(g(\boldsymbol{Y}, W))$ where $\boldsymbol{Y}$ is independent of W and can be sampled by inversion, whereas W cannot. Various practical problems are of this form, such as estimating expected shortfall for mixture models where W is stable or generalized inverse Gaussian and $\boldsymbol{Y}$ is multivariate normal. We consider two settings: In the first, we assume that there is a non-uniform random variate generation method to sample W in the form of a non-modifiable "black-box". The methods we propose for this setting are based on approximations of the quantile function of W. In the second setting, we assume that there is an acceptance-rejection (AR) algorithm to sample from W and explore different ways to feed it with quasi-random numbers. This has been studied previously, typically by rejecting points of constant dimension from a low-discrepancy sequence and moving along the sequence. We also investigate the use of a point set of constant (target) size where the dimension of each point is increased until acceptance. In addition, we show how to combine the methods from the two settings in such a way that the non-monotonicity inherent to AR is removed.

1 Introduction

Consider the problem of estimating the quantity

Erik Hintz
Department of Statistics and Actuarial Science, University of Waterloo, 200 University Avenue West, Waterloo, ON, N2L 3G1, e-mail: erik.hintz@uwaterloo.ca

Marius Hofert
Department of Statistics and Actuarial Science, University of Waterloo, 200 University Avenue West, Waterloo, ON, N2L 3G1, e-mail: marius.hofert@uwaterloo.ca

Christiane Lemieux
Department of Statistics and Actuarial Science, University of Waterloo, 200 University Avenue West, Waterloo, ON, N2L 3G1, e-mail: clemieux@uwaterloo.ca

$$\mu = \mathbb{E}(g(\boldsymbol{Y}, W)) \tag{1}$$

where $g : \mathbb{R}^{d+1} \longrightarrow \mathbb{R}$ is integrable and $\boldsymbol{Y} \sim F_{\boldsymbol{Y}}$ is a d-dimensional random vector independent of the random variable $W \sim F_W$. For instance, if $\boldsymbol{Y}$ is multivariate normal and W follows a generalized inverse Gaussian (GIG) distribution (see, e.g., [10] for an AR algorithm to sample from GIG distributions), we could be estimating the expected shortfall of a generalized hyperbolic distribution; this is an important class of multivariate distributions in risk management, see, e.g., [17].

The classical Monte Carlo (MC) estimator $\hat{\mu}_n^{\mathrm{mc}}$ based on n samples for μ is given by

$$\hat{\mu}_n^{\mathrm{mc}} = \frac{1}{n}\sum_{i=1}^{n} g(\boldsymbol{Y}_i, W_i),$$

where $(\boldsymbol{Y}_i, W_i) \overset{\text{ind.}}{\sim} F_{\boldsymbol{Y}} \times F_W$ for $i = 1, \dots, n$.

We assume that there is an easy way to sample from $F_{\boldsymbol{Y}}$ based on uniforms; e.g., based on the Rosenblatt transform ([21]). That is to say, assume there is a transformation $T_{\boldsymbol{Y}} : (0,1)^{d+k} \longrightarrow \mathbb{R}^d$ such that $T_{\boldsymbol{Y}}(\boldsymbol{U}) \sim F_{\boldsymbol{Y}}$ for $\boldsymbol{U} \sim \mathrm{U}^{d+k}$ for constant $k \geq 0$; if, e.g., $\boldsymbol{Y} \sim \mathrm{N}_d(\boldsymbol{\mu}, \Sigma)$ then $k = 0$ and the function $T_{\boldsymbol{Y}}(\boldsymbol{u})$ is given by $T_{\boldsymbol{Y}}(\boldsymbol{u}) = \boldsymbol{\mu} + A(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_d))^\top$ where A is such that $AA^\top = \Sigma$.

In this paper, we investigate the following question:

How can a randomized quasi-Monte Carlo (RQMC) estimator for μ be constructed when W cannot be sampled by inversion?

More precisely, we assume that the (always existing) quantile function $F_W^{\leftarrow}(u) = \inf\{x : F_W(x) \geq u\}$ is intractable and instead we rely on other methods for non-uniform random variate generation (NRVG), such as AR algorithms, where at first glance it may seem hard to directly apply RQMC methods.

We investigate the above question under two sets of assumptions on what we mean by the existence of a "NRVG" method for W.

1. *Black-box case.* Here, we assume that we have a function $R_W : \mathbb{N} \longrightarrow \mathbb{R}^n$ such that if $R_W(n) = \boldsymbol{W}$ for $\boldsymbol{W} = (W_1, \dots, W_n)$ then $W_i \overset{\text{ind.}}{\sim} F_W$ for $i = 1, \dots, n$. As such, we have a "black box" function that returns samples from F_W of any size. The underlying sampling method could be based on MCMC, machine learning techniques or methods based on a stochastic representation (SR), among others. We study this setting in Section 2, along with introducing some notation. The proposed methods, which rely on estimating the quantile function $F_W^{\leftarrow}$, are explored and compared numerically.
2. *AR algorithms for W,* where the proposal (or envelope) distribution and the acceptance decision can be sampled by inversion of uniforms. The main difference to the black-box setting is that here, we do have access to the underlying sampling mechanism and can feed the AR sampler with a randomized low-discrepancy sequence (LDS). AR algorithms are typically not popular in RQMC as it is possibly infinite-dimensional. [18] consider using smoothed rejection and weighted uniform sampling and show in their numerical results that these outperform AR

sampling in terms of convergence speed. [22] show that the F discrepancy, i.e., $\sup_x |F_n(x) - F(x)|$, where F_n and F denote the empirical and theoretical distribution function, of a sample obtained via AR is in $O(n^{-\alpha})$ for $1/2 \leq \alpha < 1$. They improve the error convergence rate by replacing the purely binary AR decision with weights, called extended smoothed rejection. This circumvents integration of an indicator function. [23] derive discrepancy properties of points produced by totally deterministic AR methods, i.e., AR with a (non-randomized) Sobol' sequence. [20] give a convergence result, error bounds and a numerical study for AR with RQMC. What all previous references have in common is that they hold the dimension of the LDS constant and effectively use a subset of size n of the first N points in the sequence. We investigate, among other things, whether there is a difference between holding d constant (and thereby skipping points in the sequence) or holding n constant (thereby thinking of the first n points having potentially unbounded dimension). This is the topic of Section 3.

To be clear, AR could even be an algorithm used within the black-box setting, but given its prevalence, we choose to treat AR separately. We revisit this point at the end of Section 3, where we combine ideas from both settings.

Section 4 applies the methods presented in Sections 2 and 3 to the problem of estimating the price of a basket call option under t copula dependence while Section 5 concludes the paper.

2 Methods for the black box setting

Recall that the classical MC estimator $\hat{\mu}_n^{\text{mc}}$ based on n samples for (1) can be written as

$$\hat{\mu}_n^{\text{mc}} = \frac{1}{n} \sum_{i=1}^{n} g(T_{\boldsymbol{Y}}(\boldsymbol{U}_i), W_i), \tag{2}$$

where $\boldsymbol{U}_i \overset{\text{ind.}}{\sim} \mathrm{U}(0,1)^{d+k}$ is independent of $W_1, \dots, W_n \overset{\text{ind.}}{\sim} F_W$ obtained by calling $R_W(n)$. To simplify the notation, we henceforth assume $k = 0$; the case $k > 0$ is handled by replacing d by $d + k$ in what follows. In order to be able to apply RQMC to the problem, we first rewrite (1) as an integral over the unit hypercube. With a change of variable, we obtain

$$\mu = \int_{(0,1)^{d+1}} g(T_{\boldsymbol{Y}}(u_{1:d}), Q(u_{d+1}))\,\mathrm{d}\boldsymbol{u}, \tag{3}$$

where $\boldsymbol{u} = (u_{1:d}, u_{d+1})$ with $u_{1:d} = (u_1, \dots, u_d)$, and we use the function $Q : [0,1] \to \mathbb{R}$ as a shorthand notation for the quantile function $F_W^{\leftarrow}$ for the remainder of this paper.

If we were able to sample W via inversion, then RQMC sampling could be used to estimate μ using the following approach: Let $\tilde{P}_{b,n} = \{\boldsymbol{u}_{b,1}, \dots, \boldsymbol{u}_{b,n}\} \subseteq [0,1)^{d+1}$, where $\boldsymbol{u}_{b,i} = (u_{b,i,1}, \dots, u_{b,i,d+1})$ for $b = 1, \dots, B$, denote B independent randomizations of the first n points of the low-discrepancy sequence (LDS) used; here we assume that the randomization is such that each $\boldsymbol{u}_{b,i} \sim U(0,1)^{d+1}$. Then

$$\hat{\mu}^{\text{rqmc}}_{b,n} = \frac{1}{n}\sum_{i=1}^{n} g(T_{\boldsymbol{Y}}(\boldsymbol{u}_{b,i,1:d}), Q(u_{b,i,d+1})), \quad b = 1, \dots, B, \tag{4}$$

and an RQMC estimator for μ based on a total of nB points would be given by

$$\hat{\mu}^{\text{rqmc}}_{B,n} = \frac{1}{B}\sum_{b=1}^{B} \hat{\mu}^{\text{rqmc}}_{b,n}.$$

The variance/error of $\hat{\mu}^{\text{rqmc}}_{B,n}$ could then be estimated in the usual way; see [15].

However, we do not know Q, so the estimators $\hat{\mu}^{\text{rqmc}}_{b,n}$ in (4) cannot be computed. In this section, we propose two different methods to *approximate* $\hat{\mu}^{\text{rqmc}}_{b,n}$ for $b = 1, \dots, B$. Both methods essentially replace Q by an estimate thereof.

2.1 Methods based on the empirical quantile function

A simple ad-hoc method to approximate $\hat{\mu}^{\text{rqmc}}_{b,n}$ could be to replace the Q values by a random sample of F_W obtained by calling $R_W(Bn)$. More precisely, let $W_{b,i}$ for $b = 1, \dots, B$, $i = 1, \dots, n$, denote the Bn iid samples from F_W obtained by calling $R_W(Bn)$. Replacing $Q(u_{b,i,d+1})$ by $W_{b,i}$ for $b = 1, \dots, B, i = 1, \dots, n$, is then equivalent to replacing the last coordinate of the n points in $\tilde{P}_{b,n}$ by independent U(0, 1) variates. With $W_{b,i} = Q(U_{b,i})$ where $U_{b,i} \overset{\text{ind.}}{\sim} \mathrm{U}(0,1)$, $b = 1, \dots, B$, $i = 1, \dots, n$, $b = 1, \dots, B$, we can write

$$\hat{\mu}^{\text{mc-rqmc}}_{b,n} = \frac{1}{n}\sum_{i=1}^{n} g(T_{\boldsymbol{Y}}(\boldsymbol{u}_{b,i,1:d}), Q(U_{b,i}))), \quad b = 1, \dots, B. \tag{5}$$

From the inverse probability integral transform (see, e.g., [3, Theorem 2.1]), we know that $Q(U)$ for $U \sim \mathrm{U}(0,1)$ and $R_n(1)$ have the same distribution, namely F_W. As such, unbiasedness of $\hat{\mu}^{\text{mc-rqmc}}_{b,n}$ (and therefore of $(1/B)\sum_{b=1}^{B} \hat{\mu}^{\text{mc-rqmc}}_{b,n}$) for μ follows immediately.

Note that only the first d coordinates of $\tilde{P}_{b,n}$ enter the estimation, so that the good projection properties of coordinate $d+1$ (and its interactions) are lost. Loosely speaking, the last coordinate of the point set we are effectively using to integrate the function g is unrelated with the first d. A better approach is to use the sampled $W_{b,i}$ to construct B empirical quantile functions $\hat{Q}_{n,b}$, $b = 1, \dots, B$, and replace $Q(U_{b,i})$ by $\hat{Q}_{n,b}(u_{b,i,d+1}) = W_{b,(\lceil n u_{b,i,d+1} \rceil)}$, where, for $b = 1, \dots, B$, we denote by $W_{b,(i)}$,

$i = 1, \dots, n$, the order statistics of $W_{b,1}, \dots, W_{b,n}$, so $W_{b,(1)} \le \cdots \le W_{b,(n)}$. We define

$$\hat{\mu}_{b,n}^{\text{b-eqf}} = \frac{1}{n} \sum_{i=1}^{n} g(T_{\boldsymbol{Y}}(\boldsymbol{u}_{b,i,1:d}), W_{b,(\lceil n u_{b,i,d+1} \rceil)}), \quad b = 1, \dots, B,$$

where superscript "b-eqf" indicates that in each randomization b, the empirical quantile function obtained in that randomization based on n samples is used (instead of Q). That is, in each of the B randomizations (each of which requires n function evaluations), estimate Q by its empirical quantile function $\hat{Q}_{n,b}$ obtained from n independent samples from F_W via a call to the black-box function $R_W(n)$.

Note that as long as $\tilde{P}_{b,n,d+1} = \{u_{b,i,d+1} : i = 1, \dots, n\}$ is *properly stratified*, i.e., has exactly one point in each interval of the form $[j/n, (j+1)/n)$ for $j \in \{0, \dots, n-1\}$, each $W_{b,i}$, $i = 1, \dots, n$ will be sampled exactly once when using $\tilde{P}_{b,n,d+1}$ to sample the empirical quantile function $\hat{Q}_{b,n}$. Hence an alternative way to describe the estimator $\hat{\mu}_{b,n}^{\text{b-eqf}}$ that is useful from an implementation perspective is to realize that if the last coordinate of a given point $\boldsymbol{u}_{b,i}$ is the jth smallest value among those n last coordinates, we "stitch" $W_{b,(j)}$ to that ith point. Hence the last coordinate of $\tilde{P}_{b,n}$ is used to order the sample $W_{b,1}, \dots, W_{b,n}$. Note that the problem of concatenating two samples within an RQMC-based approach also appears in the Array-RQMC method of [13, 16], where one needs to assign particles to points when paths are propagated. Also note that if $\tilde{P}_{b,n}$ is a digitally shifted or scrambled Sobol' point set with $n = b^k$ points or a randomly shifted rank-1 lattice, then $\tilde{P}_{b,n,d+1}$ is properly stratified; see [14].

The estimators $\hat{\mu}_{b,n}^{\text{b-eqf}}$ for $b = 1, \dots, B$ are independent and as long as $\tilde{P}_{b,n,d+1}$ is properly stratified, they are also unbiased, see Proposition 1.

This alternative description gives rise to a slightly different estimator: Let $r^n(u_{b,i,d+1})$ be the rank of $u_{b,i,d+1}$ among $u_{b,1,d+1}, \dots, u_{b,n,d+1}$. We then define the rank-based estimator as

$$\hat{\mu}_{b,n}^{\text{b-rk}} = \frac{1}{n} \sum_{i=1}^{n} g(T_{\boldsymbol{Y}}(\boldsymbol{u}_{b,i,1:d}), W_{b,(r^n(u_{b,i,d+1}))}), \quad b = 1, \dots, B. \tag{6}$$

If $\tilde{P}_{b,n,d+1}$ is properly stratified, then $\hat{\mu}_{b,n}^{\text{b-rk}}$ and $\hat{\mu}_{b,n}^{\text{b-eqf}}$ coincide, and each sample $W_{b,i}$ is used exactly once. Otherwise, unlike $\hat{\mu}_{b,n}^{\text{b-eqf}}$, $\hat{\mu}_{b,n}^{\text{b-rk}}$ still uses every $W_{b,i}$ exactly once.

Proposition 1 *Let $b \in \{1, \dots, B\}$ and let $\tilde{P}_{b,n,d+1}$ be properly stratified. Then $\hat{\mu}_{b,n}^{b\text{-}rk}$ (and therefore $\hat{\mu}_{b,n}^{b\text{-}eqf}$) is unbiased for μ.*

Proof Let $i \in \{1, \dots, n\}$. We show that $\mathbb{E}(g(T_{\boldsymbol{Y}}(\boldsymbol{u}_{b,i,1:d}), W_{b,(r^n(u_{b,i,d+1})),n})) = \mu$. By definition, $(u_{b,i,1}, \dots, u_{b,i,d+1}) \sim \mathrm{U}(0,1)^{d+1}$, in particular, $\boldsymbol{Y} := T_{\boldsymbol{Y}}(\boldsymbol{u}_{b,i,1:d}) \sim F_{\boldsymbol{Y}}$ is independent of $u_{b,i,d+1}$. Let $r^n(u_{b,i,d+1}) = K(i)$ (a random variable) and note that $(K(1), \dots, K(n))$ is a permutation of $(1, \dots, n)$ chosen according to some distribution (which may not be uniform because of the low-discrepancy properties of

$\tilde{P}_{b,n}$). Then $W_{b,K(i)}$ is an element chosen from the list $W_{b,1},\dots,W_{b,n}$ according to some distribution, and the latter is an independent random sample from F_W. Hence, $W_{b,K(i)}$ and $\boldsymbol{Y}$ are independent, $(\boldsymbol{Y}, W_{b,K(i)}) \sim F_{\boldsymbol{Y}} \times F_W$ and the main claim follows by linearity of the expectation. □

The previous methods can be thought of as approximating the quantile function B times, each based on n samples obtained from the black box. In order to base our simulation on a sampling mechanism closer to inversion and thereby mimicking more closely the estimator in (4), we could instead construct a single rank-based quantile function estimator based on the Bn outputs $W_{b,i}$, $b=1,\dots,B$, $i=1,\dots,n$. That is, instead of reordering the n samples $W_{b,i}$, $i=1,\dots,n$ according to $u_{b,i,d+1}$ in each randomization $b=1,\dots,B$, separately, we reorder the Bn realizations $W_{b,i}$, $i=1,\dots,n$, $b=1,\dots,B$, according to the ranks of the $u_{b,i,d+1}$. That is, we construct the estimator

$$\hat{\mu}^{\text{1:B-rk}}_{b,n} = \frac{1}{n}\sum_{i=1}^{n} g(T_{\boldsymbol{Y}}(\boldsymbol{u}_{b,i,1:d}), W_{(r^{Bn}(u_{b,i,1}))}), \quad b=1,\dots,B, \tag{7}$$

where $r^{Bn}(u_{b,i,d+1}) = k$ if $u_{b,i,d+1}$ is the kth smallest among the Bn uniforms $u_{1,1,d+1},\dots,u_{1,n,d+1},\dots,u_{B,1,d+1},\dots,u_{B,n,d+1}$.

We can replace the ranks by the empirical quantile function computed from $R_W(Bn)$, and obtain as an analog of $\hat{\mu}^{\text{b-eqf}}$ the estimator

$$\hat{\mu}^{\text{1:B-eqf}}_{b,n} = \frac{1}{n}\sum_{i=1}^{n} g(T_{\boldsymbol{Y}}(\boldsymbol{u}_{b,i,1:d}), W_{(\lceil nBu_{b,i,1}\rceil)}), \quad b=1,\dots,B,$$

for a sample $W_1,\dots,W_{nB} \overset{\text{ind.}}{\sim} F_W$ obtained by calling $R_W(Bn)$. The superscript "1:B-eqf" shall indicate that in all randomizations $1,\dots,B$, the same quantile function estimator is used. Note that $\hat{\mu}^{\text{1:B-eqf}}_{b,n}$ are not independent anymore for $b=1,\dots,B$, the same applies to $\hat{\mu}^{\text{1:B-rk}}_{b,n}$.

Here we note that Pierre L'Ecuyer and his collaborators have proposed very efficient methods that combine conditional MC and RQMC to estimate quantile functions associated with a simulation output; see [19]. Our setting here is different, as we focus on estimating a univariate quantile function for the sole purpose of sampling.

2.2 Methods based on a Generalized Pareto approximation in the tail

The methods presented in the previous section are purely nonparametric and amount to replacing the true quantile function Q by an empirical estimate thereof. Empirical quantile functions typically estimate quantiles away from the tail with reasonable accuracy; this does not hold for the tails if W is unbounded. However, approximating the tail of Q well is crucial for an effective RQMC procedure to outperform MC.

In the following, assume that W is supported on $[0, \infty)$ so that only the upper tail needs to be estimated. Since this is typically the case in practice, this is a rather weak assumption. If W is instead supported on $\mathbb{R}$, the methods described here can be applied to the positive and negative real line separately.

The main idea behind the methods presented in this section is the following: Given a random sample from F_W, estimate Q in the body (say, for $u \in (0, 0.9)$) by interpolation of the empirical quantile function and in the (right) tail based on a fitted generalized Pareto Distribution (GPD), which has a cumulative distribution function (cdf)

$$G_{\xi,\beta}(x) = \begin{cases} 1 - \left(1 + \frac{\xi x}{\beta}\right)^{-\frac{1}{\xi}}, & \text{if } \xi \neq 0, \\ 1 - \exp\left(-\frac{x}{\beta}\right), & \text{if } \xi = 0, \end{cases}$$

where $\beta > 0$ and the support is $x \in [0, \infty)$ when $\xi \geq 0$ and $x \in [0, -\beta/\xi]$ when $\xi < 0$.

Let F be any cdf and let $X \sim F$. Denote by $F_u(x) = \mathbb{P}(X - u \leq x \mid X > u)$ the excess distribution over the threshold u. Under weak assumptions, the Pickands-Balkema-de-Haan Theorem (see [4, Theorem 3.4.13]) implies that for large enough u one can approximate F_u by $G_{\xi,\beta}$.

In practice, ξ and β are estimated from given data. With estimates of ξ, μ at hand, we can compute $G^{-1}_{\xi,\beta}$ analytically, which, appropriately scaled, provides us with an estimate of F^{-1}. In what follows, assume F_W fulfills the assumptions underlying the Pickands-Balkema-de-Haan Theorem, and denote by $g_{\xi,\beta}$ the density of $G_{\xi,\beta}$. The following algorithm returns a quantile function estimator $\hat{Q}$ of Q.

Algorithm 1 Given $W_1, \dots, W_{n'} \overset{\text{ind.}}{\sim} F_W$ and $\alpha \in (0, 1)$, construct an estimator $\hat{Q}$ for Q as follows:

1. Denote by $W_{(1)}, \dots, W_{(n')}$ the order statistics of $W_1, \dots, W_{n'}$.
2. Let $T = W_{(\lceil n'\alpha \rceil)}$ and denote by $N = |\{i \in \{1, \dots, n'\} : W_i > T\}|$ the number of exceedances over T. Let $\tilde{W}_i = W_{(\lceil n'\alpha \rceil + i)} - T$ for $i = 1, \dots, N$ be the excesses. Then maximize the log-likelihood function

$$l(\xi, \beta; \tilde{W}_1, \dots, \tilde{W}_N) = \sum_{k=1}^{N} \log g_{\xi,\beta}(\tilde{W}_k)$$

   with respect to ξ and β numerically over their ranges to obtain the MLEs $\hat{\xi}$ and $\hat{\beta}$.
3. Return the function

$$\hat{Q}(u) = \begin{cases} (1-\kappa)W_{(\lfloor (n'+1)u \rfloor)} + \kappa W_{(\lfloor (n'+1)u \rfloor + 1)}, & \text{if } u \leq \alpha, \\ T + \frac{\hat{\beta}}{\hat{\xi}}\left(\left(\frac{1-u}{1-\alpha}\right)^{-\hat{\xi}} - 1\right), & \text{otherwise,} \end{cases}$$

   where $\kappa = (n' + 1)u - \lfloor (n' + 1)u \rfloor$.

Algorithm 1 does not give any error estimates, nor do we have an a-priori guess of how large n should be. In order to obtain error estimates, one could use Algorithm 1 to obtain M independent estimators $\hat{Q}_m$, $m = 1, \dots, M$, and estimate the error using a CLT argument. That is, the (absolute) error of $\hat{Q}(u) = (1/M)\sum_{m=1}^{M}\hat{Q}_m(u)$ for some fixed $u \in (0,1)$ may be estimated via $3.5/\sqrt{M} \times \hat{\sigma}$, where $\hat{\sigma} = \mathrm{sd}(\hat{Q}_1(u), \dots, \hat{Q}_M(u))$. As $\hat{Q}_m(u) - Q(u)$ follows approximately a $\mathbb{N}(0, \hat{\sigma}^2/M)$ distribution, we can be 99.95% confident that the error is within $\pm 3.5/\sqrt{M} \times \hat{\sigma}$.

With an error estimation procedure at hand, one can now construct the quantile function iteratively until a pre-specified error tolerance for the estimated absolute error is met. That is, one can specify knots $u'_1, \dots, u'_N \in (0,1)$ and error tolerances $\varepsilon_1, \dots, \varepsilon_N > 0$ and construct the quantile function with more and more points until the error tolerance at all knots is met. The choice of knots and error tolerances can be guided from the function g so that important subdomains have little error, or one can put most of the knots uniformly between 0 and 1 and the remaining ones in the tails. The main idea is summarized in the following algorithm.

Algorithm 2 Given $n_0 \in \mathbb{N}$, $\alpha \in (0,1)$, NRVG R_W, knots $u'_1, \dots, u'_N \in (0,1)$, error tolerances $\varepsilon_1, \dots, \varepsilon_N > 0$, maximum number of iterations $i_{\max}$, $B \in \mathbb{N}$, construct an estimator $\hat{Q}$ for Q as follows:

```
1. Set $i = 1$, and $S_b = \{\}$ for $k = 1, \dots, B$.
2. Repeat
    a. For $b = 1, \dots, B$,
        i. Set $S_b = S_b \cup \{R_W(n_0)\}$.
        ii. Call Algorithm 1 with input sample $S_b$ to construct an estimated quantile
            function $\hat{Q}_b$.
    b. For $k = 1, \dots, N$ set $e_k = 3.5/\sqrt{B} \times \mathrm{sd}(\hat{Q}_1(u'_k), \dots, \hat{Q}_B(u'_k))$ as the estimated
        error at knot $u'_k$.
    c. Set $i = i + 1$.
   Until $e_k \leq \varepsilon_k$ for $k = 1, \dots, N$ or $i > i_{\max}$.
3. Return the estimated quantile function $\hat{Q}^{\text{eqf-gpd}}(u) = (1/B)\sum_{b=1}^{B}\hat{Q}_b(u)$.
```

The input argument $i_{\max}$ determines the maximum number of iterations allowed in case convergence cannot be achieved. Note that the superscript "eqf-gpd" shall indicate that the (interpolated) empirical quantile function is used in the body and a GPD approximation in the tail. For an implementation, in any iteration $i > 1$, results from the previous iterations should be reused; for instance, the MLE $(\hat{\xi}, \hat{\beta})$ from a previous iteration can be used as a starting value for the maximization of the log-likelihood function in the next iteration. In practice one could also return the $\hat{Q}_b$, $b = 1, \dots, B$, so that for any $u \in (0,1)$ one can compute $\hat{Q}(u)$ along with an error estimate.

Given an estimated quantile function, say $\hat{Q}^{\text{eqf-gpd}}$, an RQMC estimator for μ from (1) is given by

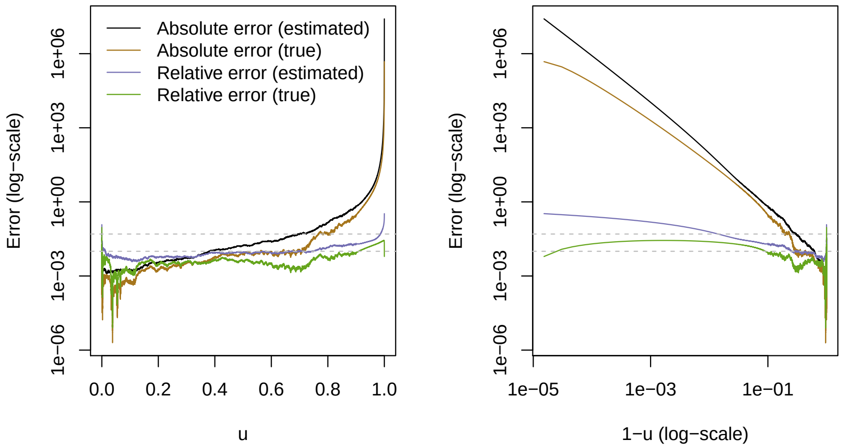

Fig. 1 Estimated and realized absolute and relative errors when estimating the quantile function of IG$(1.2, 1.2)$ using Algorithm 2 with $n_0 = 7500$, $B = 20$.

$$\hat{\mu}^{\text{eqf-gpd}}_{B,n} = \frac{1}{B}\sum_{b=1}^{B}\hat{\mu}^{\text{eqf-gpd}}_{b,n}, \tag{8}$$

where

$$\hat{\mu}^{\text{eqf-gpd}}_{b,n} = \frac{1}{n}\sum_{i=1}^{n} g(T_{\boldsymbol{Y}}(\boldsymbol{u}_{b,i,1:d}), \hat{Q}^{\text{eqf-gpd}}(u_{b,i,d+1})), \quad b = 1,\dots,B,$$

and the inputs $u_{b,i,d+1}$ and $\boldsymbol{u}_{b,i,1:d}$ are as in the previous section. In contrast to the estimators from Section 2.1, computing this estimator requires a two-stage procedure: First, Algorithm 2 needs to be applied to compute the estimated quantile function $\hat{Q}^{\text{eqf-gpd}}$, which will then, in the second stage, be treated as the “true quantile function“ when computing the estimator $\hat{\mu}^{\text{eqf-gpd}}$.

Inverse-gamma example

Consider $W \sim \text{IG}(1.2, 1)$. We use $n_0 = 7500$, $B = 20$, and uniform knots between 0.01 and 0.95 with relative error tolerance 0.025, one knot at 0.99 with relative error tolerance 0.075 and and another knot at 0.999 with relative error tolerance 0.1. The algorithm needed 10 iterations until convergence, so a total of 1 350 000 realizations of W. The approximation is very accurate and the true quantile lies within the approximated error bounds. This can be seen from Figure 1, which displays realized and estimated absolute and relative errors.

Expected shortfall of portfolio under a multivariate *t* distribution

The multivariate t distribution is a normal variance mixture distribution and falls into the general framework of this paper, if we assume that the quantile function of an inverse-gamma distribution is not available. We do this to compare our methods with the "best possible" estimator from (4). Let $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\Sigma = AA^\top$ for some covariance matrix Σ. Then $\boldsymbol{X} \sim t_d(\nu, \boldsymbol{\mu}, \Sigma)$ has stochastic representation

$$\boldsymbol{X} = \boldsymbol{\mu} + \sqrt{W}\boldsymbol{Y}, \tag{9}$$

where $W \sim \mathrm{IG}(\nu/2, \nu/2)$ independent of $\boldsymbol{Y} \sim \mathbb{N}_d(\boldsymbol{0}, \Sigma)$. For a continuous random variable $L \sim F$ with $\mathbb{E}(|L|) < \infty$ and level $\alpha \in (0, 1)$ small, expected shortfall is the mean conditional loss $\mathrm{ES}_\alpha(L) = \mathbb{E}(L \mid L > F_L^{-1}(\alpha))$. In our simulation, we assume that $L = \boldsymbol{1}^\top \boldsymbol{X}$ where $\boldsymbol{X} \sim t_d(\nu, \boldsymbol{0}, \Sigma)$; it follows from the closedness of normal variance mixtures that $L \sim t_1(\nu, 0, \boldsymbol{1}^\top \Sigma \boldsymbol{1})$. The value of $\mu = \mathrm{ES}_\alpha(L) := \mathbb{E}(g(\boldsymbol{Y}, W))$ is known in closed-form; see [17, Example 2.15]. This allows us to estimate the mean squared error (MSE) and compare it with the variance. For a range of values of the total number of function evaluations, we report in Figure 2 the mean squared error (MSE) and variance for various methods, each estimated by using $M = 50$ independent copies of the estimators, each of which is based on $B = 20$ repetitions. Here and in what follows, we use a digitally shifted Sobol' sequence as implemented in the R package qrng; see [9]. All RQMC based estimators, including MC-RQMC from (5), outperform MC, though MC-RQMC gives only a moderate variance reduction. This is in contrast to b-rk, which for small n gives MSE similar to inversion, which we recall would not be available in a realistic setting where Q is unknown.

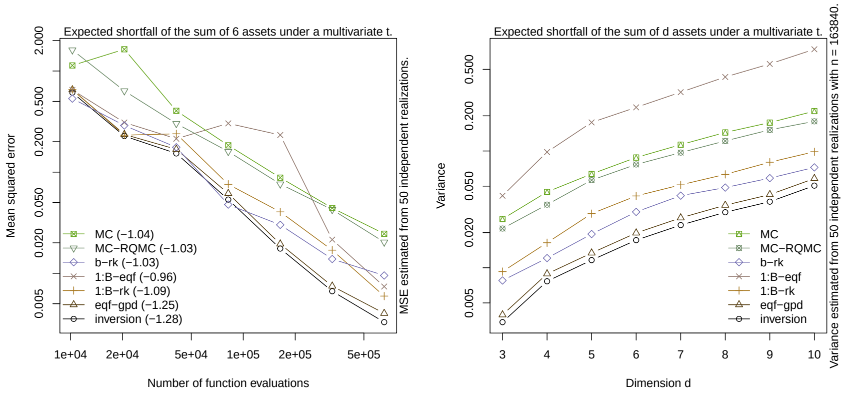

Fig. 2 Mean squared errors as a function of n (left) and variances as a function of d (right) when estimating $\mathrm{ES}_{0.95}(L)$ for $L = \boldsymbol{1}^\top \boldsymbol{X}$ where $\boldsymbol{X} \sim t_d(\nu, \boldsymbol{0}, \Sigma)$.

3 Combining AR with RQMC

Rather than working with a “black-box” RVG R_W, we assume in this section that W can be sampled using AR and explore how we can apply RQMC in this setting. Recall from (3) that we are interested in estimating $\mu = \mathbb{E}(g(\boldsymbol{Y}, W))$, so we need n samples $(\boldsymbol{Y}_i, W_i)$ where $W_i \sim F_W$. When using AR, there is no a-priori bound on how many uniforms are needed, so we have an a priori infinite-dimensional integration problem: If T_{AR} denotes the AR transformation, we can write $\mu = \mathbb{E}(h(\boldsymbol{U})) = \mathbb{E}(g(T_{\boldsymbol{Y}}(\boldsymbol{U}_{1:d}), T_{\text{AR}}(\boldsymbol{U}_{(d+1):\infty}))$ with $\boldsymbol{U} \sim \mathrm{U}(0,1)^\infty$ and h appropriately defined. The integrand h is a non-monotone and discontinuous function of its input uniforms, a result from the acceptance decision. This can diminish the variance reduction effect of RQMC over MC.

We assume that W has density f_W over $(a, b) \subseteq \mathbb{R}$, we use the proposal density f having the same support (a, b) with quantile function F^{-1}, and that $c = \sup_{x \in (a,b)} f_W(x)/f(x) < \infty$.

A major difference between the application of RQMC and MC is that with the former, we need to carefully assign which coordinate of the points is used to sample which random variable, and there is typically more than one way to do so. As in the previous section, we assume that the first d coordinates $\boldsymbol{u}_{1:d}$ of $\boldsymbol{u} \in (0,1)^\infty$ are used to sample from $F_{\boldsymbol{Y}}$. Algorithms 3 and 4 describe two AR methods to sample n copies

i	Sample $\boldsymbol{Y}$				F^{-1}	AR
1	$u_{1,1}$	$u_{1,2}$	$\dots$	$u_{1,d}$	$u_{1,d+1}$	$u_{1,d+2}$
2	$u_{2,1}$	$u_{2,2}$	$\dots$	$u_{2,d}$	$u_{2,d+1}$	$u_{2,d+2}$
3	$u_{3,1}$	$u_{3,2}$	$\dots$	$u_{3,d}$	$u_{3,d+1}$	$u_{3,d+2}$
4	$u_{4,1}$	$u_{4,2}$	$\dots$	$u_{4,d}$	$u_{4,d+1}$	$u_{4,d+2}$
5	$u_{5,1}$	$u_{5,2}$	$\dots$	$u_{5,d}$	$u_{5,d+1}$	$u_{5,d+2}$
6	$u_{6,1}$	$u_{6,2}$	$\dots$	$u_{6,d}$	$u_{6,d+1}$	$u_{6,d+2}$
$\vdots$	$\vdots$				$\vdots$	$\vdots$

Fig. 3 Schematic description of AR-n. Gray coordinates in the same row correspond to rejected coordinates.

i	Sample $\boldsymbol{Y}$				F^{-1}	AR	F^{-1}	AR	F^{-1}	AR	$\dots$
1	$u_{1,1}$	$u_{1,2}$	$\dots$	$u_{1,d}$	$u_{1,d+1}$	$u_{1,d+2}$	$u_{1,d+3}$	$u_{1,d+4}$	$u_{1,d+5}$	$u_{1,d+6}$	
2	$u_{2,1}$	$u_{2,2}$	$\dots$	$u_{2,d}$	$u_{2,d+1}$	$u_{2,d+2}$					
3	$u_{3,1}$	$u_{3,2}$	$\dots$	$u_{3,d}$	$u_{3,d+1}$	$u_{3,d+2}$					
4	$u_{4,1}$	$u_{4,2}$	$\dots$	$u_{4,d}$	$u_{4,d+1}$	$u_{4,d+2}$	$u_{4,d+3}$	$u_{4,d+4}$			

i	Sample $\boldsymbol{Y}$				F^{-1}	F^{-1}	F^{-1}	$\dots$	AR	AR	AR	$\dots$
1	$u_{1,1}$	$u_{1,2}$	$\dots$	$u_{1,d}$	$u_{1,d+1}$				$u_{1,d+M+1}$			
2	$u_{2,1}$	$u_{2,2}$	$\dots$	$u_{2,d}$	$u_{2,d+1}$	$u_{2,d+2}$	$u_{2,d+3}$		$u_{2,d+M+1}$	$u_{2,d+M+2}$	$u_{2,d+M+3}$	
3	$u_{3,1}$	$u_{3,2}$	$\dots$	$u_{3,d}$	$u_{3,d+1}$				$u_{3,d+M+1}$			
4	$u_{4,1}$	$u_{4,2}$	$\dots$	$u_{4,d}$	$u_{4,d+1}$	$u_{4,d+2}$			$u_{4,d+M+1}$	$u_{4,d+M+2}$		

Fig. 4 Schematic description of AR-d with consecutive (top) and blockwise (bottom) coordinate assignment. Gray coordinates in the same row correspond to rejected coordinates.

of $(\boldsymbol{Y}, W)$; a schematic description is given in Figures 3 and 4. The former method, henceforth referred to as AR-n, always uses coordinates $\{d+1, d+2\}$ in the AR part, and moves along the index i. If a point is rejected, just like the point in row $i = 1$ in Figure 3, the algorithm tries again with point $i + 1$. That is, when sampling n points we move along the index of a randomized LDS with constant dimension $d + 2$. In contrast, Algorithm 4 (AR-d) samples the ith point by moving along the coordinates $\{d + 1, d + 2, d + 3, \dots\}$ of the ith point in the sequence until it is accepted; see the top of Figure 4, where we assume that coordinates $d + 2j - 1$ and $d + 2j$ for $j = 1, 2, \dots$ are used for sampling from the proposal and sampling from the AR decision, respectively. Another possibility to assign the coordinates for the AR part is to consider two blocks of size M (chosen so that, with high probability, M trials are sufficient to accept a point), where the coordinates in the first block are used for the sampling in Step 2(a)i and the coordinates in the second block determine the acceptance decision in Step 2(a)ii. This version of AR-d is illustrated at the bottom of Figure 4.

Algorithm 3 (AR-*n*) Let $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \} \subset (0, 1)^{d+2}$ be a randomized LDS. Sample n copies of $(\boldsymbol{Y}, W)$ as follows.

```
1. Set $j = 1, O_n = \{\}$.
2. For $i = 1, \dots, n$,
   a. Repeat
      i. Compute $W = F^{-1}(u_{j,d+1})$ and set $U = u_{j,d+2}$.
      ii. If $U > f_W(W)/(cf(W))$ set $j = j + 1$ Else
            Set $O_n = O_n \cup \{(T_{\boldsymbol{Y}}(\boldsymbol{u}_{j,1:d}, W)\}$
            Set $j = j + 1$ and break;
3. Return $O_n$.
```

The main difference between AR-n and AR-d is that in the former approach, points in the sequence are skipped, and, effectively, a subset of size n of the first $N > n$ points in the sequence is used to integrate g, whereas in AR-d we always use the first n points in the sequence and move along the coordinates.

Algorithm 4 (AR-*d*) Let $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_n\} \subset (0, 1)^{\infty}$ be a randomized low discrepancy point-set. Sample n copies of $(\boldsymbol{Y}, W)$ as follows.

```
1. Set $O_n = \{\}$.
2. For $i = 1, \dots, n$,
   a. For $j = 1, 2, \dots,$
      i. Compute $W = F^{-1}(u_{i,d+2j-1})$.
      ii. If $u_{i,d+2j} \leq f_W(W)/(cf(W))$:
         A. Set $O_n = O_n \cup \{(T_{\boldsymbol{Y}}(\boldsymbol{u}_{i,1:d}, W)\}$
         B. Break.
3. Return $O_n$.
```

A potential advantage of AR-d over AR-n for numerical integration is that it really only uses the first n points of the LDS rather than a subset of the first $N > n$ points in the sequence. In order to highlight this point, assume that our integrand does not depend on W and that $n = 2^k$. When estimating μ based on AR-d, we will then use the first 2^k points of the underlying LDS and keep all its good projection properties. In contrast, using AR-n, we only use a subset of size 2^k of the first $N > 2^k$ points, thereby potentially loosing some of the good projection properties of the LDS. This point is illustrated in Figure 5, where we first sample (W_i, Y_{i1}, Y_{i2}) where $Y_{ij} \sim \mathbb{N}(0,1)$, $j = 1, 2$, and $W_i \sim \Gamma(1.2, 1)$ for $i = 1, \dots, n = 2^7$, and then set $\boldsymbol{U}_i = (F_W(W_i), \Phi(Y_{i1}), \Phi(Y_{i2}))$ for $i = 1, \dots, n$. By the probability integral transformation, $\boldsymbol{U}_i \sim \mathrm{U}(0,1)^3$. Note that if we had used inversion to sample the W_i, the points would be exactly the original LDS. Note how Sobol'-d gives a point set with better marginal uniformity than Sobol'-n, which is also confirmed in the histogram of the first standardized coordinate in Figure 6. Note that if we had 2^k bins with $k \leq 7$ we would see a flat histogram on the right-hand side of this figure; here and it what follows, we use the AR samplers for the Gamma distribution from [1] and [11] for $\nu > 1$ and $\nu < 1$, respectively.

Next, we show in Propositions 2 and 3 that both algorithms produce point sets with the correct distribution.

Proposition 2 *Each* $\boldsymbol{x} \in O_n$ *produced by Algorithm 3 has distribution* $F_{\boldsymbol{Y}} \times F_W$.

Proof It suffices to show that the two numbers used to sample from the proposal and the acceptance decision are independent $\mathrm{U}(0,1)$ random variables. The rest follows from the correctness of the AR algorithm; see, e.g., [5] for a proof. Let $\boldsymbol{x} = (\boldsymbol{Y}, W) \in O_n$. Then there is a $j \in \{1, 2, \dots\}$ such that $W = F^{-1}(U_1)$ and $U_2 \leq f_W(W)/(cf(W))$ where $U_1 = u_{j,d+1}$ and $U_2 = u_{j,d+2}$ satisfy $U_1, U_2 \overset{\text{ind.}}{\sim} \mathrm{U}(0,1)$ by the randomization of the LDS. □

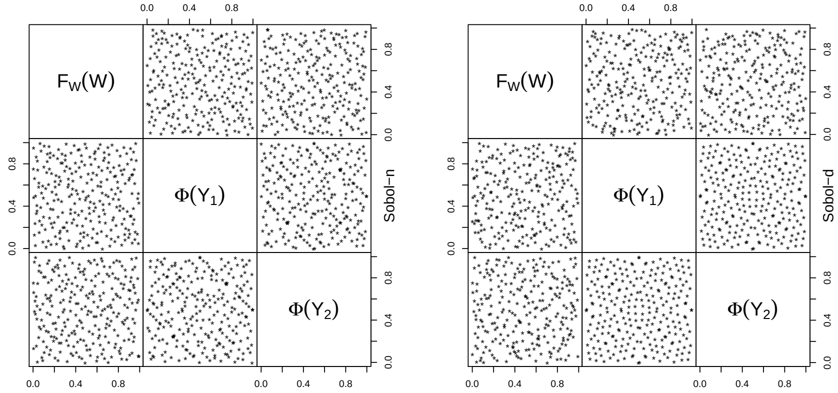

Fig. 5 Pairs plot of $(F_W(W_i), \Phi(Z_{i1}), \Phi(Z_{i2})) \sim \mathrm{U}(0,1)^3$, where the trivariate points were sampled with AR-n (left) and with AR-d (right) for $W \sim \Gamma(1.2, 1)$ and $n = 2^8$.

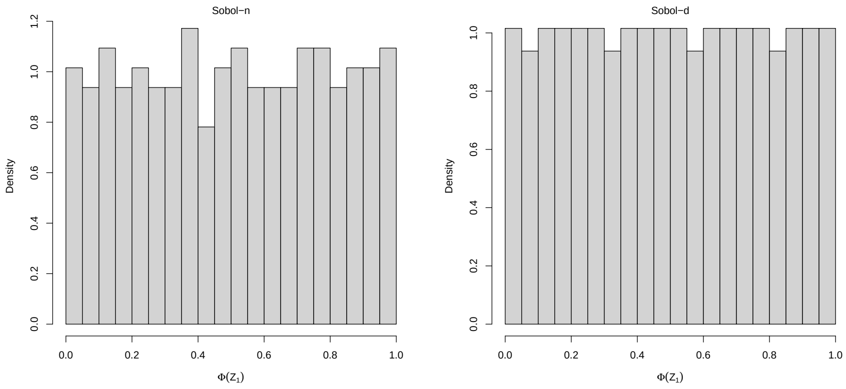

Fig. 6 Histogram of U_1 when constructed with AR-n (left) and AR-d (right).

Proposition 3 *Each $\boldsymbol{x} \in O_n$ produced by Algorithm 4 has distribution $F_{\boldsymbol{Y}} \times F_W$.*

Proof Since we assumed that the chosen LDS is randomized so that each $\boldsymbol{u}_{b,i} \sim U(0,1)^{d+1}$, the coordinates $u_{i,d+j}$ used in Step 2(a)i and 2(a)ii are independent U(0, 1) for $j \geq 1$. The claim follows from the correctness of the AR algorithm. □

Our investigation of AR-d was motivated by the argument that AR corresponds to infinite-dimensional integration; see [6, p. 62–63], who also notes that "potential drawback of AR methods, compared with the inverse transform method, is that their outputs are generally neither continuous nor monotone functions of the input uniforms." We can address the monotonicity by using the rank transformations from the black box setting in Section 2: that is, we re-order the outputs $W_1, \dots, W_n$ so that their order matches the ordering of $u_{1,d+1}, \dots, u_{n,d+1}$. If $n = 2^k$, this is exactly the b-rk method from Section 2 applied with the output of AR-d as a "black box". Note that this makes the AR-d output monotone in coordinate $d+1$ of the underlying LDS. Note that with AR-n, we always use $u_{i,d+1}$ for some i to sample from the envelope via inversion, so that the monotonicity in this coordinate is already given.

Expected shortfall example continued

We perform the same example as on page 10, but this time, using the AR based methods instead of the black-box setting. See Figure 3. All AR based methods outperform pure MC and MC-RQMC, and the convergence speed of AR-n and AR-d, 1:B-rk are almost as high as for the method "inversion", which we recall would not be available in a realistic setting.

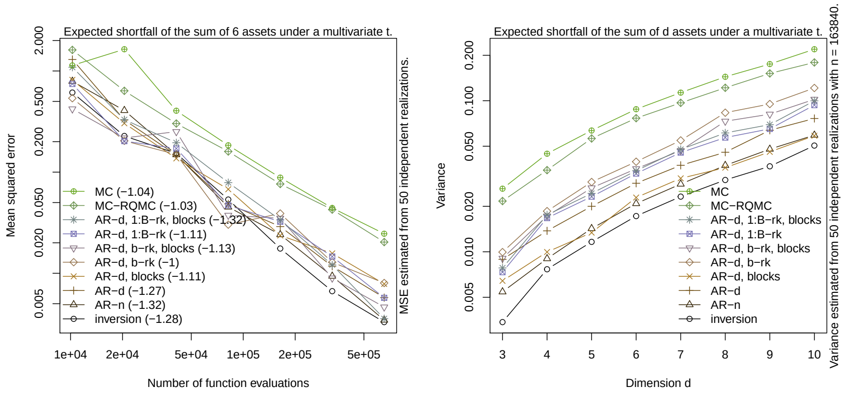

Fig. 7 Mean squared errors as a function of n (left) and variances as a function of d (right) when estimating $\mathrm{ES}_{0.95}(L)$ for $L = \mathbf{1}^\top \boldsymbol{X}$ where $\boldsymbol{X} \sim t_d(\nu, \mathbf{0}, \Sigma)$.

4 Application: Basket option pricing

Consider the problem of estimating the value of a Basket call option with strike K, whose payoff with maturity $T = 1$, can be expressed as

$$\mu_{\text{bskt}} = e^{-r}\mathbb{E}\left(\max\left\{\frac{1}{d}\sum_{j=1}^{d} S_j - K, 0\right\}\right);$$

we assume that the dependence of the log-normal assets S_j, $j = 1, \dots, d$, is modelled via a t-copula. As such, the assets S_j have stochastic representation

$$S_j = F_{\text{LN}}^{-1}(U_j), \quad U_j = F_{t_\nu}(X_j), \quad j = 1, \dots, d, \quad \boldsymbol{X} \sim t_d(\nu, \mathbf{0}, \Sigma);$$

here, Σ is a correlation matrix. The t copula is one of the most widely used copulas in risk management; see, e.g., [2] for more. Pricing basket options is a popular problem to perform RQMC experiments; see, e.g., [12]. The value of μ_{bskt} is not known, so we look at the estimated variances for the following methods:

- `MC:` Use MC for W and $\boldsymbol{Y}$;
- `MC-RQMC:` use MC for W (and inversion based on RQMC for $\boldsymbol{Y}$), i.e., compute $\hat{\mu}^{\text{mc-rqmc}}$ in (5).
- `AR-d:` Use Algorithm 4, i.e., sample W based on AR whilst moving along the coordinates of a point in the LDS until acceptance.
- `AR-n:` Use Algorithm 3, i.e., sample W based on AR whilst moving along the index of the point in the LDS until acceptance.
- `AR-d, b-rk:` Use AR-d in each repetition b and additionally reorder the n samples $W_{1,b}, \dots, W_{n,b}$ according to $u_{1,b,d+1}, \dots, u_{n,b,d+1}$ for $b = 1, \dots, B$.

- `AR, 1:B-rk:` Use AR-d and sort all the sample $W_{1,1}, \dots, W_{n,B}$ according to $u_{1,1,d+1}, \dots, u_{n,B,d+1}$.
- `b-rk:` Treat R_W as black-box and compute $\hat{\mu}_{b,n}^{\text{b-rk}}$ from (6) for $b = 1, \dots, B$.
- `1:B-rk:` Treat R_W as black-box and compute $\hat{\mu}_{b,n}^{\text{1:B-rk}}$ from (7) for $b = 1, \dots, B$.
- `eqf-gpd:` First, build gpd based estimate $\hat{Q}$ using samples obtained from the black box R_W, then treat it as true Q and proceed with inversion; see$\hat{\mu}^{\text{eqf-gpd}}$ in (8).
- `inversion:` Compute the inversion based estimator $\hat{\mu}_{b,n}^{\text{rqmc}}$ from (4) for $b = 1, \dots, B$ using the true quantile function.

The last method "inversion" is not available in a realistic setting like the GIG example at the end of this section, but is included here to compare our methods with the best possible one. All methods (except for MC) sample the multivariate normal random vector $\boldsymbol{Y}$ based on inversion of a digitally shifted Sobol' sequence.

The results in Figure 8 indicate that using RQMC for sampling $\boldsymbol{Y}$ gives at least a modest variance reduction. Furthermore, treating the sampler as a black box and reordering the W samples as described in Section 2 gives further variance reduction. On the right hand side we see the AR methods from Section 2 which all give lower variance than the black-box methods; this makes sense as we are directly manipulating the sampler. with some rank reordering, outperform AR-n.

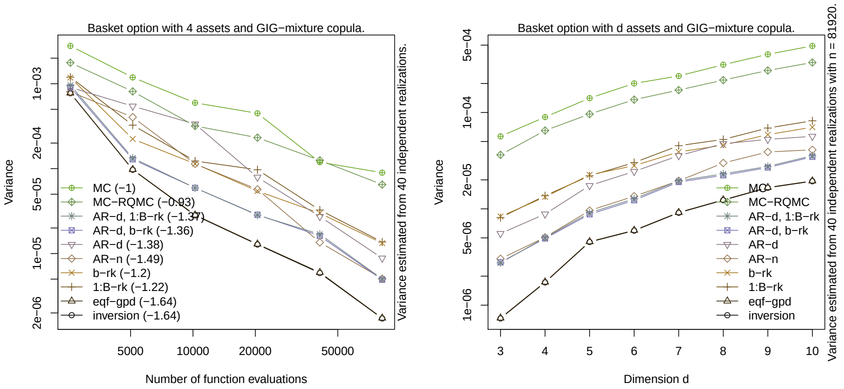

Fig. 8 Variances when estimating μ_{bskt} under a t copula with $\nu = 2.2$ dof, $r = 0.01$, $\sigma = 0.2$ (volatility for all stocks) as a function of n (left) and as a function of d (right).

Finally, we alter this example so that we end up with a model where the quantile function of W is indeed not available. To this end, we replace the t copula with a GIG-mixture copula. A random vector $\boldsymbol{X}$ has a GIG-mixture distribution if it follows the stochastic representation (9) with $W \sim \text{GIG}(\beta, \lambda)$. The marginal distribution functions F_j of X_j needed to compute the copula sample are not known, so we denote by $\hat{F}_j(x) = (n+1)^{-1} \sum_{i=1}^{n} \mathbf{1}_{\{X_{ij} \le x\}}$ the empirical distribution function of

X_j, and instead compute the pseudo observations $\boldsymbol{U}_i = (\hat{F}_1(X_{i1}), \ldots, \hat{F}_d(X_{id}))$ for $i = 1, \ldots, n$. Figure 9 shows estimated variances as a function of n (left) and the number of assets d (right). All RQMC based methods outperform MC and MC-RQMC. The remaining methods, with the exception of AR-d and AR-n, perform very similarly and give good convergence rates over all dimensions. Figure 10 shows CPU times needed to compute various estimators. All methods, with the exception of "eqf-gpd", take roughly the same time. Recall that with eqf-gpd, the idea is to estimate the quantile function Q once, and then use it as a true quantile function for all subsequent simulations.

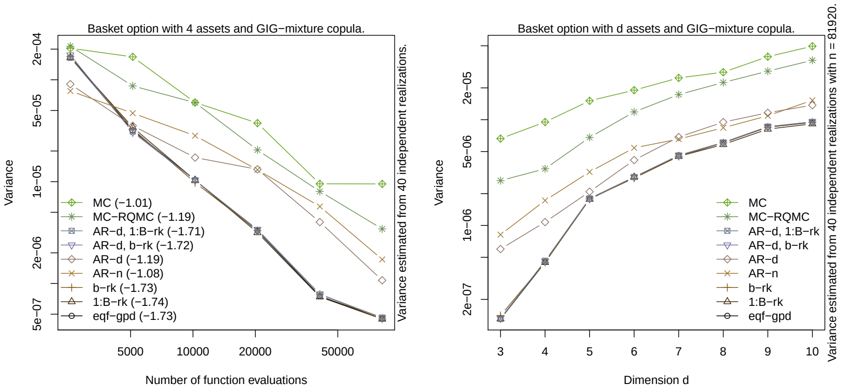

Fig. 9 Variances when estimating μ_{bskt} under a GIG mixture copula with $\lambda = 0.5$, $\beta = 0.3$, $r = 0.01$, $\sigma = 0.2$ (volatility for all stocks) as a function of n (left) and as a function of d (right).

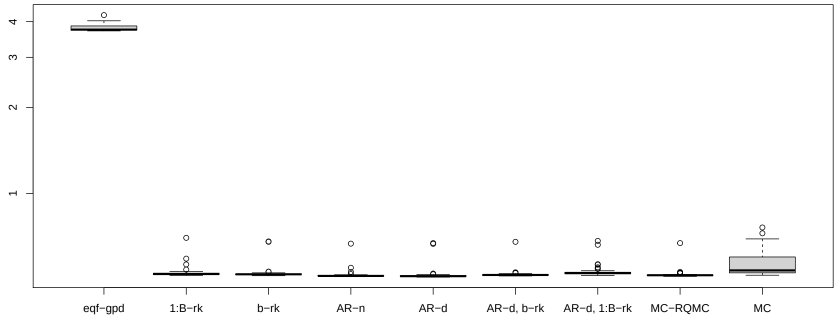

Fig. 10 Boxplot of CPU times when estimating μ_{bskt} under a GIG mixture copula with $\lambda = 0.5$, $\beta = 0.3$, $r = 0.01$, $\sigma = 0.2$ (volatility for all stocks).

5 Conclusion

We explored the question how RQMC can be applied to estimate $\mu = \mathbb{E}(g(\boldsymbol{Y}, W))$ when all components but one can be sampled via inversion, and the remaining one W by calling a NRVG only. Our proposed algorithms in the black box setting were motivated by the fact that RQMC works best when combined with inversion, so that our methods aim at mimicking this observation by exploiting the sample to estimate the quantile function. In Section 3, we assumed the existence of an AR algorithm, and motivated an AR-d algorithm that samples along the coordinates rather than moving along the sequence. Our numerical results indicate that RQMC can still provide a substantial variance reduction when combined with a NRVG. In particular we saw that the re-ordering methods outperform MC-RQMC (where we merely combine RQMC with pseudo-random sampling of W). Furthermore, we saw that moving along the coordinates as we do in AR-d can give better results than the previously proposed AR-n methods. With the methods in this paper at hand, we could extend the algorithms in [7] and [8] to estimate various quantities related to multivariate normal variance mixture distributions, such as the distribution function. Furthermore, we plan to address some questions of computational nature, such as exploring efficient implementations of AR-d based on point sets that are easily extensible in the number of coordinates, such as Korobov rules based on well-chosen generators a; see [14]. Finally, this paper mostly focused on numerical comparisons of different RQMC-based algorithms based on digitally shifted Sobol' sequences. In the near future we plan to study settings under which it might be possible to obtain theoretical results demonstrating the superiority of our proposed RQMC-based methods (perhaps based on scramblings rather than shifts) over Monte Carlo.

References

1. Cheng, R.: The generation of gamma variables with non-integral shape parameter. Journal of the Royal Statistical Society: Series C (Applied Statistics) **26**(1), 71–75 (1977)
2. Demarta, S., McNeil, A.: The t copula and related copulas. International statistical review **73**(1), 111–129 (2005). DOI 10.1111/j.1751-5823.2005.tb00254.x
3. Devroye, L.: Non-Uniform Random Variate Generation. Springer New York (1986). DOI 10.1007/978-1-4613-8643-8
4. Embrechts, P., Klüppelberg, C., Mikosch, T.: Modelling extremal events. British actuarial journal **5**(2), 465–465 (1999)
5. Flury, B.: Acceptance–rejection sampling made easy. SIAM Review **32**(3), 474–476 (1990)
6. Glasserman, P.: Monte Carlo methods in financial engineering, vol. 53. Springer Science & Business Media (2013)
7. Hintz, E., Hofert, M., Lemieux, C.: Grouped normal variance mixtures. Risks **8**(4), 103 (2020). DOI 10.3390/risks8040103
8. Hintz, E., Hofert, M., Lemieux, C.: Normal variance mixtures: Distribution, density and parameter estimation. Computational Statistics and Data Analysis **157C**, 107175 (2021). DOI 10.1016/j.csda.2021.107175
9. Hofert, M., Lemieux, C.: qrng: (Randomized) Quasi-Random Number Generators (2019). URL https://CRAN.R-project.org/package=qrng. R package version 0.0-7

10. Hörmann, W., Leydold, J.: Generating generalized inverse gaussian random variates. Statistics and Computing **24**(4), 547–557 (2014). DOI 10.1007/s11222-013-9387-3
11. Kundu, D., Gupta, R.: A convenient way of generating gamma random variables using generalized exponential distribution. Computational Statistics and Data Analysis **51**(6), 2796–2802 (2007). DOI 10.1016/j.csda.2006.09.037
12. L'Ecuyer, P.: Quasi-Monte Carlo methods in finance. In: Proceedings of the 2004 Winter Simulation Conference, vol. 2, pp. 1645–1655. IEEE (2004)
13. L'Ecuyer, P., Lécot, C., Tuffin, B.: A randomized quasi-Monte Carlo simulation method for Markov chains. Operations Research **56**(4), 958–975 (2008)
14. L'Ecuyer, P., Lemieux, C.: Variance reduction via lattice rules. Management Science **46**(9), 1214–1235 (2000)
15. L'Ecuyer, P., Lemieux, C.: Recent advances in randomized quasi-Monte Carlo methods. In: M. Dror, P. L'Ecuyer, F. Szidarovszki (eds.) Modeling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications, pp. 419–474. Kluwer Academic Publishers, Boston (2002)
16. L'Ecuyer, P., Munger, D., Lécot, C., Tuffin, B.: Sorting methods and convergence rates for array-RQMC: Some empirical comparisons. Mathematics and Computers in Simulation **143**, 191–201 (2018)
17. McNeil, A., Frey, R., Embrechts, P.: Quantitative Risk Management: Concepts, Techniques and Tools. Princeton University Press (2015). DOI 10.1007/s10687-017-0286-4
18. Moskowitz, B., Caflisch, R.: Smoothness and dimension reduction in quasi-Monte Carlo methods. Mathematical and Computer Modelling **23**(8-9), 37–54 (1996). DOI 10.1016/0895-7177(96)00038-6
19. Nakayama, M., Kaplan, Z.T., L'Ecuyer, P., Tuffin, B.: Quantile estimation via a combination of conditional Monte Carlo and randomized quasi-Monte Carlo. In: Proceedings of the 2020 Winter Simulation Conference (2020)
20. Nguyen, N., Ökten, G.: The acceptance-rejection method for low-discrepancy sequences. Monte Carlo Methods and Applications **22**(2), 133–148 (2016). DOI 10.1515/mcma-2016-0104
21. Rosenblatt, M.: Remarks on a multivariate transformation. The Annals of Mathematical Statistics **23**(3), 470–472 (1952). DOI 10.1214/aoms/1177729394
22. Wang, X.: Improving the rejection sampling method in quasi-Monte Carlo methods. Journal of computational and applied Mathematics **114**(2), 231–246 (2000). DOI 10.1016/S0377-0427(99)00194-6
23. Zhu, H., Dick, J.: Discrepancy bounds for deterministic acceptance-rejection samplers. Electronic Journal of Statistics **8**(1), 678–707 (2014). DOI 10.1214/14-EJS898