ICMA Centre

Henley Business School
UNIVERSITY OF READING

Discussion Paper

Monte Carlo Approximate Tensor Moment Simulations

September 2014

Juan C Arismendi Z
ICMA Centre, Henley Business School, University of Reading

Herbert Kimura
Faculty of Economics, Business and Accountancy, University of Brasilia

Discussion Paper Number:
ICM-2014-08

Accredited by Association of MBAs
EFMD EQUIS ACCREDITED
AACSB ACCREDITED

The aim of this series is to disseminate new research of academic distinction. Papers are preliminary drafts, circulated to stimulate discussion and critical comment.

Henley Business School is triple accredited and home to over 100 academic faculty who undertake research in a wide range of fields from ethics and finance to international business and marketing.

admin@icmacentre.ac.uk

www.icmacentre.ac.uk

Monte Carlo Approximate Tensor Moment Simulations

J. C. ARISMENDI*‡ and HERBERT KIMURA‡

* *ICMA Centre – Henley Business School, University of Reading, Whiteknights, Reading RG6 6BA, UK.*

‡ *Faculty of Economics, Business and Accountancy – University of Brasilia, Campus Darcy Ribeiro, Brasilia, Brazil.*

Abstract

An algorithm to generate samples with approximate first-, second-, and third-order moments is presented extending the Cholesky matrix decomposition to a Cholesky tensor decomposition of an arbitrary order. The tensor decomposition of the first-, second-, and third-order objective moments generates a non-linear system of equations. The algorithm solves these equations by numerical methods. The results show that the optimisation algorithm delivers samples with an approximate error of 0.1%–4% between the components of the objective and the sample moments. An application for sensitivity analysis of portfolio risk assessment with Value-at-Risk (VaR) is provided. A comparison with previous methods available in the literature suggests that methodology proposed reduces the error of the objective moments in the generated samples. [1]

Key Words: Monte Carlo Simulation, Higher-order Moments, Exact Moments Simulation, Stress-testing

In this research we consider simulation methods, and we simulate random processes with predetermined moments and cumulants. The main problem is to simulate samples where the intended statistics of the samples must have a desired moment value, such as the mean, or the second-order matrix. The moments of a distribution can affect the shape and even all the characteristics of the distribution; for example, in the case of the normal distribution, we need only the first- and second-order moments to determine the entire shape of the distribution. The statistical moments are defined from the moment generating function (as well as the central moments, or cumulants).

We produce a Monte Carlo method that generates samples with approximate first-, second-, and third-order moments for the multivariate case. The moments and cumulants are tensors, and for this reason we introduce some concepts of tensor spaces. Problems with the constraint on the first- and second-order moment have a solution. Ledermann et al. (2011), using *random*

Correspondence Address: ICMA Centre – Henley Business School,University of Reading, Whiteknights, Reading RG6 6BA, UK . Email: j.arismendi@icmacentre.ac.uk

[1] JEL Classification: C14, C15, G32.

orthogonal matrices, has found a solution of the problem of Monte Carlo simulations with exact skewness and kurtosis, using the measures defined by Mardia (1970); however, Mardia's measures report similar skewness values for different elliptical distributions. As a result, Ledermman's methodology will produce the same simulations for elliptical distributions that have equal skewness but different third-order moments. Our research produces simulations not with an skewness objective value, but with third-order objective moments.

Let X be a multivariate random variable of dimension p whose first two moments are known. The problem of generating samples from X with exact first- and second-order moments has been addressed by several authors. This problem consists in generating a matrix of m samples $\tilde{X}$ from X, where $\tilde{X}(i)$, $i \in \{1, \ldots, m\}$, is a vector of dimension p. The replicas are generated with a pseudo-random algorithm, and then a function is applied to give the desired sample moments. The samples are pairs $\{\tilde{X}(i), p(i)\}$, where $p(i)$ is the probability density associated with the sample.

Meucci (2009) classifies the methods used to generate simulations of random variables with exact mean-covariance in two groups: methods that constrain the probabilities $\{p(i)\}$, and methods that constrain the scenarios $\{\tilde{X}(i)\}$, $i \in \{1, \ldots, m\}$. In the former are the methods of Avellaneda (1998), D'Amico et al. (2003), and Glasserman and Yu (2005). In the latter are the Wedderburn (1975), Cheng (1985), Li (1992), and Ledermann et al. (2011), where the *spectral decomposition* and the *QR decomposition* methodologies are the most-used. The Meucci (2009) method constrains the scenarios and is based on the generation of matrix $\mathbf{B}$, such that $\mathbf{B}\tilde{X}$ will have the desired moments. This method offers an improvement in the performance of the size of the matrix needed to create the samples. Meucci applied this algorithm to simulate the correlation matrix of a portfolio of plain vanilla options. He offered the method as an alternative for stress-testing but he did not produce any empirical test to measure the benefits of using this exact method.

The structure of this paper is as follows: Section 1 introduces the tensor notation; Section 2 reviews the *spectral decomposition* method for exact mean-covariance (first- and second-order moments) simulations. Section 3 develops the algorithm proposed to solve the problem of Monte Carlo approximate first-, second-, and third-order moments simulations and in Section 4 a numerical approximation is provided. In Section 5 we present some numerical results. Section 6 presents an application to portfolio risks assessment. In Section 7 some concluding remarks are presented and possible extensions suggested.

1. Definitions of higher-order tensor moments

The seminal book of Kendall (1947), and the later book of McCullagh (1987) are the main references for the use of tensors in statistics. In fact, the mathematical definition of a moment of n-th order, is a tensor of n-th order. Kendall's book made an introduction to tensor cumulants and tensor moments, while McCullagh described all the mathematical theory behind tensors and their use in statistics. In the following sections we give a brief introduction to tensor calculus, as we will use it to describe tensor decompositions approach.

1.1 The summation notation

We use the exponent as an indicator of the tensor's component.

Definition 1.1: Let X and a be vectors of dimension p with components $X(i)$ and $a(i)$ for $i \in \{1, \ldots, p\}$, respectively. We are going to use the summation convention as it is the appropriate notation for working with tensors. We define the object $a_i X_i$ as:

$$a_i X_i \equiv \sum_{i=1}^{p} a(i)X(i), \tag{1}$$

where every common index, such as i in (1), denotes a summation of the components of tensors.

The vectors X and a are considered first-order tensors, and (1) is considered a tensor of zero-th order. If we refer to the vector and matrix notation, let $\mathbf{x}$ and $\mathbf{a}$ be two vectors of dimension p; an equivalent expression to (1) will be:

$$a_i X_i \equiv \mathbf{a}'\mathbf{x}.$$

Definition 1.2: Let a be a matrix of dimension $p \times p$, and $X \equiv \mathbf{x}$ as in (1). Denote using the vector notation, $\mathbf{A} \equiv a$. The matrix a is a tensor of second order. The equivalent summation notation of the vector expression $\mathbf{x}^T\mathbf{A}\mathbf{x}$ is:

$$a_{i,j} X_i X_j \equiv \sum_{i=1}^{p}\sum_{j=1}^{p} a(i,j)X(i)X(j). \tag{2}$$

Definition 1.3: Let a be a tensor of third order; the following tensor product will produce

a tensor of zero-th order, as the result of (1) and (2),

$$a_{i,j,k}X_iX_jX_k \equiv \sum_{i=1}^{p}\sum_{j=1}^{p}\sum_{k=1}^{p} a(i,j,k)X(i)X(j)X(k). \tag{3}$$

In (3), there is not an equivalent vector notation for the expression, and it is for that reason we use tensor notation for describing higher-order moments and cumulants.

1.2 Tensor moment and tensor cumulant definition

Definition 1.4: Let X be a random vector of dimension p. The characteristic function of X is defined as:

$$\psi(\xi, x) = \mathbb{E}\left[\exp\left(\xi_{l_1}X_{l_1}i\right)\right],$$

where $l_1 \in \{1,\ldots,p\}$, and ξ is a real-valued vector. The vector $x = (x(1),\ldots,x(p))$ appears after the calculation of the expected value of a function of the random variable X. This function can be expanded into the infinite series:

$$\begin{aligned}\psi(\xi, x) &= 1 + \xi_{l_1}m(1)_{l_1}i + \xi_{l_1}\xi_{l_2}m(2)_{l_1,l_2}i^2/2! + \xi_{l_1}\xi_{l_2}\xi_{l_3}m(3)_{l_1,l_2,l_3}i^3/3! + \ldots, \\ &= \sum_{j=1}^{n-1}\xi_{l_1}\ldots\xi_{l_j}m(j)_{l_1,\ldots,l_j}i^j/j! + o(\|\xi\|^n),\end{aligned} \tag{4}$$

which is convergent for small ξ. The coefficient of the series, $m(n) = \mathbb{E}(X(l_1)\ldots X(l_n))$, is denoted as the tensor moment of n-th order of X, where $l_1,\ldots,l_n \in \{1,\ldots,p\}$.

Definition 1.5: Calculate the $\log(\cdot)$ function of (4):

$$\log\psi(\xi, x) = \sum_{j=1}^{n-1}\xi_{l_1}\ldots\xi_{l_j}k(j)_{l_1,\ldots,l_j}i^j/j! + o(\|\xi\|^n).$$

The coefficient $k(n)_{l_1,\ldots,l_n} = \mathbb{E}\left[(X(l_1) - m(1;l_1))\ldots(X(l_n) - m(1;l_n))\right]$ is denoted the tensor cumulant of the n-th order of X, with $m(i;l_1,\ldots,l_j)$ the $l_1,\ldots,l_j$-th component of the i-th-order tensor moment.

The covariance matrix is a tensor cumulant of the second order. There is an equivalence between the first- and second-order tensor moments and tensor cumulants:

$$m(1) = k(1),$$

$$m(2)_{l_1,l_2} = k(2)_{l_1,l_2} + k(1)_{l_1}k(1)_{l_2}.$$

1.3 Tensor sample moments and tensor sample cumulants

Definition 1.6: Let $\tilde{X}$ be a matrix of dimension $m \times p$, of m samples of the random variable X of dimension p; we define the n-th-order tensor sample moment as:

$$M(n)_{l_1,\ldots,l_n} = m^{-1}\tilde{X}_{i,l_1}\ldots\tilde{X}_{i,l_n},$$

and the n-th-order tensor sample cumulant as:

$$K(n) = m^{-1}(\tilde{X}_{i,l_1} - \bar{X}_{i,l_1})\ldots(\tilde{X}_{i,l_n} - \bar{X}_{i,l_n}),$$

where $l_1,\ldots,l_n \in \{1,\ldots,p\}$ and $\bar{X}$ is a matrix of dimension $m \times p$ with the sample mean vector of $\tilde{X}$ repeated in every row.

1.4 Multivariate measures of skewness and kurtosis

The multivariate measures of skewness and kurtosis developed by Mardia (1970) are the standard measures in the literature. Fields like finance use these measures. In other fields like physical sciences, these are also the standard measures. In multivariate statistical analysis they represent the actual framework to measure deviations from normality. This is the concept that Mardia and other statisticians used to develop different multivariate skewness measures: determine the normality of a sample, or deviations from normality.

Mardia's (1970) skewness and kurtosis definition: Let $X = (X(1), X(2), \ldots, X(p))$ be a multivariate random vector. Let $\mu = (\mu(1), \ldots, \mu(p))$ denote the mean vector of X and $\mathbf{V}$ the covariance matrix. Denote by $Y = (X-\mu)'\mathbf{V}^{-1/2}$ the standardised vector. Let Z be a random vector with the same distribution as Y, but independent of Y. Mardia's skewness measure of X is:

$$\beta_1 = E\left[(Y'Z)\right]^3. \tag{5}$$

A fundamental property of a skewness measure is that it is invariant under non-singular transformations. Mardia's kurtosis measure is:

$$\beta_2 = E\left[(Y'Y)\right]^4. \tag{6}$$

The kurtosis measure is also invariant under non-singular transformations. Mardia (1970) shows an application where he tested the normality from two artificially generated samples: one generated from a symmetric distribution and the other from a skewed one, and the re-

sults confirm the applicability of these measures to recognise the deviations from the normal distribution in a sample. However, Mardia's measures were designed to measure third- and fourth-order moment deviations from the class of elliptical symmetric distributions, but were not designed to inform about the third- and fourth-order moment properties of elliptical symmetric distributions.

Besides being the standard measure for multivariate skewness and kurtosis, Mardia's measures report the same numeric value for some elliptical distributions of different shapes. Therefore, we develop an algorithm that considers exact tensor moment simulations, and not only exact skewness and kurtosis simulations.

2. Exact mean-covariance simulations

In the first part of this section the methods to simulate random vectors with exact covariance are covered, with a detailed explanation of one of the most common methods: *spectral decomposition*. None of the methods developed until now uses the definition of higher-order moment as a tensor.

Most of the statistical and mathematical concepts behind each of the methods for exact mean-covariance moment simulations are similar. The two most-used methods for the exact moment sampling that constrain scenarios are the *spectral decomposition* method, and the *QR decomposition* method. These two are the best-known solutions in the academic literature. Both methods use orthogonal projections. An excellent description of the first method is in Ledermann et al. (2011). Another reference for these methods can be found in Jackel (2002). The *QR decomposition* method uses a similar orthogonal decomposition as the *spectral decomposition* method, therefore we briefly describe the *spectral decomposition* method for introducing the tensor moment notation. Henceforward, to simplify notation we refer to 'tensor moment' when using the expression 'moment'.

2.1 Spectral decomposition method

Define X as a random vector of dimension p, and $\tilde{X}$ as a matrix of dimension $m \times p$, of the m samples of X. From now on, vectors and matrices will be denoted as tensors, and **bold** notation used in vector notation will be abandoned. The matrix $\tilde{X}$ has components $\tilde{X}_{i,r}, i \in \{1, \ldots, m\}$, $r \in \{1, \ldots, p\}$. Define K as the second sample cumulant of $\tilde{X}$. From

McCullagh (1987), it is known that $K_{r,s} = n^{-1}\phi_{i,j}\tilde{X}_{i,r}\tilde{X}_{j,s}$, but can also be written as:

$$K_{r,s} = n^{-1}(\tilde{X}_{i,r} - \bar{X}_{i,r})(\tilde{X}_{i,s} - \bar{X}_{i,s}),$$

where $\bar{X}$ is the matrix of $m \times p$ with the sample mean of $\tilde{X}$ repeated on every row, and $s \in \{1, \ldots, p\}$, $j \in \{1, \ldots, m\}$. The standard Monte Carlo simulation method produce samples using the following equation:

$$\tilde{X}_{i,r} = \bar{X}_{i,r} + \tilde{Z}_{i,s}A_{s,r},$$

where matrix A satisfies:

$$A_{r,t}A_{s,t} = K_{r,s},$$

for $t \in \{1, \ldots, p\}$, and $\tilde{Z}$ is a sample from a multivariate standard normal. We can see that the resulting sample second cumulant of $\tilde{X}$ is approximately the covariance or cumulant of second order of X. This approximation has two sources of errors, one is from the mean and the other from the covariance of the sample. The first one can be eliminated if we subtract the sample mean of $\tilde{Z}$:

$$\tilde{X}_{i,r} = \bar{X}_{i,r} + (\tilde{Z}_{i,s} - \bar{Z}_{i,s})A_{s,r},$$

where $\bar{Z}$ is the matrix with the mean value of vector Z repeated on every row. For the second source of error we apply a *spectral decomposition* method to find a matrix of samples $\tilde{W}$ of dimension $m \times p$, using a projection of the original multivariate normal Z, and impose the constraint of orthogonality over this matrix, i.e., $\tilde{W}_{i,r}\tilde{W}_{i,s} = I_{r,s}$, where I is the identity matrix.[2] In this case we consider square identity matrices.

Define the first sample without transformations as:

$$\tilde{Y}_{i,r} = \tilde{Z}_{i,r} - \bar{Z}_{i,r},$$

and let the matrix $\tilde{W}$ have the following form:

$$\tilde{W}_{i,r} = \tilde{Y}_{i,r}Q_{r,s}\Lambda_{s,t}^{1/2},$$

where the matrix Λ is a diagonal matrix with the eigenvalues of K, and Q is the eigenvector

[2] $I_{r,s} = 1$ if $r = s$, zero otherwise.

matrix of K; both Λ and Q are the result from the *spectral decomposition* of K as:

$$K_{r,s} = Q_{r,s}\Lambda_{r,s}Q^{-1}_{r,s}.$$

It can be shown that the sample:

$$\tilde{X}_{i,r} = \bar{X}_{i,r} + \sqrt{n}\tilde{W}_{i,s}A_{s,r},$$

will have as a second-order cumulant the matrix K.

Resuming this method, the solution is to express the sample $\tilde{X}$ as a function of the sample cumulant K and a random multivariate normal matrix:

$$\tilde{X} = f(K, \tilde{Z}).$$

3. Exact first-, second-, and third-order moment simulations

Let $\tilde{X}$ be a matrix with m samples of a random variable X. Suppose X has mean vector zero, $E(X) = 0$. Define $M(1)$ as the sample first-order moment of this multivariate random variable:

$$M(1)_r = m^{-1}\tilde{X}_{r,i}. \tag{7}$$

In this case, although there are no common indices on the right-hand side of (7) there must be a sum over the index i to reduce the second-order tensor $\tilde{X}$ to the first-order tensor $M(1)$. Define $M(2)$ as the second-order sample moment:

$$M(2)_{r,s} = m^{-1}\tilde{X}_{r,i}\tilde{X}_{s,i},$$

and $M(3)$ as the third-order sample moment:

$$M(3)_{r,s,t} = m^{-1}\tilde{X}_{r,i}\tilde{X}_{s,i}\tilde{X}_{t,i}.$$

Let $\tilde{Y}$ be a random sample of dimension $m \times p$ of Y. By construction, suppose we want to generate $\tilde{Y}$, with a first-order sample moment equal to the tensor $\widehat{M}(1)$, a second-order sample moment equal to the tensor $\widehat{M}(2)$, and a third-order sample moment equal to the tensor $\widehat{M}(3)$. We use a numerical approach to solve this problem, defining $M(1)_r - \widehat{M}(1)_r = 0$, $M(2)_{r,s} - \widehat{M}(2)_{r,s} = 0$, and $M(3)_{r,s,t} - \widehat{M}(3)_{r,s,t} = 0$ as the set of non-linear equations to be solved. The first set of equations, is solved by adding the mean value desired $M(1)$ to the

sample, as the second- and third-order moments are affine invariant.

Definition 3.1: Define by construction a square matrix A with dimension $p \times p$, with second-order sample moment equal to $\widehat{M}(2)$:

$$\widehat{M}(2)_{r,s} = m^{-1} A_{r,i} A_{s,i}, \tag{8}$$

and third-order sample moment equal to $\widehat{M}(3)$:

$$\widehat{M}(3)_{r,s,t} = m^{-1} A_{r,i} A_{s,i} A_{t,i}. \tag{9}$$

This process is defined as a Cholesky tensor decomposition for second- and third-order tensors.

It is important to study the conditions for the existence of the matrix A. It will be important in the future to study the conditions of uniqueness. Let us study the existence of the third-order tensor decomposition (9) of the case $p = 2$. Let A be a unknown matrix, and $\widehat{M}(3)$ a third-order objective tensor given by the problem:

$$A = \begin{pmatrix} a(1,1)\ a(1,2) \\ a(2,1)\ a(2,2) \end{pmatrix}, \widehat{M}(3) = \left(\begin{array}{cc|cc} \widehat{M}(3;1,1,1) & \widehat{M}(3;1,1,2) & \widehat{M}(3;2,1,1) & \widehat{M}(3;2,1,2) \\ \widehat{M}(3;1,2,1) & \widehat{M}(3;1,2,2) & \widehat{M}(3;2,2,1) & \widehat{M}(3;2,2,2) \end{array}\right).$$

The tensor decomposition generates a set of four non-linear equations. Solving the set of equations leads us to the final non-linear equation:

$$\begin{aligned} b^{1/3} c^2 + a(2,2) c - \widehat{M}(3;1,2,2) &= 0, \\ b &= \left(\widehat{M}(3;1,1,1) - \widehat{M}(3;2,2,2) + a(2,2)\right), \\ c &= \left(\frac{-b \pm \sqrt{b^2 + 4a(2,2)\widehat{M}(3;1,1,2)}}{2a(2,2)}\right), \end{aligned} \tag{10}$$

where the solution of the unknown variable $a(2,2)$ in (10) will provide the remaining values for A:

$$\begin{aligned} a(2,1) = a(1,2) &= \left(\widehat{M}(3;2,2,2) - a(2,2)^3\right)^{1/3}, \\ a(1,1) &= \left(\widehat{M}(3;1,1,1) - a(2,1)^3\right)^{1/3}. \end{aligned}$$

The non-linear equation (10) could have a solution in $\mathbb{R}$ if and only if $b^2 + 4a(2,2)\widehat{M}(3;1,1,2) \geq 0$, and even in that case we can not easily confirm that (10) has a solution. In higher dimensions ($p > 2$), the non-linear equations will be even more challenging to solve than (10), and for this reason we use a numerical method to find this decomposition.

3.1 Exact Cholesky tensor decomposition

We generate a random sample $\tilde{Z}$ from the multivariate standard normal variable Z. This sample has dimension $m \times p$. Now the product $\tilde{Z}_{i,r}A_{r,s}$ will have dimension $m \times p$, and the second-order sample moment will be equal to:

$$\widetilde{M}(2)_{r,s} = m^{-1}(\tilde{Z}A)_{i,r}(\tilde{Z}A)_{i,s},$$

and the third-order sample moment equal to:

$$\widetilde{M}(3)_{r,s,t} = m^{-1}(\tilde{Z}A)_{i,r}(\tilde{Z}A)_{i,s}(\tilde{Z}A)_{i,t}.$$

The expected values of these sample statistics are:

$$E\left(\widetilde{M}(2)_{r,s}\right) = \widehat{M}(2)_{r,s},$$
$$E\left(\widetilde{M}(3)_{r,s,t}\right) = \widehat{M}(3)_{r,s,t}.$$

This simulation will have on average the desired second- and third-order moment $\widehat{M}(2)_{r,s}$ and $\widehat{M}(3)_{r,s,t}$; however, to have an exact simulation we need to apply an orthogonal projection T to $\tilde{Z}$. By construction, we want after the application of T produce the identity[3] tensor δ_{r_1,r_2} of the second-order sample moment of $\tilde{Z}T$ multiplied by m:

$$(\tilde{Z}T)_{i,r_1}(\tilde{Z}T)_{i,r_2} = \delta_{r_1,r_2}, \tag{11}$$

and the identity[4] tensor δ_{r_1,r_2,r_3}, of the third-order sample moment of $\tilde{Z}T$ multiplied by m:

$$(\tilde{Z}T)_{i,r_1}(\tilde{Z}T)_{i,r_2}(\tilde{Z}T)_{i,r_3} = \delta_{r_1,r_2,r_3}, \tag{12}$$

with $r_1, r_2, r_3 \in \{1, \ldots, p\}$. The reason for this application is to generate an orthogonal projection of $\tilde{Z}$.

Proposition 3.2: *Let $\widehat{M}(1)$, $\widehat{M}(2)$, and $\widehat{M}(3)$ be the second-, and third-order objective moments. Assume there exists a matrix A, a result of the Cholesky tensor decomposition of $\widehat{M}(2)$ and $\widehat{M}(3)$, such that (8) and (9) hold. Define $\tilde{Y}$ as the tensor product:*

$$\tilde{Y}_{i,r} = \widehat{M}(1) + \tilde{Z}_{i,s}T_{s,t}A_{t,r},$$

[3] $\delta_{r_1,r_2} = 1$ if $r_1 = r_2$, zero otherwise.
[4] $\delta_{r_1,r_2,r_3} = 1$ if $r_1 = r_2 = r_3$, zero otherwise.

where $\tilde{Z}$ is a matrix of m samples of Z a multivariate standard normal variable. Let T be a an application such that (11) and (12) hold. Then, the sample first-, second-, and third order moments of $\tilde{Y}$ are $\widehat{M}(1)$, $\widehat{M}(2)$, and $\widehat{M}(3)$.

Proof: Let $s_1, s_2, s_3, t_1, t_2, t_3 \in \{1, \ldots, p\}$. We calculate the sample second-order moment of $\tilde{Y}$:

$$\begin{aligned} m^{-1}\tilde{Y}_{i,r}\tilde{Y}_{i,s} &= m^{-1}(ZTA)_{i,r}(ZTA)_{i,s} \\ &= m^{-1}(Z_{i,r_1}T_{r_1,r_2}A_{r_2,r})(Z_{i,s_1}T_{s_1,r_2}A_{r_2,s}) \\ &= m^{-1}A_{r_2,r}A_{r_2,s}(Z_{i,r_1}T_{r_1,r_2}Z_{i,s_1}T_{s_1,r_2}) \\ &= m^{-1}A_{r_2,r}A_{r_2,s}(ZT)_{i,r_2}(ZT)_{i,r_2} \\ &= m^{-1}A_{r_2,r}A_{r_2,s}\delta_{r_2,r_2} \\ &= m^{-1}A_{r_2,r}A_{r_2,s} = M(2)_{r,s}, \end{aligned}$$

and the sample third-order moment of $\tilde{Y}$:

$$\begin{aligned} m^{-1}\tilde{Y}_{r,i}\tilde{Y}_{s,i}\tilde{Y}_{t,i} &= m^{-1}(ZTA)_{i,r}(ZTA)_{i,s}(ZTA)_{i,t} \\ &= m^{-1}(Z_{i,r_1}T_{r_1,r_2}A_{r_2,r})(Z_{i,s_1}T_{s_1,r_2}A_{r_2,s})(Z_{i,t_1}T_{t_1,r_2}A_{r_2,t}) \\ &= m^{-1}A_{r_2,r}A_{r_2,s}A_{r_2,t}(Z_{i,r_1}T_{r_1,r_2}Z_{i,s_1}T_{s_1,r_2}Z_{i,t_1}T_{t_1,r_2}) \\ &= m^{-1}A_{r_2,r}A_{r_2,s}A_{r_2,t}(ZT)_{i,r_2}(ZT)_{i,r_2}(ZT)_{i,r_2} \\ &= m^{-1}A_{r_2,r}A_{r_2,s}A_{r_2,t}\delta_{r_2,r_2,r_2} \\ &= m^{-1}A_{r_2,r}A_{r_2,s}A_{r_2,t} = M(3)_{r,s,t}. \end{aligned}$$

□

It has been proved by construction that the sample second- and third-order moments of $\tilde{Y}$ are exactly $\widehat{M}(2)_{r,s}$ and $\widehat{M}(3)_{r,s,t}$, assuming we find matrices A and T.

To generate samples with the second and the third exact moments, we construct the matrix A such that both conditions are fulfilled:

$$m^{-1}\tilde{Y}_{r,i}\tilde{Y}_{s,i} = M(2)_{r,s},$$

$$m^{-1}\tilde{Y}_{r,i}\tilde{Y}_{s,i}\tilde{Y}_{t,i} = M(3)_{r,s,t}.$$

4. Numerical method for random sampling with approximate first-, second-, and third-order moments

In this section we find numerical approximations of matrices A and T. For matrix A we have the following set of equations derived from Section 3 construction:

$$\begin{aligned} & p^{-1}A_{r,t}A_{s,t} = \widehat{M}(2)_{r,s} \\ \Longrightarrow\ & p^{-1}\sum_{i=1}^{p} A(r,t)A(s,t) = \widehat{M}(2;r,s) \end{aligned}$$

$$\Longrightarrow \begin{cases} p^{-1}\left(A(1,1)A(1,1) + A(2,1)A(2,1) + \cdots A(p,1)A(p,1)\right) = \widehat{M}(2;1,1) \\ p^{-1}\left(A(1,1)A(1,2) + A(2,1)A(2,2) + \cdots A(p,1)A(p,2)\right) = \widehat{M}(2;1,2) \\ \qquad\qquad\qquad\qquad \vdots \\ p^{-1}\left(A(1,p)A(1,p) + A(2,p)A(2,p) + \cdots A(p,p)A(p,p)\right) = \widehat{M}(2;p,p) \end{cases}, \tag{13}$$

and,

$$\begin{aligned} & p^{-1}A_{r,u}A_{s,u}A_{t,u} = \widehat{M}(3)_{r,s,t} \\ \Longrightarrow\ & p^{-1}\sum_{u=1}^{p} A(r,u)A(s,u)A(t,u) = \widehat{M}(3;r,s,t) \end{aligned}$$

$$\Longrightarrow \begin{cases} p^{-1}\left(A(1,1)A(1,1)A(1,1) + \cdots + A(p,1)A(p,1)A(p,1)\right) = \widehat{M}(3;1,1,1) \\ p^{-1}\left(A(1,1)A(1,2)A(1,1) + \cdots + A(p,1)A(p,2)A(p,1)\right) = \widehat{M}(3;1,2,1) \\ \qquad\qquad\qquad\qquad \vdots \\ p^{-1}\left(A(1,p)A(1,p)A(1,p) + \cdots + A(p,p)A(p,p)A(p,p)\right) = \widehat{M}(3;p,p,p) \end{cases}, \tag{14}$$

where $u \in \{1,\ldots,p\}$. To find matrix T in the third-order moments case,[5] we apply a similar algorithm to solve (13) and (14). By construction, we have the following system of non-linear equations:

$$\begin{aligned} & m^{-1}(ZT)_{i,r_1}(ZT)_{i,r_2}(ZT)_{i,r_3} = \delta_{r_1,r_2,r_3} \\ \Longrightarrow\ & m^{-1}\sum_{i=1}^{m}(ZT)(2;i,r_1)(ZT)(2;i,r_2)(ZT)(2;i,r_3) = \delta(r_1,r_2,r_3) \end{aligned}$$

[5]For the second-order moments simulations, the matrix T is found with a similar approach.

$$\Longrightarrow \begin{cases} m^{-1}\Big((ZT)(2;1,1)(ZT)(2;1,1)(ZT)(2;1,1)+\cdots \\ \quad +(ZT)(2;m,1)(ZT(2;m,1)(ZT)(2;m,1)\Big)=\delta_{1,1,1}=1 \\ m^{-1}\Big((ZT)(2;1,1)(ZT)(2;1,2)(ZT)(2;1,1)+\cdots \\ +(ZT)(2;m,1)(ZT)(2;m,2)(ZT)(2;m,1)\Big)=\delta(1,2,1)=0 \\ m^{-1}\Big((ZT)(2;1,p)(ZT)(2;1,p)(ZT)(2;1,p)+\cdots \\ \quad +(ZT)(2;m,p)(ZT)(2;m,p)(ZT)(2;m,p)\Big)=\delta(p,p,p)=1 \end{cases}, \tag{15}$$

plus the set of non-linear equations originated from:

$$(ZT)_{i,r_1}(ZT)_{i,r_2}=\delta_{r_1,r_2}. \tag{16}$$

The joint non-linear system of equations (13) and (14) has $(p\times p)+(p\times p\times p)=p^2+p^3$ equations. By symmetry $\widehat{M}(2;r,s)=\widehat{M}(2;s,r)$ and $\widehat{M}(3;r,r,s)=\widehat{M}(3;r,s,r)=\widehat{M}(3;s,r,r)$. To calculate the total number of equations we use combinatorics. The total number of equations is only a multiset of 3 elements from p possible elements:

$$\left(\binom{p}{2}\right)+\left(\binom{p}{3}\right)=\binom{p+1}{p-1}+\binom{p+2}{p-1}=\frac{(p+1)!}{(p-1)!2!}+\frac{(p+2)!}{(p-1)!3!}.$$

For example, for $p=2$ we have 7 equations, for $p=3$ we have 16 equations. The number of unknowns is $(p+1)p/2$. For $p=2$ there are 3 unknowns, less than the number of equations. A Cholesky tensor decomposition A that solves simultaneously (13) and (14) is not feasible, but just for trivial cases ($\tilde{Y}=A,\tilde{Y}$ of dimension $p\times p$). Additionally, the systems of equations (15) and (16) will have $(p+1)!/((p-1)!2!)+(p+2)!/((p-1)!3!)$ equations with just $(p+1)p/2$ unknowns. Therefore, we have to propose a two-stage approximation.

Define a first-order tensor B that contains all the non-linear equations of (15) and (16),

$$B_u=m^{-1}(ZT)_{u,r_1}(ZT)_{u,r_2}-\delta_{r_1,r_2},$$

for $u=\{1,\ldots,(p+1)!/((p-1)!2!)\}$, and,

$$B_u=m^{-1}(ZT)_{u,r_1}(ZT)_{u,r_2}(ZT)_{u,r_3}-\delta_{r_1,r_2,r_3}, \tag{17}$$

for $u=\{(p+1)!/((p-1)!2!)+1,\ldots,(p+1)!/((p-1)!2!)+(p+2)!/((p-1)!3!)\}$. The following

minimisation is proposed as a solution for the systems of equations (15) and (16):

$$\min_{T} \|B\|. \tag{18}$$

Denote by C a first-order tensor that contains all the non-linear equations of (13) and (14), i.e.,

$$C_u = p^{-1} A_{r,u} A_{s,u} - \widehat{M}(2)_{r,s},$$

for $u = \{1, \ldots, (p+1)!/((p-1)!2!)\}$, and,

$$C_u = p^{-1} A_{r,t} A_{s,t} A_{t,u} - \widehat{M}(3)_{r,s,t}, \tag{19}$$

for $u = \{(p+1)!/((p-1)!2!)+1, \ldots, (p+1)!/((p-1)!2!)+(p+2)!/((p-1)!3!)\}$. The following minimisation is proposed as a solution for the systems of equations (13) and (14):

$$\min_{A} \|C\| \tag{20}$$

The solution B^* of the minimisation problem (18), and the solution C^* of the minimisation problem (20) are proposed as the optimal approximation of Cholesky tensor decompositions, and they are used for the approximate first-, second-, and third-order moments simulations. We solve this system of non-linear equations using the MATLAB optimisation toolbox.

4.1 Relationship between approximate moment simulations and multi-linear singular value decomposition

The *spectral decomposition* method for exact covariance simulation is based on singular value decomposition (SVD). An approach to solve the exact simulation problem with a tensor is to find an equivalent definition in tensor calculus of SVD. The two most important tensor decomposition methods are *CANDECOMP/PARAFAC* and the *TUCKER* decomposition. The *CANDECOMP/PARAFAC* tensor decomposition is based on the idea of expressing the N-th-order tensor as the sum of finite rank-one tensor. In later part of this section we define rank-one tensors. This concept is intuitively similar to the rank of a matrix. Then decomposing a tensor into several rank-one tensors is similar to decomposing a matrix into several vectors, as SVD does. The *TUCKER* decomposition is a high-order tensor form of matrix principal component analysis (PCA). In Lathauwer et al. (1994) and Lathauwer et al. (2000) a generalisation of SVD for tensors is developed, termed *multi-linear singular value decompo-*

sition (MSVD). This MSVD is based on the the *TUCKER* decomposition. They highlight the expansion of tensors in the development of higher-order statistics, specially in higher-order moments and cumulants. On the other hand, SVD is one of the most useful methods of linear algebra. For this reason, Lathauwer et al. (2000) derives the MSVD using matrix based notation, instead of using the summation notation of physics. The *TUCKER* decomposition consists of decomposing a third-order tensor $A(3)$ into 4 components:

$$A(3)_{i_1,i_2,i_3} = s(3)_{j_1,j_2,j_3} u^{(1)}_{i_1,j_1} u^{(2)}_{i_2,j_2} u^{(3)}_{i_3,j_3}, \tag{21}$$

where $j_1 \in \{1,\ldots,I_1\}.j_2 \in \{1,\ldots,I_2\}, j_3 \in \{1,\ldots,I_3\}$, $u^{(1)}_{i_1,j_1} u^{(2)}_{i_2,j_2} u^{(3)}_{i_3,j_3}$ are the entries of three orthogonal matrices, and $s(3)_{j_1,j_2,j_3}$ is an orthogonal tensor, i.e., $\sum_{i_1,i_2} s(3;i_1,i_2,\alpha)s(3;i_1,i_2,\beta) = \sum_{i_1,i_3} s(3;i_1,\alpha,i_3)s(3;i_1,\beta,i_3) = \sum_{i_2,i_3} s(2\alpha,i_2,i_3)s(3;\beta,i_2,i_3) = 0$ for $\alpha \neq \beta$. In this section we use the **bold** notation for matrices for preserving the notation used by Lathauwer et al. (2000).

Lathauwer et al. (2000) define the concept of the **matrix unfolding** of a tensor, a generalisation of matrix decomposition as row (column) vectors. Let $A_{i_1 i_2 \ldots i_N} \in \mathbb{C}^{I_1 \times I_2 \ldots I_N}$ be a N-dimensional tensor, the matrix unfolding denoted as $\mathbf{A}_{(j)}$ of dimension $(I_j) \times (I_{j+1} I_{j+2} \ldots I_N I_1 I_2 \ldots I_{j-1})$ is a matrix with the element $a_{i_1 i_2 \ldots i_N}$ at the row position i_n and column position equal to $(i_{j+1}-1)I_{j+2}I_{j+3}\ldots I_N I_1 I_2 \ldots I_{j-1} + (i_{j+2}-1)I_{j+3}I_{j+4}\ldots I_N I_1 I_2 \ldots I_{j-1} + (i_N - 1)I_1 I_2 \ldots I_{j-1} + (i_1 - 1)I_2 I_3 \ldots I_{j-1} + (i_2 - 1)I_3 I_4 \ldots I_{j-1} + i_{n-1}$. The j-**mode** vector of A is obtained from the components $a_{i_1 i_2 \ldots i_j \ldots i_N}$, where j is the only changing index. The j-**rank** of a tensor A, defined as $R_j = \text{rank}_j(A)$, is the dimension of spanned vector space of the j-**mode** vectors. The **rank** of an N-th-order tensor A, $R = \text{rank}(A)$ is the minimal number of the rank-one tensors that yields A as a linear combination. The j-**mode** product of a N-th-order tensor A by a matrix $\mathbf{U}$ of dimension $J_n \times I_n$, denoted as $A \times_n \mathbf{U}$, is a Q-th-order tensor whose components are:

$$(A \times_n \mathbf{U})(Q;i_1,i_2,\ldots,i_{n-1},j_n,i_{n+1},\ldots,i_N) = \sum_{i_n} a(Q;i_1,i_2,\ldots,i_{n-1},i_n,i_{n+2},\ldots i_N,u_{j_n,i_n}).$$

This is a special case of inner product of two tensors, using the new notation of Lathauwer et al. (2000). A sub-tensor $S_{i_j=\alpha}$ from N-th-order tensor S, is the $N-1$-th-order tensor formed from S with the index i_j fixed to $\alpha \in (1,\ldots,N)$. With all these definitions Lathauwer et al. (2000) formulate the most important result, with the derivation of MSVD:

Definition 4.1: A N-th-order tensor A, can be decomposed as the product:

$$A = S \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_N \mathbf{U}^{(N)},$$

where $\mathbf{U}^{(j)}$ is a unitary matrix, and S is a N-th-order tensor with the property of all tensors being orthogonal. Two sub-tensors are orthogonal if the inner product $\langle S_{i_n=\alpha}, S_{i_m=\beta}$ between them is zero.

Nevertheless, the MSVD produces a decomposition with second-order tensors $\mathbf{U}^i, i \in \{1, \ldots, N\}$, such as they are orthogonal in pairs. To achieve the decomposition in (8) and (9), we will need the tensors $\mathbf{U}^i$ to be orthogonal in triplets, and orthogonal in N-tuplets in the case we would like to extend Proposition 3.2 for higher-order moments exact simulation. An extension to our work is suggested as a MSVD that could accomplish orthogonality in N-tuplets.

5. Numerical examples

We solve the system of equations (13), (14), and (15) using the MATLAB optimisation toolbox. The function *fsolve(·)* is used to find a solution to the systems of non-linear equations.

Example 5.1 A second-order objective moment $\widehat{M}(2)$, and third-order objective moment $\widehat{M}(3)$ are generated as the result of calculating the sample moments of a random vector of 2-dim of $N = 500$ samples. In the right-hand columns of Table A1, are the objective moments. In the left-hand columns are the sample moments of the vector X, of 2-dim with $N = 20$ samples generated with the method described in Section 4. We observe differences in sample moments, with the objective moments being relatively small. The first-order moment is exactly the same, as we translate the resulting sample by the vector of objective means. For this reason, we do not report first-order moment differences in the following examples.

Example 5.2 In this case we use the same objective moments $\widehat{M}(2)$, $\widehat{M}(3)$ as Example 5.1 with a small modification. The third-order central moment is increased to $\widehat{M}(3; 2, 2, 2) = 0.80$. Figure B1(a) shows the resulting samples. In blue are the samples generated applying the algorithm in Section 4, in red are the initial samples from a BVN distribution used to generate the blue samples. Increasing the third-order objective moment $\widehat{M}(3; 2, 2, 2)$ produces an increase in the second-order moments (correlation) and the bias we can see in the blue

samples. A second test is to change the third-order moment to a negative value: $\widehat{M}(3;2,2,2) = -0.80$. In Figure B1(b) we observe the resulting samples in blue. The second-order moments (correlation) is higher, but not as in Figure B1(a). A constraint, given that the second-order moments are positive, prevents the correlation of blue samples becoming negative.

Example 5.3 Now a more complex scenario is generated. We proceed to generate th second- and third-order objective moments $\widehat{M}(2)$, $\widehat{M}(3)$ as in Example 5.1, for a 5-dimensional (5-dim) vector. In Table A2 are the objective moments. The vector X of 5-dim is generated with $N = 20$ samples. Table A3 presents the samples generated. The sample moments of X are presented in Table A4. The differences between objective and sample second- and third-order moments are small. The increase in dimension will increase the number of variables of the systems of equations (13), (14), and (15), then the non-linear solver will have a better performance in finding a solution.

Example 5.4

A final set of examples is generated with different dimensions, from 2-dim to 5-dim, with a different number of samples required, from N=20 to N=100. For each combination we repeat the Monte Carlo algorithm 4 by 20 times, and we calculate the Euclidean norm error of the moments of the resulting samples against the objective moments, and we average the 20 repetitions. Tables A5 and A6 present the results for the second-order sample moment norm error, and the third-order sample moment norm error. The error reported is $\sim 0.2\% - 1\%$ in the case of the second-order moments; and approximately $1\% - 4\%$ in the case of the third-order moments; with an error of 9% only for the 4-dim, $N = 100$ case.

6. Application for portfolio risk assessment: Value-at-Risk (*VaR*)

Risk measurement is fundamental for a portfolio manager, and exploring the sensitivity of the portfolio's risk to changes in the weights of its components will be part of this assignment. The standard measure for measuring market risk adopted by the industry is the *VaR*. There basically three methods for calculating the *VaR*: historical methods, parametric methods like normal *VaR*, and Monte Carlo simulation. The first method does not allow us to measure extreme events, unless they are part of the history of the asset. Even in that case, we do not have control over the scale of the tail event. Parametric methods are perfect for sensitivity analysis, but we will need to fit a parametric distributions and some errors could be gained

in this process. Additionally, some parametric distributions do not acknowledge the presence of some higher-order moments in the data, like the normal distribution. In this case, the Monte Carlo simulations will suit the needs of the sensitivity analysis. The Monte Carlo approximate first-, second-, and third-order moment simulations will offer an advantage over classical exact first- and second-order moments simulation when measuring risk, like *VaR*. We explore an example of the *VaR* of a portfolio.

Let us define a portfolio with three assets, $\omega = (\omega(1), \omega(2), \omega(3))$. Assume that the density of the asset's returns follows a multivariate Student-t distribution, with $\nu = 3$ degrees of freedom, and parameter,

$$\Sigma = \begin{pmatrix} 1 & 0.8 & 0.8 \\ 0.8 & 1 & 0.8 \\ 0.8 & 0.8 & 1 \end{pmatrix}.$$

A simulated vector $\tilde{X}$ with 100 days is generated, this vector represents the data observed by the portfolio manager, and has a resulting mean $\mu = (0.025, -0.030, 0.084)$, and covariance,

$$V = \begin{pmatrix} 2.33 & 2.23 & 2.12 \\ 2.23 & 3.76 & 2.92 \\ 2.12 & 2.92 & 3.20 \end{pmatrix}.$$

We calculate the 1-day *VaR* at 99% confidence level of the sample for three different weights' combinations of the portfolio: $\omega_A = (0.9, 0.05, 0.05), \omega_B = (0.05, 0.9, 0.05), \omega_C = (0.05, 0.05, 0.9)$, with resulting $VaR_A = 3.96\%$, $VaR_B = 4.01\%$, $VaR_C = 5.35\%$. The third portfolio has a higher *VaR*, although the variance of the third asset is lower than the variance of the second asset, as a result of large third-order moments present in the sample. Figure B2 shows three scatter plots of the market returns (blue cross) from $\tilde{X}$, where we can see the bias of the returns, from the first and second assets towards the third asset. As the market distribution is unknown for portfolio manager, we generate a Monte Carlo exact first- and second-order moment simulation of 100-days $\tilde{Z}$, with the parameters μ, V, extracted from the market sample $\tilde{X}$, following the *QR decomposition* algorithm described in Meucci (2009). The *VaR* of the resulting sample $\tilde{Z}$ for the three scenarios is: $VaR_A = 2.93\%$, $VaR_B = 3.53\%$, $VaR_C = 3.71\%$. The three scenarios are underestimated by the exact first- and second-order moments methodology. Now we generate a 100-days sample $\tilde{Y}$, applying the method-

ology described in Section 4. The resulting VaR for the three scenarios are: $VaR_A = 3.59\%$, $VaR_B = 3.71\%$, $VaR_C = 4.53\%$. Although our methodology is still underestimating the real values, it over-performs the methodology of Monte Carlo exact first- and second-order moments simulation, improving from a 77% of accuracy of the exact first- and second-order moments method to a 89% of accuracy of the real VaR value on average.

7. Conclusions

A methodology to generate samples with a Monte Carlo approximate first-, second-, and third-order moment has been presented. The methodology is based on the theory of tensors. The first step of the algorithm is to generate a multivariate standard normal (MVSN) sample X. Then the algorithm determines the first-, second-, and third-order objective moments; and by an extension of the Cholesky decomposition to tensors of arbitrary dimension, a set of non-linear equations are established. The system of non-linear equations are solved, and the equivalent Cholesky tensor decomposition is multiplied by the sample X. The algorithm was tested in a MATLAB environment, and the functions for solving non-linear equations provided by MATLAB were used. The results demonstrate that the methodology can transform the moments of a generated sample X close to a desired level. Extensions to our work include providing the structure of the non-linear problem to the optimisation software, an extension to a Monte Carlo approximate first-, second-, third-, and fourth-order moments method and testing another optimisation software for the solution of the system of non-linear equations.

References

Avellaneda, M. (1998), 'Minimum-relative-entropy calibration of asset pricing models', *International Journal of Theoretical & Applied Finance* **1**(4), 447.

Cheng, R. C. H. (1985), 'Generation of multivariate normal samples with given sample mean and covariance matrix', *Journal of Statistical Computation and Simulation* **21**, 39–49.

D'Amico, M., G. Fusai and A. Tagliani (2003), 'Valuation of exotic options using moments', *Operational Research* **2**, 157–186.

Glasserman, P. and B. Yu (2005), 'Large sample properties of weighted monte carlo estimators', *Operations Research* **53**, 298–312.

Jackel, P. (2002), *Monte Carlo Methods in Finance*, Wiley.

Kendall, M. G. (1947), *The Advanced Theory of Statistics*, Charles Griffin & Co.

Lathauwer, L. D., B. D. Moor and J. Vandewalle (1994), 'Blind source separation by higher-order singular value decomposition', *Proceedings of EUSIPCO-94 VIIth European Signal Processing Conference* **1**, 175–178.

Lathauwer, L.D., B. D. Moor and J. Vandewalle (2000), 'A multilinear singular value decomposition', *SIAM J. Matrix Anal. Appl.* **21**(4), 1253–1278.

Ledermann, W., C. Alexander and D. Ledermann (2011), 'Random orthogonal matrix simulation', *Linear Algebra and its Applications* **434**(6), 1444–1467.

Li, K. H. (1992), 'Generation of random matrices with orthonormal columns and multivariate normal variates with given sample mean and covariance', *Journal of Statistical Computation and Simulation* **43**, 11–18.

Mardia, K. V. (1970), 'Measures of multivariate skewness and kurtosis with applications', *Biometrika* **57**(3), 519–530.

McCullagh, P. (1987), *Tensor Methods in Statistics*, Chapman & Hall.

Meucci, A. (2009), 'Simulations with exact means and covariances', *Risk* **22**(7), 89–91.

Wedderburn, R.W.M. (1975), 'Random rotations and multivariate normal simulation'. Research Report, Rothamsted Experimental Station.

Appendix A. Tables

Table A1.: First-, second-, and third-order objective moments and first-, second-, and third-order sample moments obtained from the Monte Carlo approximate simulation method (2-dim case).

First-order objective moment ($\widehat{M}_r$)	
s_1	s_2
0.1217	0.1112

Sample first-order moment (M_r)	
s_1	s_2
0.1217	0.1112

Second-order objective moment ($\widehat{M}_{r,2}$)		
	s_1	s_2
s_1	0.3432	0.2289
s_2	0.2289	0.2820

Second-order sample moment ($M_{r,s}$)		
	s_1	s_2
s_1	0.3581	0.2581
s_2	0.2581	0.3236

Third-order objective moment ($\widehat{M}_{r,s,t}$)				
	s_1^2	$s_2 s_1$	$s_2 s_1$	s_2^2
s_1	0.2386	0.1626	0.1626	0.1652
s_2	0.1626	0.1652	0.1652	0.2442

Sample third-order moment ($M_{r,s,t}$)				
	s_1^2	$s_2 s_1$	$s_2 s_1$	s_2^2
s_1	0.1967	0.1817	0.1817	0.1621
s_2	0.1817	0.1621	0.1621	0.1732

Table A2.: First-, second-, and third-order objective moments (5-dim case).

First-order objective moment				
s_1	s_2	s_3	s_4	s_5
0.1197	0.1901	0.1759	0.1844	0.1974

Second-order objective moment					
	s_1	s_2	s_3	s_4	s_5
s_1	0.2440	0.2540	0.1987	0.2574	0.2291
s_2	0.2540	0.4145	0.2699	0.3045	0.3168
s_3	0.1987	0.2699	0.2626	0.2522	0.2338
s_4	0.2574	0.3045	0.2522	0.3758	0.2833
s_5	0.2291	0.3168	0.2338	0.2833	0.3375

Third-order objective moment

	s_1^2	$s_2 s_1$	$s_3 s_1$	$s_4 s_1$	$s_5 s_1$	$s_2 s_1$	s_2^2	$s_2 s_3$	$s_2 s_4$	$s_2 s_5$
s_1	0.3084	0.1532	0.2423	0.1741	0.2111	0.1532	0.1269	0.1275	0.0868	0.1322
s_2	0.1532	0.1269	0.1275	0.0868	0.1322	0.1269	0.1724	0.1302	0.0853	0.1463
s_3	0.2423	0.1275	0.2318	0.1586	0.1787	0.1275	0.1302	0.1476	0.1029	0.1376
s_4	0.1741	0.0868	0.1586	0.1641	0.1227	0.0868	0.0853	0.1029	0.0994	0.0990
s_5	0.2111	0.1322	0.1787	0.1227	0.1974	0.1322	0.1463	0.1376	0.0990	0.1700

	$s_3 s_1$	$s_3 s_2$	s_3^2	$s_3 s_4$	$s_3 s_5$	$s_4 s_1$	$s_4 s_2$	$s_4 s_3$	s_4^2	$s_4 s_5$
s_1	0.2423	0.1275	0.2318	0.1586	0.1787	0.1741	0.0868	0.1586	0.1641	0.1227
s_2	0.1275	0.1302	0.1476	0.1029	0.1376	0.0868	0.0853	0.1029	0.0994	0.0990
s_3	0.2318	0.1476	0.2690	0.1934	0.2078	0.1586	0.1029	0.1934	0.1944	0.1441
s_4	0.1586	0.1029	0.1934	0.1944	0.1441	0.1641	0.0994	0.1944	0.2268	0.1474
s_5	0.1787	0.1376	0.2078	0.1441	0.2067	0.1227	0.0990	0.1441	0.1474	0.1542

	$s_5 s_1$	$s_5 s_2$	$s_5 s_3$	$s_5 s_4$	s_5^2
s_1	0.2111	0.1322	0.1787	0.1227	0.1974
s_2	0.1322	0.1463	0.1376	0.0990	0.1700
s_3	0.1787	0.1376	0.2078	0.1441	0.2067
s_4	0.1227	0.0990	0.1441	0.1474	0.1542
s_5	0.1974	0.1700	0.2067	0.1542	0.2660

Table A3.: Values of a sample generated with a Monte Carlo approximate moments simulation method (5-dim case).

Sample generated (5-dim, N=20)				
X_1	X_2	X_3	X_4	X_5
0.2791	0.5235	0.2834	1.0298	0.6866
0.4316	1.0944	0.1355	0.5152	0.6132
-0.6010	-0.8691	-0.6412	-0.7440	-0.2394
0.6724	0.8090	0.9592	0.6549	1.1254
0.5077	0.4888	1.0831	1.4710	0.4992
-0.2419	-0.8959	-0.3187	-0.4106	-0.3037
-0.2415	-0.8342	-0.2978	-0.8691	-0.5319
0.4499	-0.2079	-0.2486	0.4633	-0.3993
0.0197	0.3657	0.4868	0.2051	1.0965
0.0499	1.1647	0.4716	0.3478	0.2056
0.2582	0.7430	-0.0030	0.1326	0.6598
-0.1747	-0.1212	0.0219	-0.3373	-0.4584
-0.2855	-0.2423	0.2057	-0.3922	-0.3945
-0.5299	-0.3700	-0.3843	-0.4616	-0.5498
-0.3707	0.1682	0.0146	0.1336	0.2216
-0.0315	-0.0037	0.3269	-0.0768	-0.1086
0.1746	0.1989	-0.1472	0.5354	0.3068
0.3424	0.7746	0.1880	0.1552	0.4486
-0.0808	0.0760	-0.0357	0.3721	-0.1134
1.7655	0.9390	1.4185	0.9629	1.1847

Table A4.: First-, second-, and third-order sample moments obtained from the Monte Carlo approximate moments simulation method (5-dim case).

First-order sample moment				
s_1	s_2	s_3	s_4	s_5
0.1197	0.1901	0.1759	0.1844	0.1974

Second-order sample moment					
	s_1	s_2	s_3	s_4	s_5
s_1	0.2763	0.2345	0.2229	0.2451	0.2274
s_2	0.2345	0.4235	0.2503	0.3031	0.3114
s_3	0.2229	0.2503	0.2841	0.2484	0.2423
s_4	0.2451	0.3031	0.2484	0.3836	0.2696
s_5	0.2274	0.3114	0.2423	0.2696	0.3538

Third-order sample moment										
	s_1^2	s_2s_1	s_3s_1	s_4s_1	s_5s_1	s_2s_1	s_2^2	s_2s_3	s_2s_4	s_2s_5
s_1	0.2860	0.1626	0.2397	0.1748	0.2109	0.1626	0.1119	0.1410	0.0993	0.1474
s_2	0.1626	0.1119	0.1410	0.0993	0.1474	0.1119	0.1707	0.1095	0.0827	0.1491
s_3	0.2397	0.1410	0.2235	0.1630	0.1948	0.1410	0.1095	0.1523	0.1132	0.1421
s_4	0.1748	0.0993	0.1630	0.1425	0.1377	0.0993	0.0827	0.1132	0.0971	0.1149
s_5	0.2109	0.1474	0.1948	0.1377	0.1830	0.1474	0.1491	0.1421	0.1149	0.1771

	s_3s_1	s_3s_2	s_3^2	s_3s_4	s_3s_5	s_4s_1	s_4s_2	s_4s_3	s_4^2	s_4s_5
s_1	0.2397	0.1410	0.2235	0.1630	0.1948	0.1748	0.0993	0.1630	0.1425	0.1377
s_2	0.1410	0.1095	0.1523	0.1132	0.1421	0.0993	0.0827	0.1132	0.0971	0.1149
s_3	0.2235	0.1523	0.2453	0.2006	0.2044	0.1630	0.1132	0.2006	0.1855	0.1596
s_4	0.1630	0.1132	0.2006	0.1855	0.1596	0.1425	0.0971	0.1855	0.2307	0.1410
s_5	0.1948	0.1421	0.2044	0.1596	0.2011	0.1377	0.1149	0.1596	0.1410	0.1557

	s_5s_1	s_5s_2	s_5s_3	s_5s_4	s_5^2
s_1	0.2109	0.1474	0.1948	0.1377	0.1830
s_2	0.1474	0.1491	0.1421	0.1149	0.1771
s_3	0.1948	0.1421	0.2044	0.1596	0.2011
s_4	0.1377	0.1149	0.1596	0.1410	0.1557
s_5	0.1830	0.1771	0.2011	0.1557	0.2464

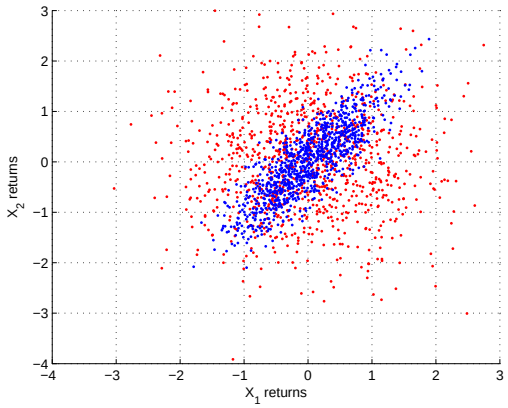
Table A5.: Euclidean norm error of the difference between a second-order objective moment and a second-order sample moment obtained from the Monte Carlo approximate moments simulation method for different dimensions and number of samples required.

Norm error of second-order sample moments			
	20	40	100
2-dim	0.001194	0.000430	0.008636
3-dim	0.002032	0.001939	0.006134
4-dim	0.008078	0.003842	0.044682
5-dim	0.004368	0.010661	0.008345

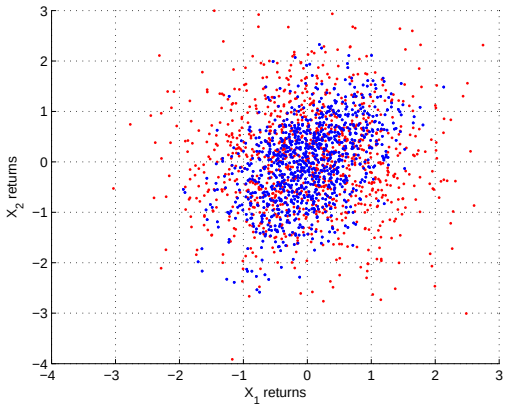
Table A6.: Euclidean norm error of the difference between a third-order objective moment and a third-order sample moment of a Monte Carlo approximate moments simulation method for different dimensions and number of samples required.

Norm error of third-order sample moments			
	20	40	100
2-dim	0.005828	0.003919	0.024863
3-dim	0.012324	0.017701	0.028279
4-dim	0.013384	0.011493	0.095806
5-dim	0.019222	0.041512	0.040928

Appendix B. Figures

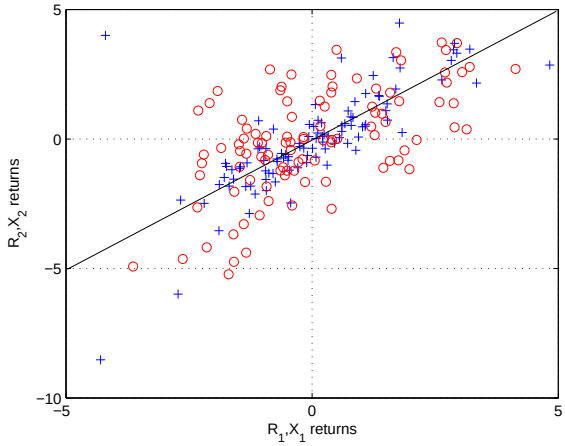


(a) Bivariate samples generated with a high value on the third-order objective moment m_{03}.

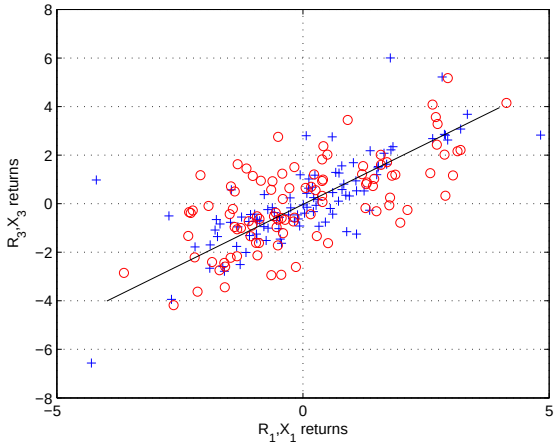


(b) Bivariate samples generated with a negative value on the third-order objective moment m_{03}.

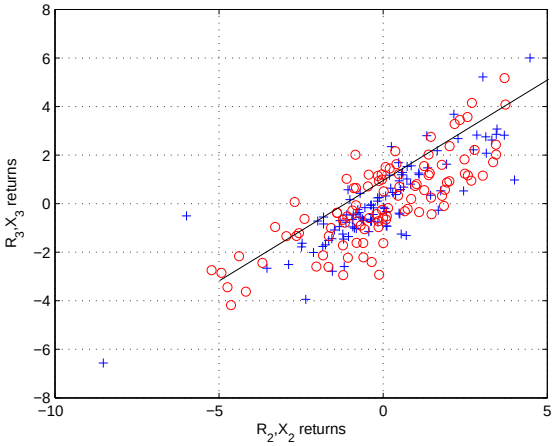
Figure B1.: Scatter plots of the Monte Carlo approximate moments simulation method generated samples (blue) and of BVSN generated samples (red).



(a) Scatter plot of returns of Student-t generated samples (blue cross) and of Monte Carlo approximate moments simulation method generated samples (red circle) between the first and the second component.



(b) Scatter plot of returns of Student-t generated samples (blue cross) and of Monte Carlo approximate moments simulation method generated samples (red circle) between the first and the third component.



(c) Scatter plot of returns of Student-t generated samples (blue cross) and of Monte Carlo approximate moments simulation method generated samples (red circle) between the second and the third component.

Figure B2.: Scatter plot of sample returns of the market (blue), and of Monte Carlo approximate moments simulation method generated samples.