Linear Algebra and its Applications 434 (2011) 1444–1467

Contents lists available at ScienceDirect

Linear Algebra and its Applications

journal homepage: www.elsevier.com/locate/laa

Random orthogonal matrix simulation ☆

Walter Ledermann [a], Carol Alexander [b,*], Daniel Ledermann [b]

[a] University of Sussex, Brighton BN1 9RH, UK
[b] ICMA Centre, University of Reading, Reading RG6 6BA, UK

ARTICLE INFO

Article history:
Received 21 August 2009
Accepted 28 October 2010
Available online 25 November 2010

Submitted by T. Laffey

AMS classification:
C14
C15
C63

Keywords:
Simulation
Orthogonal matrix
Random matrix
Ledermann matrix
L matrices
Multivariate moments
Volatility clustering
Value-at-risk

ABSTRACT

This paper introduces a method for simulating multivariate samples that have exact means, covariances, skewness and kurtosis. We introduce a new class of rectangular orthogonal matrix which is fundamental to the methodology and we call these matrices *L* matrices. They may be deterministic, parametric or data specific in nature. The target moments determine the *L* matrix then infinitely many random samples with the exact same moments may be generated by multiplying the *L* matrix by arbitrary random orthogonal matrices. This methodology is thus termed "ROM simulation". Considering certain elementary types of random orthogonal matrices we demonstrate that they generate samples with different characteristics. ROM simulation has applications to many problems that are resolved using standard Monte Carlo methods. But no parametric assumptions are required (unless parametric *L* matrices are used) so there is no sampling error caused by the discrete approximation of a continuous distribution, which is a major source of error in standard Monte Carlo simulations. For illustration, we apply ROM simulation to determine the value-at-risk of a stock portfolio.

© 2010 Elsevier Inc. All rights reserved.

Introduction

Parametric Monte Carlo simulation is the ubiquitous tool for generating multivariate random samples. However, unless the sample size is very large there can be substantial differences between the target distribution and the empirical distributions of the simulations. But generating very large samples

☆ The publication of this paper is timed to coincide with the centennial of Walter Ledermann's birth on 18 March 1911 in Berlin. As a researcher, teacher and writer he was one of the most influential mathematicians of the twentieth century. This is Walter's last research paper. He died soon after publishing his 'Encounters of a Mathematician', aged 98.

* Corresponding author.
E-mail addresses: c.alexander@icmacentre.rdg.ac.uk (C. Alexander), d.ledermann@icmacentre.rdg.ac.uk (D. Ledermann).

0024-3795/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.laa.2010.10.023

can be extremely time-consuming. For instance, when Monte Carlo simulation is applied to financial risk assessment it is standard to use at least 100,000 simulations, and for each simulation a valuation model must be applied to each instrument in the portfolio. A typical portfolio held by a large bank contains many thousands of financial instruments, and when the portfolio contains exotic or path-dependent contingent claims the valuation models themselves require very large numbers of simulations. Thus, assessing portfolio risk via simulation usually takes many hours of computational time, even though most banks have highly sophisticated computers.[1]

This paper introduces a new method for multivariate simulation based on random orthogonal matrices, which we call "ROM simulation". The focus of ROM simulation is to eliminate sampling error in the sample mean vector, covariance matrix and the Mardia [23] multivariate skewness and kurtosis measures, so that in each simulation they are exactly equal to their target values. Methods for matching the moments of a multivariate simulation have undergone many recent developments. In particular, Lyhagen [19] constructs multivariate samples using linear combinations of random variables with known multivariate distributions. Although the distribution of this sum may be unknown, the coefficients of the combination may be optimised in order to control the moments of the multivariate simulation. This approach is applied to the analysis of non-normal residuals, which are observed frequently in empirical finance. Date et al. [7] consider symmetric multivariate distributions and focus on matching marginal kurtosis. They achieve this via an optimisation algorithm which constrains the probabilities of simulated scenarios and applications to portfolio risk management in finance and sigma point filtering in engineering are discussed.

The methods of Lyhagen [19] and Date et al. [7] are statistical in nature. We attempt to solve the same type of problem using linear algebra. By introducing L matrices, a new class of rectangular orthogonal matrices that are fundamental to ROM simulation, we provide constructive algorithms for generating exact moment simulations. We believe that ROM simulation, like Monte Carlo simulation, will have numerous applications across many different disciplines. In quantitative finance alone there are immediate applications to three very broad areas: exotic option pricing and hedging, portfolio risk assessment and asset allocation techniques. Since this paper focuses on introducing the class of L matrices and describing the basic theoretical properties of ROM simulation, there is only space to include a brief example of one such application.

To motivate our research, consider the multivariate normal (MVN) case where we have a target mean vector, $\boldsymbol{\mu}_n$ and a target covariance matrix, $\mathbf{S}_n$ for n random variables. We want to generate a random sample $\mathbf{X}_{mn}$, which contains m observations on the n random variables and where the sample mean vector and sample covariance matrix of $\mathbf{X}_{mn}$ match the target mean vector $\boldsymbol{\mu}_n$ and covariance matrix $\mathbf{S}_n$.[2] That is,[3]

$$m^{-1}(\mathbf{X}_{mn} - \mathbf{1}_m\boldsymbol{\mu}_n')'(\mathbf{X}_{mn} - \mathbf{1}_m\boldsymbol{\mu}_n') = \mathbf{S}_n. \tag{1}$$

Clearly, setting

$$\mathbf{X}_{mn} = \mathbf{Z}_{mn} + \mathbf{1}_m(\boldsymbol{\mu}_n' - \bar{\mathbf{z}}_n'),$$

where $\mathbf{Z}_{mn}$ is a MVN simulation with sample mean $\bar{\mathbf{z}}_n$, yields a simulation with mean $\boldsymbol{\mu}_n$. However, at first, it is not immediately obvious that a matrix $\mathbf{X}_{mn}$ satisfying (1) will exist, for any covariance matrix $\mathbf{S}_n$. But since $\mathbf{S}_n$ is positive semi-definite we can find a decomposition $\mathbf{S}_n = \mathbf{A}_n'\mathbf{A}_n$, where $\mathbf{A}_n$ is a Cholesky matrix, for example.[4] Then by considering the transformation

[1] This sampling error problem has spawned a voluminous literature on methods for increasing the speed and reducing the sampling error when Monte Carlo methods are applied to financial problems. See [10] for a review.

[2] Throughout this paper we will use the following notation, unless stated otherwise: Vectors, will be columns vectors, written in lowercase bold with a single index, e.g. $\mathbf{a}_n$. Rectangular matrices will be uppercase bold with a row and column index, e.g. $\mathbf{A}_{mn}$. Square matrices will have a single index, e.g. $\mathbf{A}_n$.

[3] For ease of exposition we ignore any bias adjustments in sample moments, so that m rather than $m-1$ appears in the denominator of the sample variances and covariances of our random variables. When samples moments are bias-adjusted the equivalent formulae in our theoretical results are available from the authors on request.

[4] Another alternative would be to set $\mathbf{S}_n = \mathbf{Q}_n\boldsymbol{\Lambda}_n\mathbf{Q}_n'$ where $\boldsymbol{\Lambda}_n$ is the diagonal matrix of eigenvalues, and $\mathbf{Q}_n$ is the orthogonal matrix of eigenvectors of $\mathbf{S}_n$, so that $\mathbf{A}_n = \boldsymbol{\Lambda}_n^{-\frac{1}{2}}\mathbf{Q}_n'$.

$$\mathbf{L}_{mn} = m^{-1/2}(\mathbf{X}_{mn} - \mathbf{1}_m \boldsymbol{\mu}_n')\mathbf{A}_n^{-1}, \tag{2}$$

it is clear that solving (1) is equivalent to finding a matrix $\mathbf{L}_{mn}$ which satisfies

$$\mathbf{L}_{nm}'\mathbf{L}_{mn} = \mathbf{I}_n \quad \text{with} \quad \mathbf{1}_m'\mathbf{L}_{mn} = \mathbf{0}_n'. \tag{3}$$

By solving (3) and inverting transformation (2) we will be able to produce exact MVN samples $\mathbf{X}_{mn}$, given any desired sample mean $\boldsymbol{\mu}_n$ and covariance matrix $\mathbf{S}_n$.

Any matrix satisfying (3) will be referred to as an *L* matrix in memory of Walter Ledermann, who motivated this work by finding an explicit, deterministic solution to (3) at the age of 97. The condition (3) actually turns out to be fundamental to our simulation methodology, in which many different types of *L* matrices have since been applied. Thus, we shall begin by categorising the *L* matrices that are considered in this paper.

Characterising a sample via it's mean and covariance matrix is sufficient for MVN random variables, but it is not necessarily sufficient for more complex multivariate distributions. Moreover, in some applications it may not be advantageous to assume any particular parametric form for the distribution of $\mathbf{X}_{mn}$. For example, when assessing portfolio risk, most financial institutions prefer to perform simulations that are based on historical data; this is the so-called "historical simulation" approach. Here the portfolio return distribution is characterised by fitting a kernel to an historical distribution, or by the moments of this distribution.[5] Therefore, the main focus of our research is to introduce methods for generating exact covariance samples which also match the Mardia multivariate skewness and kurtosis of the random variables in question. In this framework there is no need to specify a parametric form for the distribution function.

The paper is outlined as follows. Section 1 provides a classification of the different types of *L* matrices that we consider in this paper. Deterministic, parametric and data-specific techniques are described for constructing *L* matrices and within each construction many different types of *L* matrices exist. For the sake of brevity, here we specify only three distinct types, being of increasing complexity. Section 2 develops exact moment simulation algorithms which combine *L* matrices with other, random, orthogonal matrices. We have given the name "random orthogonal matrix (ROM) simulation" to this approach. We characterise the skewness and kurtosis of ROM simulations and, by focusing on a particular type of deterministic *L* matrix, we investigate how different orthogonal matrices can affect the dynamic and marginal characteristics of ROM simulated samples. In Section 3 we apply ROM simulation to estimate the value-at-risk for a portfolio invested in Morgan Stanley Country Indices (MSCI). Section 4 concludes. Some proofs are in the Appendix.

1. Classification of *L* matrices

A square matrix $\mathbf{Q}_n$ is *orthogonal* if and only if its inverse is equal to its transpose. That is

$$\mathbf{Q}_n'\mathbf{Q}_n = \mathbf{Q}_n\mathbf{Q}_n' = \mathbf{I}_n. \tag{4}$$

This is equivalent to saying that the rows of $\mathbf{Q}_n$ form an orthonormal set of vectors, as do the columns of $\mathbf{Q}_n$. However for our work we need to extend definition (4) to rectangular matrices. We do this in the obvious way, stating that $\mathbf{Q}_{mn}$ is a (rectangular) orthogonal matrix if and only if

$$\mathbf{Q}_{nm}'\mathbf{Q}_{mn} = \mathbf{I}_n. \tag{5}$$

From this definition, and the rank-nullity Theorem, it follows that we must have $m \geq n$. Then (5) implies that the column vectors of the matrix $\mathbf{Q}_{mn}$ are orthonormal.[6]

[5] A recent survey found that, of the 65% of firms that disclose their value-at-risk methodology, 73% used historical simulation, see [25].

[6] However, (5) does not necessarily imply that the row vectors of $\mathbf{Q}_n$ are orthonormal.

We can now begin to think geometrically about the condition (3) which defines an L matrix. Essentially, in order to satisfy (3) we must find n orthonormal vectors in $\mathbb{R}^m$, which all lie on the hyper-plane

$$\mathcal{H} = \{(h_1, \ldots, h_m)' \in \mathbb{R}^m \mid h_1 + \cdots + h_m = 0\}.$$

Certainly $\mathcal{H}$ contains $m-1$ linearly independent vectors and since $\mathcal{H} \subset \mathbb{R}^m$, it follows that $\dim(\mathcal{H}) = m-1$. So, provided that $m > n$, we will be able to select n vectors from an orthonormal basis for $\mathcal{H}$. By allocating these n vectors to the columns of a matrix $\mathbf{L}_{mn}$, we will have at least one solution fulfilling (3). Therefore the key to defining an L matrix lies in finding an orthonormal set within the hyper-plane $\mathcal{H}$.

Fortunately, we have the Gram-Schmidt procedure at our disposal, which provides a sequential method for finding an orthonormal set from a set of linearly independent vectors.[7] Formally, let $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ be a set of linearly independent vectors, equipped with an inner product. A set of orthogonal vectors $\{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$ can be found via the Gram-Schmidt procedure where

$$\begin{aligned}
\mathbf{u}_1 &= \mathbf{v}_1 \\
\mathbf{u}_2 &= \mathbf{v}_2 - \operatorname{proj}_{\mathbf{u}_1}(\mathbf{v}_2) \\
&\vdots \\
\mathbf{u}_n &= \mathbf{v}_n - \sum_{i=1}^{n-1} \operatorname{proj}_{\mathbf{u}_i}(\mathbf{v}_n),
\end{aligned} \tag{6}$$

and where, at each step, the projection operator $\operatorname{proj}_{\mathbf{u}} : \mathbb{R}^m \to \mathbb{R}^m$, is defined as

$$\operatorname{proj}_{\mathbf{u}}(\mathbf{v}) := \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\langle \mathbf{u}, \mathbf{u} \rangle} \mathbf{u}.$$

Our final step is to map this orthogonal set to an orthonormal set $\{\mathbf{w}_1, \ldots, \mathbf{w}_n\}$, using the standard normalisation

$$\mathbf{w}_i = \langle \mathbf{u}_i, \mathbf{u}_i \rangle^{-1/2} \mathbf{u}_i \quad \text{for} \quad 1 \le i \le n. \tag{7}$$

For simplicity we write this transformation of n linearly independent vectors $\mathbf{V}_{mn} = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ to n orthonormal vectors $\mathbf{W}_{mn} = \{\mathbf{w}_1, \ldots, \mathbf{w}_n\}$ as

$$\mathbf{W}_{mn} = GS(\mathbf{V}_{mn}).$$

With this notation we will often refer to $\mathbf{W}_{mn}$ as the *GS image* of $\mathbf{V}_{mn}$ and $\mathbf{V}_{mn}$ as the *GS pre-image* of $\mathbf{W}_{mn}$.

It is well known that the Gram-Schmidt procedure is subspace preserving. That is, if $\mathbf{W}_{mn} = GS(\mathbf{V}_{mn})$ then $\operatorname{span}(\mathbf{V}_{mn}) = \operatorname{span}(\mathbf{W}_{mn})$, where $\operatorname{span}(\mathbf{V}_{mn})$ denotes the subspace of $\mathbb{R}^m$ spanned by the n columns of $\mathbf{V}_{mn}$. So, in particular, if the linearly independent columns of $\mathbf{V}_{mn}$ belong to $\mathcal{H}$, then we may write $\mathbf{W}_{mn} = \mathbf{L}_{mn}$ since it will satisfy (3).

The rest of this section explores a range of possible solutions to (3) that can be found by specifying different linearly independent sets within the hyper-plane $\mathcal{H} \subseteq \mathbb{R}^m$. Typically, solutions are constructed as follows:

- Take a pair $(m, n) \in \mathbb{Z}_+^2$ with $m > n$ and pick $N(m)$ linearly independent vectors in $\mathcal{H}$ to form a matrix $\mathbf{V}_{m,N(m)}$, where $m > N(m) \ge n$.
- Apply the Gram-Schmidt procedure to $\mathbf{V}_{m,N(m)}$ and hence find an orthonormal basis whose elements form the columns of $\mathbf{W}_{m,N(m)}$.
- Select n columns from $\mathbf{W}_{m,N(m)}$ to form a matrix $\mathbf{L}_{mn}$.

[7] Orthogonalisation methods employing [14] reflections or [9] rotations are also available.

In general, the properties of an L matrix are inherited from the linearly independent vectors which are used in it's construction. In Section 1.1 we specify these vectors non-randomly, using zeros, ones and integer parameters. These vectors produce deterministic L matrices, whose properties are determined by the parameters specifying the vectors. This contrasts with parametric L matrices, which are defined in Section 1.2. Here linearly independent vectors are generated randomly, using parametric multivariate distributions. Properties of these L matrices are dependent upon the type of distribution used. Section 1.3 relies on existing sample data to generate linearly independent vectors. These produce "data-specific" L matrices, whose properties, as the name suggests, are determined by the data in question.

1.1. Deterministic L matrices

To construct a deterministic L matrix we take a pair $(m, n) \in \mathbb{Z}_+^2$ with $m > n$ and set $N(m) = m-1$. Then define a matrix $\mathbf{V}_{m,N(m)} = (\mathbf{v}_1, \ldots, \mathbf{v}_{N(m)})$ whose columns are specified as

$$\mathbf{v}_j = [\underbrace{0, \ldots, 0}_{j-1}, 1, -1, \underbrace{0, \ldots, 0}_{m-1-j}]' \quad \text{for} \quad 1 \le j \le N(m).$$

It is clear that $\{\mathbf{v}_1, \ldots, \mathbf{v}_{N(m)}\}$ is a linearly independent set and that $\mathbf{v}_j \in \mathcal{H}$ for $1 \le j \le N(m)$. We then construct $\mathbf{L}_{mn}$ using the last n columns from $\mathbf{W}_{m,N(m)}$, the GS image of $\mathbf{V}_{m,N(m)}$. After some algebraic manipulation we find that $\mathbf{W}_{m,N(m)}$ takes the following form:

$$\begin{pmatrix}
\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & \cdots & \frac{1}{\sqrt{j(j+1)}} & \cdots & \cdots & \frac{1}{\sqrt{(m-1)m}} \\
\frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & & \vdots & & & \vdots \\
0 & \frac{-2}{\sqrt{6}} & & \vdots & & & \vdots \\
\vdots & 0 & \ddots & \vdots & & & \vdots \\
\vdots & \vdots & \ddots & \frac{-j}{\sqrt{j(j+1)}} & & & \vdots \\
\vdots & \vdots & & 0 & \ddots & & \vdots \\
\vdots & \vdots & & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & \cdots & 0 & \frac{-(m-1)}{\sqrt{(m-1)m}}
\end{pmatrix}. \tag{8}$$

We remark that the matrix $\mathbf{W}_{m,N(m)}$ above is the last $m-1$ columns of a type of $m \times m$ generalised Helmert matrix.[8] Interestingly, it has been applied to change coordinate systems for geometric morphometric analysis, by Mardia and Walder [22], Mardia and Dryden [21] and others.

The last n columns of the matrix (8) form the L matrix $\mathbf{L}_{mn}$ which Walter Ledermann first defined as an explicit solution to (3). If we write $\mathbf{L}_{mn} = (\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_n)$, then for $1 \le j \le n$,

$$\boldsymbol{\ell}_j = [(m-n+j-1)(m-n+j)]^{-1/2}(\underbrace{1, \ldots, 1}_{m-n+j-1}, -(m-n+j-1), \underbrace{0, \ldots, 0}_{n-j})'. \tag{9}$$

Few other L matrices have a simple closed form such as (9), so it will be convenient to use this L matrix for illustrating the main concepts of ROM simulation. For this reason in the following we shall refer to this particular L matrix as the "Ledermann matrix".

[8] After the German geodecist Friedrich Helmert, 1843–1917. A standard Helmert matrix is an $n \times n$ matrix in which all elements occupying a triangle below the first row and above the principal diagonal are zero, i.e. $\mathbf{H} = (h_{ij})$ where $h_{ij} = 0$ for $j > i > 1$. If further $h_{1j} > 0$ for all j, $\sum_1^n h_{ij}^2 = 1$, and $h_{ij} > 0$ for $j < i$ then $\mathbf{H}$ is a Helmert matrix in the strict sense. Generalised Helmert matrices are square orthogonal matrices that can be transformed by permutations of their rows and columns and by transposition and by change of sign of rows, to a form of a standard Helmert matrix. See [17] for further details.

The remainder of this section defines other L matrices, written $\mathbf{L}_{mn}^{k}$ where k is an additional integer parameter, which generalise the Ledermann matrix in the sense that $\mathbf{L}_{mn} = \mathbf{L}_{mn}^{1}$ or $\mathbf{L}_{mn}^{0}$. As is clear from (9), the entries of $\mathbf{L}_{mn}$ only depend on the matrix dimension m and n. Now to define L^k-matrices we consider triples of the form $(m, n, k) \in \mathbb{Z}_+^3$. Each triple must satisfy $m > n$. Restrictions on k may also be imposed. With the introduction of a "free" parameter k these L^k-matrices may be quite complex to express explicitly. In the following we suggest three types of L^k-matrices, although there are endless possibilities.

Type I

For Type I matrices we allow only those triples with $2k \leq m + 1 - n$ and set $N(m) = m + 1 - 2k$, so $n \leq N(m)$. We define the Type I matrix $\mathbf{V}_{m,N(m)} = (\mathbf{v}_1, \ldots, \mathbf{v}_{N(m)})$ by setting

$$\mathbf{v}_j = [\underbrace{0, \ldots, 0}_{j-1}, \underbrace{1, -1, \ldots, 1, -1}_{2k}, \underbrace{0, \ldots, 0}_{m+1-2k-j}]' \quad \text{for} \quad 1 \leq j \leq N(m). \tag{10}$$

Again it is straightforward to check that the columns of $\mathbf{V}_{mn}$ form a linearly independent set in $\mathcal{H}$. Now the $\mathbf{L}_{mn}^{k}$ matrix of Type I is the last n columns of the GS image of $\mathbf{V}_{m,m+1-2k}$, defined above. By considering GS pre-images it is clear that $\mathbf{L}_{mn}^{1}$ is indeed the Ledermann matrix. For $k > 2$ one can easily check that $\mathbf{L}_{mn}^{k}$ will solve Eq. (3), but it is less easy to write down the form of $\mathbf{L}_{mn}^{k}$ explicitly. A direct specification of Type I, L^k-matrices is available from the authors on request.

Type II

Again we start with a triple $(m, n, k) \in \mathbb{Z}_+^3$, now satisfying the condition $N(m) = m - k \geq n$. In this case the GS pre-image matrix $\mathbf{V}_{m,N(m)} = (\mathbf{v}_1, \ldots, \mathbf{v}_{N(m)})$ is defined by

$$\mathbf{v}_j = [\underbrace{0, \ldots, 0}_{j-1}, \underbrace{1, \ldots, 1}_{k}, -k, \underbrace{0, \ldots, 0}_{m-k-j}]' \quad \text{for} \quad 1 \leq j \leq N(m). \tag{11}$$

We now define an L matrix of Type II as the last n columns of the GS image of $\mathbf{V}_{m,m-k}$, defined above. Once again this coincides with the Ledermann matrix when $k = 1$, but for $k > 1$ we have another type of L matrix which is distinct from the Type I L matrices defined above. At this stage an explicit specification of Type II L matrices has not been found.

Type III

In this case our triple $(m, n, k) \in \mathbb{Z}_+^2 \times \mathbb{Z}$ is only constrained by m and n since we simply set $N(m) = m - 2$, and require that $n \leq N(m)$. Now define $\mathbf{V}_{m,N(m)} = (\mathbf{v}_1, \ldots, \mathbf{v}_{N(m)})$ where

$$\mathbf{v}_j = [\underbrace{0, \ldots, 0}_{j-1}, k, -1, 1-k, \underbrace{0, \ldots, 0}_{m-2-j}]' \quad \text{for} \quad 1 \leq j \leq N(m). \tag{12}$$

Taking the last n columns of the GS image of the above yields L matrices of Type III. Note that, unlike the Type I and Type II definitions, k is no longer restricted to the positive integers, provided it is real. The case $k = 0$ corresponds to the Ledermann matrix, otherwise we have another distinct class of L matrices.

It is easy to generalise Type III L^k-matrices further to allow for more free parameters. For example, the GS pre-image columns could be defined as

$$\mathbf{v}_j = [\underbrace{0, \ldots, 0}_{j-1}, k_1, -k_2, k_2 - k_1, \underbrace{0, \ldots, 0}_{m-2-j}]' \quad \text{for} \quad 1 \leq j \leq N(m). \tag{13}$$

This would then produce a matrix $\mathbf{L}_{mn}^{k_1,k_2}$ which solves (3) with the flexibility of two parameters. Continuing in this way, an abundance of interesting solutions can be found. Although some will be rather complex, every matrix constructed in this way is an L matrix.

1.2. Parametric L matrices

In this section we construct linearly independent sets in $\mathcal{H}$ from elliptical probability distributions. This method can be interpreted as Monte Carlo simulation, adjusted to achieve exact covariance. In this approach, the columns of our GS pre-image matrix $\mathbf{V}_{mn} = (\mathbf{v}_1, \ldots, \mathbf{v}_n)$ are random vectors. We assume that these vectors are drawn from a zero mean elliptical multivariate distribution, whose marginal components are independent. That is,

$$\mathbf{V}_{mn} \sim \mathrm{D}(\mathbf{0}_n, \mathbf{I}_n),$$

where D is an elliptical distribution. Of course, $\mathbf{V}_{mn}$ is just m random observations on the n random variables, whose population distribution is D. So in particular, the sample mean of our observations will only satisfy the approximation

$$\bar{\mathbf{v}}_n \approx \mathbf{0}_n,$$

and hence the columns of $\mathbf{V}_{mn}$ will not necessarily be members of $\mathcal{H}$. However, we can simply re-define

$$\mathbf{V}_{mn} \leftarrow \mathbf{V}_{mn} - \mathbf{1}_m \bar{\mathbf{v}}_n',$$

so that the column vectors of $\mathbf{V}_{mn}$ do indeed lie in $\mathcal{H}$.[9] We would like to transform these vectors into an orthonormal set using the Gram-Schmidt procedure. However we can only do this if our column vectors are linearly independent. Once again, although we assumed independence for our multivariate distribution D, the sample covariance matrix of $\mathbf{V}_{mn}$ will only satisfy the approximation

$$m^{-1}\mathbf{V}_{nm}'\mathbf{V}_{mn} \approx \mathbf{I}_n. \tag{14}$$

We now normalise the columns of $\mathbf{V}_{mn}$ by re-defining[10]

$$\mathbf{v}_j \leftarrow \mathbf{v}_j \langle \mathbf{v}_j, \mathbf{v}_j \rangle^{-1/2}.$$

Since the vectors of $\mathbf{V}_{mn}$ now have unit length, approximation (14) can be written more explicitly as

$$\mathbf{V}_{mn}'\mathbf{V}_{mn} = \mathbf{I}_n + \epsilon \mathbf{P}_n, \tag{15}$$

where ϵ is a small number, sometimes written as $\epsilon \ll 1$, and $\mathbf{P}_n$ is a square matrix with zeros on the diagonal. Due to sampling error, $\epsilon \neq 0$ almost always.

Recall that our aim is to show that the columns of $\mathbf{V}_{mn}$ are linearly independent. We first note that a key result in linear algebra states that the columns of a matrix, $\mathbf{V}_{mn}$, are linearly independent if and only if its corresponding Gramian matrix, $\mathbf{G}_V$, is non-singular, where

$$\mathbf{G}_V = \mathbf{V}_{mn}'\mathbf{V}_{mn}.$$

Therefore for the classification of linear independence we must calculate the determinant of the matrix $\mathbf{G}_V$. Now using (15), and a perturbation result for determinants, we know that

$$\det(\mathbf{G}_V) = \det(\mathbf{I}_n + \epsilon \mathbf{P}_n) = 1 + \mathrm{tr}(\mathbf{P}_n)\epsilon + \circ(\epsilon^2).$$

Since $\mathrm{tr}(\mathbf{P}_n) = 0$ we have that $\det(\mathbf{G}_V) = 1 + \circ(\epsilon^2) \neq 0$. Therefore the n column vectors of the matrix $\mathbf{V}_{mn} = (\mathbf{v}_1, \ldots, \mathbf{v}_n)$ are linearly independent and the Gram-Schmidt algorithm may be applied to them. If we let $\mathbf{L}_{mn}^P = GS(\mathbf{V}_{mn})$, then we have yet another L matrix.

If we use this matrix for exact covariance simulation, assuming a target mean of $\mathbf{0}_n$ and covariance matrix $\mathbf{I}_n$, then from (2) our sample will simply be

$$\mathbf{X}_{mn} = m^{1/2}\mathbf{L}_{mn}^P.$$

Since $\mathbf{L}_{mn}^P = GS(\mathbf{V}_{mn} - \mathbf{1}_m \bar{\mathbf{v}}_n')$ we have

$$\mathbf{V}_{mn} - \mathbf{1}_m \bar{\mathbf{v}}_n' = \mathbf{L}_{mn}^P \mathbf{R}_n, \tag{16}$$

[9] Other authors, such as Meucci [24], augment their samples using negative complements in order to achieve a zero mean sample.

[10] Normalisation at this step is applied to aid the argument of linear independence. In practise, it is superfluous at this point, since the Gram-Schmidt algorithm, which is employed later, includes a normalisation.

where $\mathbf{R}_n$ is an upper (or right) triangular matrix.[11] Now, from the approximation (14) and the orthogonality of $\mathbf{L}^P_{mn}$ we have $\mathbf{R}'_n\mathbf{R}_n \approx m\mathbf{I}_n$. But $\mathbf{I}_n$ is diagonal so $\mathbf{R}'_n\mathbf{R}_n \approx \mathbf{R}_n\mathbf{R}'_n$. We require the following lemma:

Lemma 1.1. *Suppose that* $\mathbf{U}_n$ *is an upper triangular matrix satisfying* $\mathbf{U}'_n\mathbf{U}_n = \mathbf{U}_n\mathbf{U}'_n$. *Then* $\mathbf{U}_n$ *is a diagonal matrix. In particular, if* $\mathbf{U}_n$ *is orthogonal, then* $\mathbf{U}_n = \text{diag}(\pm 1, \ldots, \pm 1)$.

Proof. See [26, Chapter III, Section 17], for example. □

Using an approximate form of Lemma 1.1 we deduce that $\mathbf{R}_n \approx \text{diag}(\pm\sqrt{m}, \ldots, \pm\sqrt{m})$. Now, by inspecting the Gram-Schmidt procedure further, we find that the diagonal elements of $\mathbf{R}_n$ must be positive. Hence, $\mathbf{R}_n \approx m^{1/2}\mathbf{I}_n$ and, after inverting (16), we arrive at the approximation

$$\mathbf{X}_{mn} \approx \mathbf{V}_{mn} - \mathbf{1}_m\bar{\mathbf{v}}'_n.$$

Hence $\mathbf{X}_{mn}$ is in fact very close to mean deviations of our original sample $\mathbf{V}_{mn}$. This shows that parametric L matrices generate simulations that are a small adjustment to standard parametric Monte Carlo simulations.

1.3. Data-specific L matrices

We now introduce a method for exact covariance simulation when the covariance matrix is estimated from existing data. It is applicable to both cross-section and time-series data, but for the sake of exposition we shall assume that we observe a multivariate time series on n random variables. The advantage of this approach is that the properties of an exact moment simulation will closely resemble the corresponding properties of the original sample, if one exists.

Suppose that we have a sample matrix $\mathbf{Y}_{mn}$ representing m historical observations on n random variables. The columns of this matrix need not have zero sample mean, so define

$$\mathbf{V}_{mn} = \mathbf{Y}_{mn} - \mathbf{1}_m\bar{\mathbf{y}}'_n,$$

whose columns will lie in $\mathcal{H}$. We assume the columns of $\mathbf{V}_{mn}$ are linearly independent.[12] Clearly, $\mathbf{L}^D_{mn} = GS(\mathbf{V}_{mn})$ is an L matrix. Now, given a target mean $\boldsymbol{\mu}_n$ and covariance matrix $\mathbf{S}_n$, we use this data-specific solution to form an exact sample

$$\mathbf{X}_{mn} = m^{1/2}\mathbf{L}^D_{mn}\mathbf{A}_n + \mathbf{1}_m\boldsymbol{\mu}'_n,$$

where $\mathbf{A}'_n\mathbf{A}_n = \mathbf{S}_n$.

It is straightforward to show that $\mathbf{X}_{mn} = \mathbf{Y}_{mn}$ when the target moments $\boldsymbol{\mu}_n$ and $\mathbf{S}_n$ are chosen to match the moments of the existing data $\mathbf{Y}_{mn}$.[13] However one may generate exact covariance simulated samples that differ from the existing sample by targeting moments that are different from the sample moments. For instance, one very useful application to financial risk management is to "stress" the elements of a covariance matrix, while keeping other, less quantifiable characteristics of the existing data unchanged.

1.4. Hybrid L matrices

A data-specific solution $\mathbf{L}^D_{mn}$ can be combined with deterministic and/or parametric L matrices to produce "hybrid" L matrices which reflect both existing and hypothetical properties of the data. Of course, in situations where little or no existing data are available only deterministic and parametric L matrices are applicable. For instance, we may construct a hybrid L matrix which utilises both the deterministic and parametric L matrix methodologies, using the following lemma (its proof is direct).

[11] This is an example of a *Thin QR Decomposition*, see [11].

[12] If any variable in the system is a linear combination of other variables then it is not necessary to simulate this variable separately, and it should be removed.

[13] This assumes that the covariance matrix of $\mathbf{Y}_{mn}$ is positive definite.

Lemma 1.2. *Suppose we have k rectangular matrices* $\mathbf{Q}_{mn}^1, \ldots, \mathbf{Q}_{mn}^k \in \mathbb{R}^{mn}$ *satisfying*

$$(\mathbf{Q}_{mn}^i)'(\mathbf{Q}_{mn}^j) = \delta_{ij}\mathbf{I}_n \quad \text{for all} \quad 1 \le i, j \le k,$$

where δ_{ij} *is the Kronecker delta function. Then, given k non-zero scalars* $\alpha_1, \ldots, \alpha_k \in \mathbb{R}$*, the linear combination of matrices*

$$\mathbf{S}_{mn} = \left(\sum_{i=1}^{k} \alpha_i^2\right)^{-\frac{1}{2}} \sum_{i=1}^{k} \alpha_i \mathbf{Q}_{mn}^i,$$

is a rectangular orthogonal matrix.

Let $\mathbf{L}_{mn}$ be a deterministic L matrix, which is constructed using one of the techniques from Section 1.1. Also let $\mathbf{V}_{mn}$ be a random sample from an elliptical distribution $\mathrm{D}(\mathbf{0}_n, \mathbf{I}_n)$. Assuming that $\mathbf{V}_{mn}$ has been adjusted to have a zero sample mean, we construct the augmented matrix

$$\mathbf{A}_{m,2n} = (\mathbf{L}_{mn}, \mathbf{V}_{mn}).$$

We then apply the Gram-Schmidt procedure, setting $\hat{\mathbf{A}}_{m,2n} = GS(\mathbf{A}_{m,2n})$ where

$$\hat{\mathbf{A}}_{m,2n} = (\hat{\mathbf{L}}_{mn}, \hat{\mathbf{V}}_{mn}).$$

Because of the sequential nature of the Gram-Schmidt algorithm, and since $\mathbf{L}_{mn}$ is already orthogonal, we know that $\hat{\mathbf{L}}_{mn} = \mathbf{L}_{mn}$. Furthermore, the matrix $\hat{\mathbf{V}}_{mn}$ will not only be orthogonal but will also satisfy $\mathbf{L}_{mn}'\hat{\mathbf{V}}_{mn} = \mathbf{0}$. Therefore we can apply Lemma 1.2, so that for any ϵ the matrix

$$\mathbf{L}_{mn}^P = \frac{1}{\sqrt{1+\epsilon^2}}\left(\mathbf{L}_{mn} + \epsilon\hat{\mathbf{V}}_{mn}\right), \tag{17}$$

will be rectangular orthogonal. Since the columns of $\mathbf{A}_{m,2n}$ lie in $\mathcal{H}$, the subspace preserving property of the Gram-Schmidt process ensures that the columns of $\hat{\mathbf{A}}_{m,2n}$ are also in $\mathcal{H}$. Therefore $\mathbf{L}_{mn}^P$ defined by (17) is another L matrix. The parameter ϵ controls the mix between deterministic and random structure in the matrix. If it is small the hybrid L matrix approximates the deterministic characteristics of $\mathbf{L}_{mn}$.

2. ROM simulation

The motivation for the research presented in this section is that, if $\mathbf{L}_{mn}$ is an L matrix then so is $\mathbf{L}_{mn}\mathbf{R}_n$, for any square orthogonal matrix $\mathbf{R}_n$. Also, if $\mathbf{Q}_m$ is a permutation matrix then $\mathbf{Q}_m\mathbf{L}_{mn}$ is an L matrix, since $\mathbf{1}_m'\mathbf{Q}_m = \mathbf{1}_m'$. Together, these two observations lay the foundation for a new approach to simulating infinitely many random samples that have identical sample mean vectors and covariance matrices. Each random sample is generated as

$$\mathbf{X}_{mn} = \mathbf{1}_m\boldsymbol{\mu}_n' + \sqrt{m}\mathbf{Q}_m\mathbf{L}_{mn}\mathbf{R}_n\mathbf{A}_n, \tag{18}$$

where $\mathbf{L}_{mn}$ is an L matrix, $\mathbf{Q}_m$ is a random permutation matrix, $\mathbf{R}_n$ is any random orthogonal matrix, $\boldsymbol{\mu}_n$ is the target mean vector and $\mathbf{A}_n'\mathbf{A}_n = \mathbf{S}_n$, where $\mathbf{S}_n$ is the target covariance matrix. Thus, (18) provides a novel and very flexible method for generating random samples.[14]

Any deterministic, parametric or data-specific L matrix of type I, II or III may be applied in (18) but in this section for simplicity we shall work with the Ledermann matrix (9), since it is available in an elegant closed form. The purpose of this section is to investigate how the characteristics of the simulated sample are affected when we choose different types of random orthogonal matrices $\mathbf{R}_n$ in (18).

[14] Many thanks to Prof. Nick Higham for recently pointing us towards the paper by Li [18] where a special case of (18) is proposed in which $n = m - 1$, $\mathbf{Q}_m$ is the identity matrix and $\mathbf{L}_{mn}$ is the matrix (8), or any matrix such that $(m^{-1/2}\mathbf{1}_m, \mathbf{L}_{mn})$ is an orthogonal matrix.

2.1. Skewness and kurtosis

There are many ways to measure the skewness and kurtosis, even of a univariate sample, as discussed by Cramer [6]. For our applications we require a statistic for the skewness and kurtosis of a multivariate sample, and there are also many different definitions of multivariate skewness and kurtosis, see [15]. In the following we employ the definitions introduced by Mardia [23]. However, as highlighted by Gutjahr et al. [12], samples from different distributions may have identical Mardia characteristics. Even so, we choose to work with Mardia's multivariate moments since, as scalar-valued measurements, they are highly tractable.

Given a random sample $\mathbf{X}_{mn} = (\mathbf{x}_1', \ldots, \mathbf{x}_m')'$, written using row vector notation where $\mathbf{x}_i = (x_{i1}, \ldots, x_{in})$ for $1 \le i \le m$, Mardia [23] proposes the following measures for the multivariate skewness and kurtosis of $\mathbf{X}_{mn}$:

$$\tau_M(\mathbf{X}_{mn}) = m^{-2} \sum_{i=1}^{m} \sum_{j=1}^{m} \left\{ (\mathbf{x}_i - \bar{\mathbf{x}}) \mathbf{S}_X^{-1} (\mathbf{x}_j - \bar{\mathbf{x}})' \right\}^3,$$

$$\kappa_M(\mathbf{X}_{mn}) = m^{-1} \sum_{i=1}^{m} \left\{ (\mathbf{x}_i - \bar{\mathbf{x}}) \mathbf{S}_X^{-1} (\mathbf{x}_i - \bar{\mathbf{x}})' \right\}^2$$

where $\mathbf{S}_X$ is the $n \times n$ sample covariance matrix of $\mathbf{X}_{mn}$ and $\bar{\mathbf{x}}$ is the row vector of sample means. Mardia [23] proves that these measures are invariant under non-singular, affine transformations. That is, $\tau_M(\mathbf{Y}_{mn}) = \tau_M(\mathbf{X}_{mn})$ and $\kappa_M(\mathbf{Y}_{mn}) = \kappa_M(\mathbf{X}_{mn})$, whenever

$$\mathbf{Y}_{mn} = \mathbf{X}_{mn} \mathbf{B}_{nn} + \mathbf{1}_m \mathbf{b}_n, \tag{19}$$

where $\mathbf{B}_{nn}$ is any invertible matrix and $\mathbf{b}_n$ is any row vector. This invariance property is of particular significance in ROM simulation, since the transformation (18) is a (random) non-singular affine transformation of the L matrix $\mathbf{L}_{mn}$. Therefore $\tau_M(\mathbf{L}_{mn})$ and $\kappa_M(\mathbf{L}_{mn})$ will determine the multivariate skewness and kurtosis of all ROM simulated samples $\mathbf{X}_{mn}$.

Proposition 2.1. *Let $\mathbf{L}_{mn}$ be the Ledermann matrix, defined by (9). Then*

$$\tau_M(\mathbf{L}_{mn}) = n\Big[(m-3) + (m-n)^{-1}\Big] =: f(m,n), \tag{20}$$

$$\kappa_M(\mathbf{L}_{mn}) = n\Big[(m-2) + (m-n)^{-1}\Big] =: g(m,n). \tag{21}$$

Proof. See Section A.1 of the appendix. □

By comparing Eqs. (21) and (20) we see that a very simple relationship links the multivariate kurtosis and multivariate skewness of the Ledermann matrix. That is, for $m > n$, we have

$$\kappa_M(\mathbf{L}_{mn}) - \tau_M(\mathbf{L}_{mn}) = n.$$

The expected skewness and kurtosis of MVN variables are also closely linked, since their ratio is a simple function of n and m. See Proposition 2.3 below.

We now study the asymptotic behaviour of the Mardia skewness and kurtosis derived from large samples that are generated using Ledermann matrices. For a fixed n and a comparatively large m, written $m \gg n$, we have that $(m-n)^{-1} \approx 0$. Hence we can approximate (20) and (21) as

$$f(m,n) \approx n(m-3), \qquad g(m,n) \approx n(m-2) \qquad \text{for} \qquad m \gg n. \tag{22}$$

So the functions $g(m,n)$ and $f(m,n)$ are asymptotically linear, with gradient n, and will become unbounded as m becomes large. It is also true that if $n_2 > n_1$ then $f(m,n_2) > f(m,n_1)$ for all $m > \max\{n_1, n_2\}$ and that $f(m,n_2)$ will tend to infinity faster than $f(m,n_1)$, as m tends to infinity, and similarly for $g(m,n)$.

Note that if we use a Ledermann matrix in ROM simulation then we can target either skewness or kurtosis but not both, by choosing an appropriate value m, since the number of variables n is typically fixed. To target both moments we shall need to use a generalised L matrix, with at least one additional

parameter. Before we discuss this we remark that higher moments targeting inadvertently places a constraint on the size m of a single simulation. In many cases, the value of m which achieves a desired skewness or kurtosis level will be much smaller than the number of observations required for a standard simulation problem. The obvious solution is to repeat simulations of size m until enough observations have been generated, and we refer to this technique as sample concatenation. Each small simulation will have the desired mean, covariance matrix and multivariate skewness or kurtosis but what can we say about the first four multivariate moments of the concatenated sample?

Definition 2.1. Suppose that we have two different samples $\mathbf{X}_{m_X,n} = (\mathbf{x}_1', \ldots, \mathbf{x}_m')'$ and $\mathbf{Y}_{m_Y,n} = (\mathbf{y}_1', \ldots, \mathbf{y}_m')'$, on the same n random variables. We define their multivariate *co-skewness* as

$$\tau_C(\mathbf{X}_{m_X,n}, \mathbf{Y}_{m_Y,n}) = (m_X + m_Y)^{-2} \sum_{i=1}^{m_X} \sum_{j=1}^{m_Y} \left\{ 2(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{S}_X + \mathbf{S}_Y)^{-1}(\mathbf{y}_j - \bar{\mathbf{y}})' \right\}^3, \tag{23}$$

where $\bar{\mathbf{x}}, \bar{\mathbf{y}}$ and $\mathbf{S}_X, \mathbf{S}_Y$ are the sample means and covariance matrices of $\mathbf{X}_{m_X,n}$,$\mathbf{Y}_{m_X,n}$ respectively. In can be shown that τ_C is invariant under certain non-singular affine transformations. See Appendix A.2.

Proposition 2.2. *Consider r random samples $\mathbf{X}_{m_1,n}, \ldots, \mathbf{X}_{m_r,n}$, each with sample mean $\boldsymbol{\mu}_n$ and sample covariance matrix $\mathbf{S}_n$. Set $m = \sum_{k=1}^r m_k$ and define $\mathbf{X}_{mn} = (\mathbf{X}_{m_1,n}', \ldots, \mathbf{X}_{m_r,n}')'$. Then*

$$m^{-1}\mathbf{1}_m'\mathbf{X}_{mn} = \boldsymbol{\mu}_n' \quad \text{and} \quad m^{-1}\mathbf{X}_{nm}'\mathbf{X}_{mn} = \mathbf{S}_n,$$

with the multivariate skewness and kurtosis satisfying,

$$\tau_M(\mathbf{X}_{mn}) = m^{-2} \sum_{k=1}^r m_k^2 \tau_M(\mathbf{X}_{m_k,n}) + 2m^{-2} \sum_{k<l}^r (m_k + m_l)^2 \tau_C(\mathbf{X}_{m_k,n}, \mathbf{X}_{m_l,n}),$$

$$\kappa_M(\mathbf{X}_{mn}) = m^{-1} \sum_{k=1}^r m_k \kappa_M(\mathbf{X}_{m_k,n}).$$

Proof. See Appendix A.3. □

The first half of Proposition 2.2 states that sample means and covariance matrices are preserved under sample concatenation, provided that the smaller samples that build the concatenation have identical sample means and covariances. In particular, a large exact moment simulation, which targets a covariance matrix $\mathbf{S}_n$, can be constructed by concatenating many smaller exact moment simulations, each with the same target covariance matrix $\mathbf{S}_n$.

However, to draw similar conclusions from the second half of Proposition 2.2, regarding concatenated skewness and kurtosis, we must be more restrictive. Firstly, for kurtosis to be preserved under concatenation, our small samples must have equal size and have identical kurtosis. In a ROM simulation, this amounts to using the same L matrix for each small simulation. That is,

$$\mathbf{X}_{rm,n} = \mathbf{1}_{rm}\boldsymbol{\mu}_n' + \sqrt{rm}\left((\mathbf{L}_{mn}\mathbf{R}_n^{(1)})', \ldots, (\mathbf{L}_{mn}\mathbf{R}_n^{(r)})' \right)' \mathbf{A}_n. \tag{24}$$

So even for different orthogonal matrices $\mathbf{R}_n^{(1)}, \ldots, \mathbf{R}_n^{(r)}$ our large sample $\mathbf{X}_{rm,n}$ will have mean $\boldsymbol{\mu}_n$, covariance matrix $\mathbf{A}_n'\mathbf{A}_n = \mathbf{S}_n$ and kurtosis $\kappa_M(\mathbf{X}_{rm,n}) = \kappa_M(\mathbf{L}_{mn})$, where $\kappa_M(\mathbf{L}_{mn})$ is calibrated via m.

The behaviour of multivariate skewness is more complex, due to the presence of co-skewness terms. Noting that

$$\tau_C(\mathbf{X}_{mn}, \mathbf{X}_{mn}) = \tau_M(\mathbf{X}_{mn})/4,$$

Proposition 2.2 informs us that to ensure that $\tau_M(\mathbf{X}_{rm,n}) = \tau_M(\mathbf{L}_{mn})$ we must concatenate ROM simulations of the form

$$\mathbf{X}_{rm,n} = \mathbf{1}_{rm}\boldsymbol{\mu}_n' + \sqrt{rm}(\underbrace{\mathbf{L}_{nm}', \ldots, \mathbf{L}_{nm}'}_{r \text{ times}})'\mathbf{R}_n\mathbf{A}_n. \tag{25}$$

Since (25) is a special case of (24), $\mathbf{X}_{mn}$ will still have sample mean $\boldsymbol{\mu}_n$ and covariance matrix $\mathbf{S}_n$, with multivariate kurtosis satisfying $\kappa_M(\mathbf{X}_{rm,n}) = \kappa_M(\mathbf{L}_{mn})$.

We have defined a method for generating large samples of size rm, with exact target first and second moments and third and fourth moments that are determined by the skewness and kurtosis of the L matrix used in (25). With n fixed, the skewness and kurtosis of a matrix $\mathbf{L}^k_{mn}$ is a function of m and k. In general, these functions are not available in closed form, so a numerical optimisation must be applied to calibrate the values of m and k which achieve the target skewness and kurtosis. With these moment targeted, a simulation of the form (25) ensures that these moments are preserved in large simulations.

We now study the skewness and kurtosis of ROM simulations which employ parametric L matrices, as defined in Section 1.2. Recall that a parametric L matrix $\mathbf{L}^P_{mn}$ is a Gram-Schmidt orthogonalisation of a random sample $\mathbf{V}_{mn}$, drawn from an elliptical multivariate distribution. One can show this orthogonalisation process in an invertible affine transformation and hence the skewness and kurtosis of $\mathbf{L}^P_{mn}$ will equal the skewness and kurtosis of $\mathbf{V}_{mn}$. Unlike the deterministic case, $\tau_M(\mathbf{L}^P_{mn})$ and $\kappa_M(\mathbf{L}^P_{mn})$ depend on a random sample $\mathbf{V}_{mn}$, and will therefore be random quantities themselves. However, if $\mathbf{V}_{mn}$ is drawn from a multivariate normal distribution, then we can use the following result of Mardia [23], to find the expected values of $\tau_M(\mathbf{L}^P_{mn})$ and $\kappa_M(\mathbf{L}^P_{mn})$.

Proposition 2.3. *Let $\mathbf{V}_{mn}$ be a random sample drawn from any multivariate normal distribution $N(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$. Then*

$$\mathbb{E}[\tau_M(\mathbf{V}_{mn})] = n(n+1)(n+2)m^{-1},$$
$$\mathbb{E}[\kappa_M(\mathbf{V}_{mn})] = n(n+2)(m-1)(m+1)^{-1}.$$

Therefore for large m the skewness and kurtosis of a parametric L matrix generated from a multivariate normal distribution will be:

$$\mathbb{E}\left[\tau_M(\mathbf{L}^P_{mn})\right] \approx 0,$$
$$\mathbb{E}\left[\kappa_M(\mathbf{L}^P_{mn})\right] \approx n(n+2).$$

Hence, the skewness and kurtosis in a parametric ROM simulation is restricted by the underlying distribution, and these moments are also prone to sampling error.

By contrast, when using deterministic L matrices in ROM simulations one can target a range of different skewness and kurtosis levels. There will also be no sampling error since the deterministic values $\tau_M(\mathbf{L}_{mn})$ and $\kappa_M(\mathbf{L}_{mn})$ fix the multivariate skewness and kurtosis of all ROM simulated samples. One can also concatenate deterministic L matrix ROM simulations and parametric L matrix ROM simulations and the moments of this concatenated sample may be estimated using Proposition 2.2.

Example 1. Concatenating deterministic and parametric ROM simulations. Using the same target mean and covariance matrix, suppose we use ROM simulation to generate two samples $\mathbf{X}_{m_1,n}$ and $\mathbf{X}_{m_2,n}$. The first sample is simulated from the Ledermann matrix and the second sample is a simulated from a normally distributed parametric L matrix. Assuming that m_1 and m_2 are large, the multivariate kurtosis of the concatenated sample $\mathbf{X}_{mn} = (\mathbf{X}'_{m_1,n}, \mathbf{X}'_{m_2,n})'$, has approximate expected value

$$\mathbb{E}[\kappa_M(\mathbf{X}_{mn})] \approx (m_1+m_2)^{-1}\left\{m_1 n(m_1-2) + m_2 n(n+2)\right\}.$$

Concatenation is not the only way of combining deterministic and parametric ROM simulations. One can also use perturbed L matrices, as defined in (17).

Example 2. Kurtosis of perturbed L matrices. Let $\mathbf{L}^P_{mn}$ be a ϵ-perturbation of the deterministic L matrix $\mathbf{L}_{mn}$. Then, provided that ϵ is small, it is straightforward to derive the approximation

$$\kappa_M(\mathbf{L}^P_{mn}) \approx \frac{\kappa_M(\mathbf{L}_{mn})}{(1+\epsilon^2)^2}.$$

Finally, we note that the skewness and kurtosis of data-specific L matrices are completely determined by the underlying data. So, similar to parametric ROM simulations, the higher moments of data-specific ROM simulations are limited. However, if desired, one can affect these limits while maintaining some characteristics of the existing data by using a concatenation or perturbation approach.

2.2. Orthogonal matrices and sample characteristics

In Section 2.1 we used the multivariate skewness and kurtosis measures of Mardia [23] to quantify how different L matrices affect ROM simulations. We now turn our attention to the random square orthogonal matrices which pre- or post-multiply a given L matrix during a ROM simulation. Geometrically, square orthogonal matrices fall into three categories: permutations, reflections and rotations. Methods for generating such matrices are numerous, particularly in the latter case. We choose to focus on rotation matrices which appear in upper Hessenberg form. However techniques involving skew-symmetric matrices and [3] transforms, or the matrix exponential function, may also be used to generate random, rotation matrices, see [5]. For the sake of clarity and because this matrix has a relatively simple explicit form, $\mathbf{L}_{mn}$ will again denote the Ledermann matrix defined by (9), unless otherwise specified. Continuing the notation of (18), we shall now investigate how random permutation, reflection and other random orthogonal matrices can alter the characteristics of multivariate time series ROM simulations.

Permutation matrices are the primary means of controlling the time ordering of observations in multivariate time series ROM simulation. Such a matrix, denoted by $\mathbf{Q}_m$, is a square matrix of order m, which has exactly one entry 1 in every row and every column, and has entries 0 everywhere else. Pre-multiplying any $m \times n$ matrix $\mathbf{A}_{mn}$ with $\mathbf{Q}_m$ permutes the rows of $\mathbf{A}_{mn}$. For example,

$$\mathbf{Q}_m\mathbf{A}_{mn} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} = \begin{pmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \\ a_{11} & a_{12} \end{pmatrix}.$$

To see how permutation matrices can affect ROM simulations we first recall the structure of $\mathbf{L}_{mn}$, inferred from (8). From this we see that the largest values in $\mathbf{L}_{mn}$ appear on the diagonal of it's lowest $n \times n$ block. Therefore, when the Ledermann matrix is applied to ROM simulations without permutations, large values only appear at the end of the simulated sample paths. However, by including a random permutation matrix $\mathbf{Q}_m$ in (18) we can ensure that these large values are randomly positioned throughout the whole sample path. Furthermore, by restricting the permutation matrix, and hence controlling the rows of the L matrix that are fixed and those that are randomly permuted, we can influence the random times at which these large values appear.

In time series analysis one regularly observes areas of large changes (positive or negative) followed by areas of small changes. This phenomenon is referred to as volatility clustering. [15] *Cyclic* permutation matrices enable us to produce ROM samples which exhibit this volatility clustering feature. When a random cyclic permutation matrix pre-multiplies a matrix the rows of the matrix are all offset downwards by means of a random shift, thus maintaining their ordering. By using *random* cyclic permutations in ROM simulation the large values of the Ledermann matrix can be made to appear together in a random cluster. Alternatively, to remove the volatility clustering that is implicit in the L matrix, a general random permutation matrix can be used.

To illustrate these effects, Fig. 1 exhibits plots for three different simulated sample paths. Each simulation has the form (18), with $\mathbf{R}_n$ fixed as the identity. However, in the top path $\mathbf{Q}_m$ is also the identity, in the middle path $\mathbf{Q}_m$ is a random cyclic permutation and in the bottom path $\mathbf{Q}_m$ is a general random permutation matrix.

The assertion in (18) that $\mathbf{Q}_n$ must be a permutation matrix, while $\mathbf{R}_n$ can be any orthogonal matrix, is made for the following reason. Let $\mathbf{X}_{mn}$ be a ROM simulated sample, generated from $\mathbf{L}_{mn}$. In order for the sample mean of $\mathbf{X}_{mn}$ to be $\boldsymbol{\mu}_n$, we must have that

$$\mathbf{1}'_m\mathbf{Q}_m\mathbf{L}_{mn}\mathbf{R}_n\mathbf{A}_n = \mathbf{0}'_n. \tag{26}$$

[15] In finance, these volatility clusters are often apparent in historical returns series, see [20].

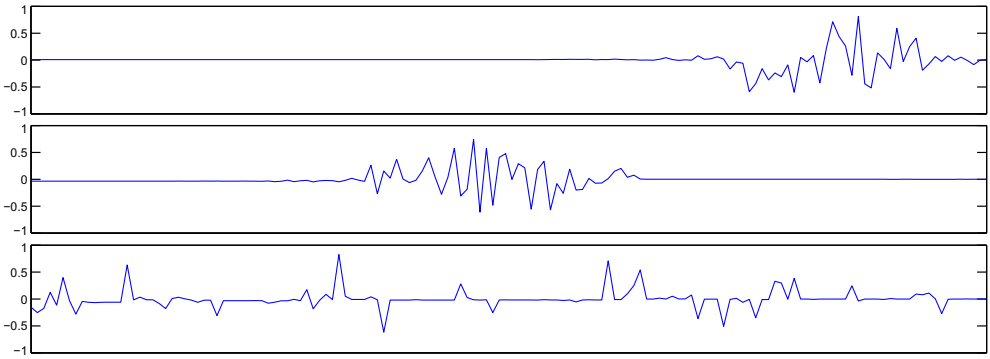

Fig. 1. Three plots demonstrating how permutation matrices affect volatility clustering in sample paths simulated from the Ledermann matrix.

Now, since $\mathbf{L}_{mn}$ is an L matrix, given by (3), we know that $\mathbf{1}'_m\mathbf{L}_{mn} = \mathbf{0}'_n$. If $\mathbf{Q}_m$ is a permutation matrix then $\mathbf{1}'_m\mathbf{Q}_m = \mathbf{1}'_m$ and (26) will hold regardless of the form of $\mathbf{R}_n$. However, this may not be the case if $\mathbf{Q}_m$ is not a permutation matrix. So now, assuming that $\mathbf{Q}_m$ is a permutation matrix, we discuss possible choices for the random orthogonal matrix $\mathbf{R}_n$.

Although we have examined the multivariate kurtosis of ROM simulations, as defined by Mardia [23], we have not yet discussed marginal kurtosis, that is, the kurtosis of the individual variables within the simulation. Since permutation matrices only re-order our samples, we must find orthogonal matrices which can affect the magnitude of our simulated observations. An *upper Hessenberg* matrix $\mathbf{H}_n$ of degree n is a matrix with zero entries below the first sub-diagonal. Berry et al. [2], Gemignani [8] and many other authors show that we can construct orthogonal upper Hessenberg matrices using Givens rotation matrices. A Givens rotation $\mathbf{G}_j(\theta_j)$ is the identity matrix except for the 2×2 principal sub-matrix

$$\mathbf{G}_j[j, j+1; j, j+1] = \begin{pmatrix} \cos(\theta_j) & \sin(\theta_j) \\ -\sin(\theta_j) & \cos(\theta_j) \end{pmatrix}. \tag{27}$$

Orthogonal upper Hessenberg matrices $\mathbf{H}_n$ can be constructed as a product of $n-1$ Givens rotations via

$$\mathbf{H}_n = \mathbf{G}_1(\theta_1)\mathbf{G}_2(\theta_2)\ldots\mathbf{G}_{n-1}(\theta_{n-1}). \tag{28}$$

In order to produce random orthogonal upper Hessenberg matrices we allow the arguments of (28) to be stochastic, choosing θ_j in the interval $[0, 2\pi)$ for $1 \le j \le n-1$ at random.[16] An example of a low-dimensional Hessenberg matrix is given below, where we simplify notation by setting $c_i = \cos(\theta_i)$ and $s_i = \sin(\theta_i)$ for $1 \le i \le 3$.

$$\mathbf{H}_4 = \begin{pmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c_2 & -s_2 & 0 \\ 0 & s_2 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & c_3 & -s_3 \\ 0 & 0 & s_3 & c_3 \end{pmatrix}$$

$$= \begin{pmatrix} c_1 & -s_1c_2 & s_1s_2c_3 & -s_1s_2s_3 \\ s_1 & c_1c_2 & -c_1s_2c_3 & c_1s_2s_3 \\ 0 & s_2 & c_2c_3 & -c_2s_3 \\ 0 & 0 & s_3 & c_3 \end{pmatrix}.$$

[16] Random orthogonal $\mathbf{H}_n$ can be generated quickly in MatLab using the function `gallery('randhess',n)`, see [13].

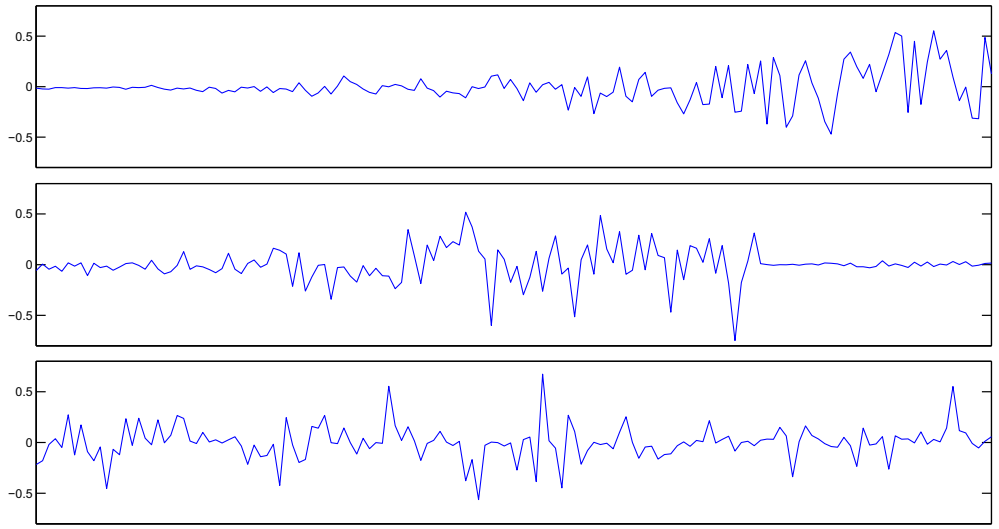

Fig. 2. Three plots demonstrating how upper Hessenberg and permutation matrices affect the characteristics of sample paths simulated from the Ledermann matrix.

In ROM simulation we can use products of random upper Hessenberg matrices to form the orthogonal matrix $\mathbf{R}_n$, which post-multiplies the L matrix appearing in (18). Recall that the magnitudes of the elements of an L matrix depend on its degree, which depends on the number of observations m we want to simulate. When m is large, the large elements on the lowest diagonal of the L matrix create a "pulse", where clusters of large positive and negative observations are interspersed with "flat" periods of less activity in the sample (see the third plot in Fig. 1 for example). Permutation matrices only control the timing of such pulses, but we may also like to affect their magnitude and have the potential to eliminate the flat periods. Crucially, unlike permutation matrices, post-multiplying an L matrix with many random Hessenberg matrices diminishes and dissipates the effect of the large elements, randomly throughout the sample. Fig. 2 shows how the three samples paths shown in Fig. 1 are affected by post-multiplication with random Hessenberg matrices.

The number of Hessenberg matrices used in the orthogonal product affects the marginal kurtosis of the random variables. The bottom $n \times n$ block of an L matrix $\mathbf{L}_{mn}$ is upper triangular. Furthermore, the number of non-zero elements in each column of $\mathbf{L}_{mn}$ increases by one every time we post-multiply $\mathbf{L}_{mn}$ by a random upper Hessenberg matrix. Therefore, to remove all zeros from $\mathbf{L}_{mn}$ we must include the product of at least $n - 1$ upper Hessenberg matrices in the random orthogonal matrix $\mathbf{R}_n$ of (18).

Having looked at multivariate skewness in Section 2.1, we now show how to induce some degree of positive or negative marginal skewness into ROM simulations. For this we will use an orthogonal matrix $\mathbf{B}_n$ of the form

$$\mathbf{B}_n = \operatorname{diag}\left\{(-1)^{d_1}, \ldots, (-1)^{d_n}\right\}, \tag{29}$$

where each index d_i is either zero or one. When the indices are realisations of Bernoulli variables, we call such a matrix a random *sign* matrix. It will multiply the rows of any matrix it pre-multiplies by plus or minus one.

Suppose $\mathbf{R}_n$ is chosen in (18) to be a product of random upper Hessenberg matrices. This matrix pre-multiplies a matrix $\mathbf{A}_n$, which realises the target covariance matrix $\mathbf{S}_n$. We write this product as

$$\mathbf{T}_n = \mathbf{R}_n\mathbf{A}_n.$$

Now $\mathbf{T}_n$ will be a random matrix containing both positive and negative entries. Since $\mathbf{T}_n$ post-multiplies $\mathbf{L}_{mn}$ in (18), large positive elements of $\mathbf{T}_n$ will combine with the large negative diagonal elements of the L matrix to form large negative observations in our simulated sample matrix $\mathbf{X}_{mn}$. Similarly, large negative elements of $\mathbf{T}_n$ cause large positive observations in the sample. However, pre-multiplying $\mathbf{T}_n$

by a random sign matrix $\mathbf{B}_n$ affects these large negative or positive values.[17] Without this extra pre-multiplication, large positive and negative elements appear equally often in $\mathbf{T}_n$, resulting in a simulated sample which is approximately symmetric. A judicious choice for the Bernoulli probabilities can ensure that the product $\mathbf{B}_n\mathbf{T}_n$ has more large negative elements then large positive elements (or vice versa). Thus positive (or negative) marginal skewness is introduced into ROM simulated samples.[18]

There are many ways to influence skewness by altering the Bernoulli probabilities. For instance, letting $\mathbf{t}_i$ denote the ith row of $\mathbf{T}_n$, if $\mathbf{T}_n$ has at least one negative entry we can set

$$p_i = \left| \frac{\min_j \{\min(t_{ij}, 0) : t_{ij} \in \mathbf{t}_i\}}{\min_{i,j} \{t_{ij} : t_{ij} \in \mathbf{T}_n\}} \right|. \tag{30}$$

Then, randomly selecting $r \in (0, 1)$, assign:

$$d_i = \begin{cases} 1, & \text{if } r < p_i \\ 0, & \text{otherwise} \end{cases} \quad \text{for } 1 \leq i \leq n.$$

If there is a large negative entry in row $\mathbf{t}_i$ then the Bernoulli probability p_i will be close to one. Therefore the index d_i is more likely to take the value 1 and hence row $\mathbf{t}_i$ will change sign when $\mathbf{T}_n$ is pre-multiplied by $\mathbf{B}_n$. Thus this particular Bernoulli sign matrix converts large negative elements of $\mathbf{T}_n$ to large positive elements. As a result negative marginal skewness is introduced into simulations. Positive marginal skewness can be introduced by defining the p_i using maxima rather than minima in (30).

This section has shown how different classes of orthogonal matrices affect the characteristics of ROM simulated samples in very different ways. Although we have focussed on the Ledermann matrix, the above techniques can also be applied to ROM simulations involving other L matrices. Investigating these more general ROM simulations, and exploring further classes of orthogonal matrices, is left to future research.

3. Application to value-at-risk

ROM simulation has applications to virtually any problem that may be resolved using Monte Carlo simulation. Generally, such problems require the forecasting of future multivariate distributions, using historical or scenario sample data. In finance these problems include the pricing and hedging of complex options, any decision problem based on a future distribution such as portfolio optimisation, and financial risk assessment where, since its introduction during the 1990's, the value-at-risk (VaR) of a portfolio has become the industry benchmark. This section demonstrates how ROM simulation may be applied to VaR estimation.

The portfolio VaR is the loss that would be equalled or exceeded, with a given probability α, over a certain period of time (which is called the "risk horizon") if the portfolio is left un-managed during this interval. Measured as a percentage of the current portfolio value, the $100\alpha\%$ h-day VaR is minus the α-quantile of the distribution of the portfolio's h-day returns, in present value terms. For example, if the h-day excess returns on the portfolio's risk factors have a multivariate normal distribution and the portfolio return is a linear function of it's risk factor returns, the $100\alpha\%$ h-day VaR, expressed as a percentage of the current portfolio value is:[19]

$$\text{VaR}(\alpha, h) = -\Phi^{-1}(\alpha)\sigma_h \tag{31}$$

where $\Phi^{-1}(\alpha)$ is the lower α-quantile of a standard normal distribution and σ_h is the standard deviation of the normal h-day portfolio returns distribution.

However, a multivariate normal assumption is not usually appropriate and many portfolios have returns that are non-linear functions of the risk factor returns, so it is very common to use simulation to

[17] Algebraically, we are simply including a sign matrix into the Hessenberg product which comprises $\mathbf{R}_n$.

[18] If there are more large positive elements in $\mathbf{B}_m\mathbf{T}_{mn}$ than large negative elements, the samples will have negative skewness. If there are more large negative elements in $\mathbf{B}_m\mathbf{T}_{mn}$ than large positive elements, the samples will have positive skewness.

[19] This formula applies if the portfolio is expected to return the discount rate over the risk horizon. Otherwise, a small mean adjustment applies. See [1, Section IV.1.5].

estimate portfolio VaR. Monte Carlo simulation assumes a parametric multivariate risk factor returns distribution and thus generates i.i.d. simulations on each risk factor. To each simulation we then apply the portfolio mapping, which gives the portfolio value as a function of the returns on its risk factors, to obtain a simulated h-day return. Repeating the simulations many times generates a distribution for the portfolio's h-day return and the Monte Carlo VaR, expressed as a percentage of the portfolio value, is estimated as minus the α-quantile of this distribution. However, in addition to the sampling error described in the introduction, the parametric assumptions underlying Monte Carlo simulation pose a significant source of model risk. That is, different distributional assumptions typically give rise to quite different VaR estimates. It is well known that the empirical characteristics of financial asset returns are often misrepresented by multivariate normal or Student-t distributions, yet it is usually not clear what distributional assumption is most appropriate.

For these reasons historical simulation is by far the most common VaR resolution method used in the finance industry. Standard historical simulation uses a large number of historical returns on the portfolio to build an empirical distribution and hence estimate the VaR from a lower quantile. An advantage is that it does not make parametric assumptions regarding the risk factor returns; a major disadvantage is data limitation. A long historical sample may be difficult to obtain for all risk factors, or may be thought to be inappropriate since market behaviour has changed considerably in recent years. Thus the empirical distribution is typically built using only 3–5 years of daily returns, i.e. a sample size of around 750–1250 data points, so the accuracy of the quantile estimator is severely impaired.[20] Moreover, due to lack of sufficient weekly or monthly historical data, 1-day historical VaR estimates are usually scaled up to h-day estimates for $h > 1$ and for this crude assumptions are commonly made, such as a "square-root-of-time" scaling rule. Although square-root scaling is very common, it is not justified because it requires the portfolio returns to be independent and identically normally distributed. Hence the scaling of historical VaR to risk horizons longer than 1 day is a significant source of model risk, especially in options portfolios.[21]

3.1. ROM VaR methodology

Using ROM simulation to estimate portfolio value-at-risk can circumvent the disadvantages of existing VaR methodologies summarised above. Suppose that we have a portfolio with n assets. Similar to elliptical Monte Carlo VaR, ROM VaR starts by targeting a mean vector $\boldsymbol{\mu}_n^h$ and covariance matrix $\mathbf{S}_n^h$ for the h-day returns on risk factors under examination. We then simulate a sample $\mathbf{X}_{mn}^h$, using a ROM simulation of the form (18). This matrix $\mathbf{X}_{mn}^h$ represents m observations for the h-day returns on the portfolio's n risk factors. Since ROM simulation is an exact covariance simulation technique, the sample mean and covariance matrix of the simulated risk factor returns $\mathbf{X}_{mn}^h$, will match the targets $\boldsymbol{\mu}_n^h$ and $\mathbf{S}_n^h$ exactly. Assuming a portfolio mapping, $\pi : \mathbb{R}^n \to \mathbb{R}$, we construct a vector of h-day portfolio returns, $\mathbf{Y}_m^h = (y_1, \ldots, y_m)'$, defining

$$y_i = \pi(\mathbf{x}_i) \quad \text{for} \quad 1 \leq i \leq m$$

where $\mathbf{x}_i$ denotes the ith row of $\mathbf{X}_{mn}^h$. So given the α-quantile estimator $q_\alpha : \mathbb{R}^m \to \mathbb{R}$, we calculate ROM VaR as

$$\text{VaR}(\alpha, h) = -q_\alpha(\mathbf{Y}_m^h). \tag{32}$$

Typically the covariance matrix $\mathbf{S}_n^h$ will have a large influence on ROM VaR.[22] But VaR, being a lower quantile of the portfolio returns distribution, is also much affected by the skewness and kurtosis of the risk factor returns, and this is one reason why the (non-parametric) historical VaR model is favoured by many practitioners. However, the limitation of using only one realisation of the risk factors – i.e. the one that was observed historically – is that it assumes that history will repeat itself, in the sense

[20] Some practitioners bootstrap the sample, i.e. they re-sample randomly many times from the same set of returns, but this only serves to make the tails of the distribution blunt, as no new information is added. See [1, Section IV.3.2].

[21] See [1, Section IV.6.2].

[22] This is similar to normal VaR, and to Student-t VaR where $\mathbf{S}_n^h$ affects the VaR through the portfolio variance σ_h.

that the risk factor returns distribution over the risk horizon will be identical to their distribution in the historical sample.

With ROM VaR we can simulate a very large number of realisations for the risk factor returns, each being consistent with the sample moments that were observed in the historical sample. Alternatively, when stress testing a portfolio for extreme realisations of its risk factors, we can target other sample moments that are consistent with periods of financial crisis. As we saw in Sections 1 and 2, different L matrices and different orthogonal matrices will affect the characteristics of a ROM simulation in different ways. In particular, the effect of skewness and kurtosis can be incorporated into ROM VaR through a judicious choice of these matrices.[23]

ROM simulation has numerous applications to VaR estimation. For example, in stress-testing portfolios a risk manager would hypothesise values for sample moments that are believed to reflect stressful market conditions, apply ROM simulations to generate samples that have exactly these moments, and compute the stress-VaR as a lower quantile of the simulated distribution. In general, ROM VaR can be thought of as an extension of historical VaR, or a form of scenario VaR, where there is no need for the crude time-scaling approximations that are normally used because definition (32) does not explicitly depend on h; the risk horizon is implicit within each of the ROM simulations (18).

Suppose a portfolio manager holds a large portfolio of n assets and would like to gauge his risk exposure by calculating VaR. The only information available is an historical sample on the portfolio risk factor returns, which we denote $\mathbf{H}_{mn}$. Empirically he discovers that the historical data does not strongly resemble any elliptical multivariate distribution. In particular, the multivariate historical kurtosis is much larger than the kurtosis of a multivariate normal distribution. The length m of the historical data is typically too small to be confident in the accuracy of historical quantile estimates, especially for small α. Anyway, the manager does not believe that history will repeat itself exactly, only that the risk factor's sample moments are related to those observed in the historical sample. He is not performing a stress test, so he will use ROM simulations that target exactly the historical sample means and covariance matrix of the risk factor returns. However, given the recent crisis in the banking industry, investors are now seeking quite conservative estimates of potential losses, and want to be careful to incorporate large outliers into the VaR simulations. For this reason he would like the multivariate kurtosis of the simulated weekly returns to be $100\beta\%$ *greater* than the historical kurtosis level.

Given the scenario painted above, we choose to adopt a technique for ROM VaR where the L matrix of our simulations is chosen to reflect the manager's views on kurtosis, while preserving other properties of the historical data. Starting with the historical sample $\mathbf{H}_{mn}$ we calculate the historical mean vector $\boldsymbol{\mu}_n$, the historical covariance matrix $\mathbf{S}_n$, and the historical multivariate kurtosis $\kappa = \kappa_M(\mathbf{H}_{mn})$ of the weekly returns. As in Section 1.3, we orthogonalise our historical sample to produce a data-specific L matrix $\mathbf{L}^D_{mn} = GS(\mathbf{H}_{mn})$. We also construct a deterministic L matrix, $\mathbf{L}_{pn}$, of the form (9), where the integer parameter p will be determined by our kurtosis target.

First, targeting the historical mean vector and covariance matrix, the ROM simulated risk factor returns take the form

$$\mathbf{X}_{m+p,n} = \begin{pmatrix} \mathbf{1}_m \boldsymbol{\mu}_n' \\ \\ \mathbf{1}_p \boldsymbol{\mu}_n' \end{pmatrix} + \begin{pmatrix} \sqrt{m}\, \mathbf{L}^D_{mn} \\ \\ \sqrt{p}\, \mathbf{L}_{pn}\mathbf{R}_n \end{pmatrix} \mathbf{A}_n, \tag{33}$$

where $\mathbf{R}_n$ is a product of random orthogonal matrices and $\mathbf{A}_n'\mathbf{A}_n = \mathbf{S}_n$. Within $\mathbf{R}_n$ we include $n-1$ random upper Hessenberg matrices and a random sign matrix, which induces negative marginal skewness, to convey the specific concern for large negative returns. By applying proposition 2.2 to the concatenated sample $\mathbf{X}_{m+p,n}$, and recalling the invariance property of Mardia's measures, we deduce that

$$(m+p)\kappa_M(\mathbf{X}_{m+p,n}) \;=\; m\kappa_M(\mathbf{L}^D_{mn}) \;+\; p\kappa_M(\mathbf{L}_{pn}).$$

The kurtosis of $\mathbf{L}^D_{mn}$, by definition, equals the historical kurtosis κ. But we would like the kurtosis of our simulated sample to be $100\beta\%$ more than the historical value. That is, $\kappa_M(\mathbf{X}_{m+p,n}) = (1+\beta)\kappa$.

[23] This contrasts with Monte Carlo methods, where parametric assumptions place strong restrictions on the higher moments.

This will be achieved through an appropriate choice of p. Using approximation (22) for $\kappa_M(\mathbf{L}_{mn})$, we find that p must satisfy:

$$(m+p)(1+\beta)\kappa = m\kappa + pn(p-2). \tag{34}$$

Provided that β is positive, this quadratic equation always yields a positive solution p^+. Taking the closest integer part of this solution, we set $p = \text{int}(p^+)$, and use $\mathbf{L}_{pn}$ for our ROM simulations (33).

Now the portfolio risk manager knows that the ROM VaR simulations will resemble historical data through $\mathbf{L}^D_{mn}$, while representing his views on kurtosis via $\mathbf{L}_{pn}$. Targeting kurtosis in this way would not be possible with parametric VaR. And the presence of random orthogonal matrices in (33) ensures that we are not simply relying on our single historical sample, as we are forced to do with historical VaR.

3.2. Empirical results

In this section we compare four VaR models. The first is the ROM VaR method described above, which utilises deterministic and historical L matrices. The second and third are parametric Monte Carlo VaR, based on multivariate normal and Student-t distributions for the risk factors. Since ROM VaR is based on an exact covariance simulation method, we will adjust our Monte Carlo VaR simulations to achieve this same precision. This is equivalent to using parametric L matrices in (18), without any random orthogonal matrices. Our final method is standard historical VaR, which will form the benchmark of our investigation.

In order to test the accuracy of the different VaR methods we assume that the portfolio risk manager holds a portfolio of 10 MSCI World country indices.[24] For this portfolio we will calculate 1-day and 5-day (or 1-week) VaRs at 0.1%, 1% and 5% levels.[25] Our historical sample spans the period January 1995 to July 2009 (3789 days, 757 weeks). Starting in January 2000 we will use the first 5 years (1300 days, 260 weeks) of the historical sample to calculate historical means, covariances and higher moments of our weekly risk factor returns. We assume that we have an equally weighted portfolio, which corresponds to the portfolio mapping:

$$\pi : (x_1, \ldots, x_n) \longmapsto (x_1 + \cdots + x_n)/n.$$

Assuming that $\beta = 0.1$ the kurtosis of future returns will be 10% more than that of the historical returns, so we solve (34) for p and use ROM simulations of the form (33) to generate 10,000 returns on each of our 10 risk factors.[26] The mean and covariance matrix of our simulated sample will match the corresponding historical moments exactly. We then use our equally weighted portfolio mapping π to form 10,000 observations for the total portfolio return and then apply the quantile function, as in (32), to obtain the ROM VaR.

To test the accuracy of the VaR models we employ unconditional coverage tests, introduced by Kupiec [16], and the independence and conditional coverage tests, introduced by Christoffersen [4]. To perform these tests we first determine the number of "exceedances", i.e. the number of days (or weeks) when the portfolio lost more than the VaR that was estimated for that period. Under the null hypothesis of the unconditional coverage test, the proportion of exceedances should equal α. Over the period January 2000 to July 2009 there are 2489 daily observations in total period, the ROM VaR model yields 4 exceedances of the 0.1% VaR, 36 exceedances of the 1% VaR and 162 exceedances of the 5% VaR; that is the proportion of exceedance was 0.2%, 1.5% and 6.5% respectively. Out of 497 weekly observations the number of exceedances at the 0.1%, 1% and 5% levels were 1, 11 and 31 respectively, which correspond to exceedance proportions of 0.2%, 2.2% and 6.2%. This indicates that ROM simulation performs well, in general, for VaR estimation. However, one of the assumptions underlying the unconditional coverage test is that exceedances are independent events. If they are not, VaR estimates could be exceeded many

[24] We include indices for the United States, United Kingdom, Japan, Germany, France, India, China, Switzerland, South Africa and Australia in the portfolio.

[25] Very often weekly rather than daily historical data are used by portfolio managers for allocation decisions. This is because rebalancing is rarely done on a daily basis and, without the external regulatory control placed upon investment banks, risk assessment may be performed on a weekly rather than daily basis.

[26] A minimum of 10,000 simulations are standard in the VaR models used by the financial industry.

Table 1
Daily coverage test statistics. The 1% critical values are 6.63 for the unconditional and independence tests and 9.21 for the conditional test. Values in the table shown in bold therefore reject the appropriate null hypotheses at 1% significance.

Coverage test	Unconditional			Independence			Conditional		
Quantile	0.1%	1.0%	5.0%	0.1%	1.0%	5.0%	0.1%	1.0%	5.0%
ROM	0.77	4.40	**10.94**	0.01	**14.85**	**26.62**	0.79	**19.25**	**37.55**
Historical	1.96	6.01	**10.38**	0.02	**13.80**	**27.18**	1.98	**19.80**	**37.56**
Student-*t*	**65.94**	**64.03**	**13.28**	**10.51**	**16.47**	**37.83**	**76.45**	**80.49**	**51.11**
Normal	**131.19**	**77.83**	**10.38**	**14.31**	**17.06**	**32.56**	**145.50**	**94.89**	**42.94**

Table 2
Weekly coverage test statistics. The 1% critical values are 6.63 for the unconditional and independence tests and 9.21 for the conditional test. Values in the table shown in bold therefore reject the appropriate null hypotheses at 1% significance.

Coverage test	Unconditional			Independence			Conditional		
Quantile	0.1%	1.0%	5.0%	0.1%	1.0%	5.0%	0.1%	1.0%	5.0%
ROM	0.39	5.49	1.49	0.00	5.52	4.06	0.40	**11.02**	5.55
Historical	5.79	5.49	1.49	0.04	5.52	4.06	5.83	**11.02**	5.55
Student-*t*	2.57	**18.05**	6.35	0.02	2.36	5.16	2.58	**20.41**	**11.51**
Normal	**14.12**	**23.30**	6.35	0.10	1.69	5.16	**14.22**	**24.99**	**11.51**

days or weeks in succession, in which case they would not be capturing a change in market conditions adequately. In other words, the VaR model would not be sufficiently risk-sensitive.

Tables 1 and 2 report the results of unconditional and conditional coverage tests, and of the independent exceedance tests, for each of the four VaR models considered in this exercise. A successful VaR model must be sensitive to changes in kurtosis, but since normal VaR can only adapt to changes in the mean and standard deviation of returns, this model fails to cope with periods of exceptional returns. Student-*t* VaR performs better than normal VaR since, with 6 degrees of freedom, the marginal distributions have positive excess kurtosis. The best performing models are ROM and historical VaR. In some cases these two VaR methodologies have identical coverage test statistics. However when these test statistics differ, the ROM VaR method performs better than historical VaR. This is especially visible for 0.1% weekly historical VaR, where quantile estimation is inaccurate given the small in-sample period. Overall, the four VaR methods appear to be better at forecasting weekly, rather than daily VaR. In particular few VaR estimates pass the daily independence test, even though their unconditional performance is generally quite good.

It is worth noting that the length of the in-sample period (5-years) was not optimised for this study, and neither were the L matrices and the random orthogonal matrices used in the ROM simulations. The close relationship between our results for ROM VaR and historical VaR arises through the use of data-specific L matrices. If less emphasis were placed on data-specific L matrices the difference between ROM and historical VaR would be more significant.

4. Summary

This paper began by addressing the problem of finding a sample whose covariance matrix matches a target covariance matrix exactly. In order to form such a sample we introduced a new class or rectangular orthogonal matrix, the L matrices, which are fundamental for exact covariance simulation. Techniques from linear algebra were used to construct three different types of deterministic L matrices, while existing Monte Carlo methods were orthogonalised to produce parametric L matrices. L matrices engineered from existing data were also introduced.

Each L matrix, can be combined with random orthogonal matrices to generated infinitely many exact covariance samples. We call this approach ROM simulation. For certain deterministic and parametric L matrices we derive exact formulae for the multivariate skewness and kurtosis of ROM simulations. We then demonstrate how to target these moments exactly by choosing the parameters of the ROM

simulation appropriately. That is, in addition to exact covariance simulation, we have no simulation error for these higher moments. There is an extremely small rounding error which arises when many large matrices are multiplied together, but this is minuscule. However, since Monte Carlo methods are parametric, simulation errors may still have a significant influence on the higher moments for ROM simulations based on parametric L matrices.

We also explored the effect that some elementary classes of orthogonal matrices have on the dynamic sample characteristics, and on the features of marginal distributions. Note that the multivariate sample moments are left invariant by these ROMs, so we can generate many different types of sample paths that have the exactly the same multivariate moments. Features such as volatility clustering, or mimicking certain features of marginal distributions, may be important in some applications.

Finally we carried out an empirical study that compared four different methods for computing the value-at-risk (VaR) of a stock portfolio; this demonstrated some clear advantages of ROM simulation as a VaR resolution methodology. Clearly, the application of ROM simulation to VaR estimation – and by extension to portfolio optimisation – will be a productive area for future research. The ability of ROM simulation to produce infinitely many random samples which capture historical and/or hypothetical characteristics may also have great significance in other areas of finance whenever simulation is required. And there is much scope for further investigation into the applications of ROM simulation to other disciplines.

A. Appendix

A.1. Proof of Proposition 2.1

(1) *Kurtosis*

We start by expressing the Ledermann matrix in row vector notation, writing $\mathbf{L}_{mn} = (\boldsymbol{\ell}_1', \ldots, \boldsymbol{\ell}_m')'$. Our objective is to evaluate $\kappa_M(\mathbf{L}_{mn})$, for all $n < m$. Now since $\mathbf{L}_{mn}$ is an L matrix we know that it has sample mean vector $\boldsymbol{\mu}_n = \mathbf{0}_n$ and sample covariance matrix $\mathbf{S}_n = m^{-1}\mathbf{I}_n$. Simply observing that $\mathbf{S}_n^{-1} = m\mathbf{I}_n$, we deduce the following:

$$\kappa_M(\mathbf{L}_{mn}) = \frac{1}{m}\sum_{i=1}^{m}\left\{\boldsymbol{\ell}_i m\mathbf{I}_n\boldsymbol{\ell}_i'\right\}^2$$

$$= m\sum_{i=1}^{m}\left\{\boldsymbol{\ell}_i\boldsymbol{\ell}_i'\right\}^2.$$

Recall that $\mathbf{L}_{mn}$ is obtained by deleting the first $r = m - 1 - n$ columns of the L matrix $\mathbf{L}_{m,m-1}$, displayed in (8). We first consider the rows $\boldsymbol{\ell}_i$ where $r + 2 \le i \le m$. The first $i - r - 2$ entries of row i are all zero. Therefore

$$\boldsymbol{\ell}_i\boldsymbol{\ell}_i' = \left(\frac{-(i-1)}{\sqrt{(i-1)i}}\right)^2 + \sum_{k=i}^{m-1}\left(\frac{1}{\sqrt{k(k+1)}}\right)^2$$

$$= \frac{i-1}{i} + \frac{m-i}{mi}$$

$$= \frac{m-1}{m} \quad \text{for} \quad r+2 \le i \le m.$$

For the remaining rows $\boldsymbol{\ell}_i$, with $1 \le i \le r + 1$ we have that

$$\boldsymbol{\ell}_i\boldsymbol{\ell}_i' = \sum_{k=r+1}^{m-1}\left(\frac{1}{\sqrt{k(k+1)}}\right)^2 = \frac{m-(r+1)}{m(r+1)}. \tag{A.1}$$

Replacing r with n we calculate that

$$\kappa_M(\mathbf{L}_{mn}) = m\left[\frac{(m-n)n^2}{m^2(m-n)^2} + \frac{n(m-1)^2}{m^2}\right]$$

$$= n\Big[(m-2) + (m-n)^{-1}\Big].$$

(2) *Skewness*

Once again, knowing the sample mean and covariance matrix of $\mathbf{L}_{mn}$ exactly we can write:

$$\tau_M(\mathbf{L}_{mn}) = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\left\{\boldsymbol{\ell}_i m\mathbf{I}_n\boldsymbol{\ell}_j'\right\}^3$$

$$= m\sum_{i=1}^{m}\left\{\boldsymbol{\ell}_i\boldsymbol{\ell}_i'\right\}^3 + 2m\sum_{i<j}^{m}\left\{\boldsymbol{\ell}_i\boldsymbol{\ell}_j'\right\}^3.$$

As with the evaluation of kurtosis, we consider two cases separately. Firstly, for $1 \le i, j \le r+1$, we observe that $\boldsymbol{\ell}_i = \boldsymbol{\ell}_j$ and hence, using (A.1), we deduce that

$$\boldsymbol{\ell}_i\boldsymbol{\ell}_j' = \frac{m-(r+1)}{m(r+1)} \quad \text{for} \quad 1 \le i, j \le r+1.$$

Our second case concerns rows $\boldsymbol{\ell}_i$ and $\boldsymbol{\ell}_j$ such that $r+2 \le j \le m$ and $i < j$. In this case we see that

$$\boldsymbol{\ell}_i\boldsymbol{\ell}_j' = \frac{-(j-1)}{(j-1)j} + \sum_{k=j}^{m-1}\left(\frac{1}{\sqrt{k(k+1)}}\right)^2$$

$$= -\frac{1}{j} + \frac{m-j}{mj}$$

$$= -\frac{1}{m}.$$

Combining all these terms, and after some simplification, we calculate that

$$\tau_M(\mathbf{L}_{mn}) = m\left[\frac{(m-n)n^3}{m^3(m-n)^3} + \frac{n(m-1)^3}{m^3} + 2\left\{\frac{(m-n)(m-n-1)n^3}{2m^3(m-n)^3} + \frac{n(2m-n-1)}{2m^2}\right\}\right]$$

$$= n\Big[(m-3) + (m-n)^{-1}\Big].$$

A.2. Invariance of co-skewness

Let us begin with two different samples $\mathbf{X}_{m_X,n} = (\mathbf{x}_1', \ldots, \mathbf{x}_m')'$ and $\mathbf{Y}_{m_Y,n} = (\mathbf{y}_1', \ldots, \mathbf{y}_m')'$, on the same n random variables. Let $\bar{\mathbf{x}}$, $\bar{\mathbf{y}}$ and $\mathbf{S}_X$, $\mathbf{S}_Y$ denote the sample means and covariance matrices of $\mathbf{X}_{m_X,n}$,$\mathbf{Y}_{m_X,n}$ respectively. Given any invertible matrix $\mathbf{B}_n$ and column vector $\mathbf{b}_n$ we define

$$\tilde{\mathbf{X}}_{m_X,n} = \mathbf{X}_{m_X,n}\mathbf{B}_n + \mathbf{1}_{m_X}\mathbf{b}_n' \quad \text{and} \quad \tilde{\mathbf{Y}}_{m_Y,n} = \mathbf{Y}_{m_Y,n}\mathbf{B}_n + \mathbf{1}_{m_Y}\mathbf{b}_n'. \tag{A.2}$$

Now the means of $\tilde{\mathbf{X}}_{m_X,n}$ and $\tilde{\mathbf{Y}}_{m_Y,n}$ are $\bar{\mathbf{x}}\mathbf{B}_n + \mathbf{b}_n'$ and $\bar{\mathbf{y}}\mathbf{B}_n + \mathbf{b}_n'$, while the covariance matrices are $\mathbf{B}_n'\mathbf{S}_X\mathbf{B}_n$ and $\mathbf{B}_n'\mathbf{S}_Y\mathbf{B}_n$. So, noticing that

$$\left\{2(\mathbf{x}_i\mathbf{B}_n + \mathbf{b}_n' - \bar{\mathbf{x}}\mathbf{B}_n - \mathbf{b}_n')\mathbf{B}_n^{-1}(\mathbf{S}_X + \mathbf{S}_Y)^{-1}(\mathbf{B}_n^{-1})'(\mathbf{y}_j\mathbf{B}_n + \mathbf{b}_n' - \bar{\mathbf{y}}\mathbf{B}_n - \mathbf{b}_n')'\right\}^3$$

$$= \left\{2(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{S}_X + \mathbf{S}_Y)^{-1}(\mathbf{y}_j - \bar{\mathbf{y}})'\right\}^3,$$

we deduce that $\tau_C(\tilde{\mathbf{X}}_{m_X,n}, \tilde{\mathbf{Y}}_{m_Y,n}) = \tau_C(\mathbf{X}_{m_X,n}, \mathbf{Y}_{m_Y,n})$. Hence τ_C is invariant under non-singular affine transformations of the form (A.2).

A.3. Proof of Proposition 2.2

Our proposition begins with r random samples $\mathbf{X}_{m_1,n}, \ldots, \mathbf{X}_{m_r,n}$, each with sample mean $\boldsymbol{\mu}_n$ and sample covariance matrix $\mathbf{S}_n$. We adopt the following row vector notation

$$\mathbf{X}_{m_k,n} = \begin{pmatrix} \mathbf{x}_1^k \\ \vdots \\ \mathbf{x}_{m_k}^k \end{pmatrix} \quad \text{for} \quad 1 \le k \le r, \tag{A.3}$$

and setting $m = \sum_{k=1}^r m_k$, we write our concatenated sample as

$$\mathbf{X}_{m,n} = (\mathbf{X}'_{m_1,n}, \ldots, \mathbf{X}'_{m_r,n})'.$$

Throughout this proof we will use the dot product notation

$$\mathbf{x} \cdot \mathbf{y} = \begin{cases} \mathbf{x}\mathbf{y}' & \text{for row vectors,} \\ \mathbf{x}'\mathbf{y} & \text{for column vectors.} \end{cases}$$

(1) *Concatenated mean vector*

The mean of our concatenated sample $\mathbf{X}_{mn}$ is

$$m^{-1}\mathbf{1}'_m\mathbf{X}_{mn} = m^{-1}\sum_{k=1}^r \mathbf{1}'_{m_k}\mathbf{X}_{m_k,n} = m^{-1}\sum_{k=1}^r m_k\boldsymbol{\mu}_n = \boldsymbol{\mu}_n.$$

(2) *Concatenated covariance matrix*

Having established the mean of our concatenated sample, we can calculate the covariance matrix of $\mathbf{X}_{mn}$ as

$$m^{-1}\mathbf{X}'_{mn}\mathbf{X}_{mn} = m^{-1}\sum_{k=1}^r \mathbf{X}'_{m_k,n}\mathbf{X}_{m_k,n} = m^{-1}\sum_{k=1}^r m_k\mathbf{S}_n = \mathbf{S}_n.$$

(3) *Concatenated skewness*

We now calculate the Mardia [23] skewness of the concatenated sample $\mathbf{X}_{mn}$ in terms of the skewness of the sample components $\mathbf{X}_{m_1,n}, \ldots, \mathbf{X}_{m_r,n}$. Since we know that Mardia's measure are invariant under non-singular affine transformations, we begin by defining

$$\mathbf{Z}_{mn} = (\mathbf{X}_{mn} - \mathbf{1}_m\boldsymbol{\mu}'_n)\mathbf{A}_n^{-1},$$

where $\mathbf{A}'_n\mathbf{A}_n = \mathbf{S}_n$. Then we can write $\mathbf{Z}_{mn} = (\mathbf{Z}'_{m_1,n}, \ldots, \mathbf{Z}'_{m_r,n})'$, where each component

$$\mathbf{Z}_{m_k,n} = (\mathbf{X}_{m_k,n} - \mathbf{1}_{m_k}\boldsymbol{\mu}'_n)\mathbf{A}_n^{-1}.$$

As a result of these non-singular affine transformations $\mathbf{Z}_{mn}$ will have sample mean vector $\mathbf{0}_n$ and covariance matrix $\mathbf{I}_n$. This will also be the case for each component $\mathbf{Z}_{m_k,n}$, for $1 \le k \le r$. Adopting a row notation equivalent to (A.3), we can express

$$\begin{aligned}
\tau_M(\mathbf{Z}_{mn}) &= m^{-2}\sum_{i=1}^m\sum_{j=1}^m (\mathbf{z}_i \cdot \mathbf{z}_j)^3 \\
&= m^{-2}\left[\sum_{k=1}^r \left\{\sum_{i=1}^{m_k}\sum_{j=1}^{m_k}(\mathbf{z}_i^k \cdot \mathbf{z}_j^k)^3\right\} + 2\sum_{k<1}^r \left\{\sum_{i=1}^{m_k}\sum_{j=1}^{m_k}(\mathbf{z}_i^k \cdot \mathbf{z}_j^l)^3\right\}\right] \\
&= m^{-2}\sum_{k=1}^r m_k^2\tau_M(\mathbf{Z}_{m_k,n}) + 2m^{-2}\sum_{k<1}^r (m_k + m_l)^2\tau_C(\mathbf{Z}_{m_k,n}, \mathbf{Z}_{m_l,n}).
\end{aligned}$$

The third statement of the proposition follows immediately from the non-singular affine invariance property of τ_M and τ_C.

(4) *Concatenated kurtosis*

For Mardia's kurtosis we calculate that

$$\begin{aligned}\kappa_M(\mathbf{Z}_{mn}) &= m^{-1}\sum_{i=1}^{m}(\mathbf{z}_i\cdot\mathbf{z}_i)^2\\ &= m^{-1}\sum_{k=1}^{r}\left\{\sum_{i=1}^{m_k}(\mathbf{z}_i^k\cdot\mathbf{z}_i^k)^2\right\}\\ &= m^{-1}\sum_{k=1}^{r}m_k\kappa_M(\mathbf{Z}_{m_k,n}).\end{aligned}$$

Applying the non-singular affine invariance property of κ_M yields the final expression of the proposition.

References

[1] C. Alexander, Market Risk Analysis Volume IV, Value-at-Risk Models, Wiley, 2008.
[2] M. Berry, J. Dongarra, Y. Kim, A highly parallel algorithm for the reduction of a non-symmetric matrix to block upper-Hessenberg form, Parallel Comput. 21 (1995) 1189.
[3] A. Cayley, Sur quelques propriétés des déterminants gauches, J. Reine Angew. Math. (Crelle's J.) 32 (1846) 119–123.
[4] P. Christoffersen, Evaluating interval forecasts, Internat. Econom. Rev. 39 (1998) 841–862.
[5] R. Courant, D. Hilbert, Wiley, 1999.
[6] H. Cramer, Mathematical Methods of Statistics, Princeton University Press, 1946.
[7] P. Date, R. Mamon, L. Jalen, A new moment matching algorithm for sampling from partially specified symmetric distributions, Oper. Res. Lett. 36 (2008) 669–672.
[8] L. Gemignani, A unitary Hessenberg QR-based algorithm via semiseparable matrices, J. Comput. Appl. Math. 184 (2005) 505–517.
[9] W. Givens, Computation of plane unitary rotations transforming a general matrix to triangular form, SIAM J. Appl. Math. 6 (1958) 26–50.
[10] P. Glasserman, Monte Carlo Methods in Financial Engineering, Springer, 2004.
[11] G. Golub, C. Loan, Matrix Computations, third ed., Johns Hopkins University Press, Baltimore, 1996.
[12] S. Gutjahr, N. Henze, M. Folkers, Shortcomings of generalized affine invariant skewness measures, J. Multivariate Anal. 71 (1999) 1–23.
[13] N. Higham, Accuracy and Stability of Numerical Algorithms, Society for Industrial and Applied Mathematics, 1996.
[14] A. Householder, Unitary triangularization of a non-symmetric matrix, J. ACM 5 (1958) 339–342.
[15] T. Kollo, Multivariate skewness and kurtosis measures with an application in ICA, J. Multivariate Anal. 99 (2008) 2328–2338.
[16] P. Kupiec, Techniques for verifying the accuracy of risk measurement models, J. Derivatives 2 (1995) 173–184.
[17] H. Lancaster, The Helmert matrices, Amer. Math. Monthly 72 (1) (1965) 4–12.
[18] K.-H. Li, Generation of random matrices with orthonormal columns and multivariate normal variates with given sample mean and covariance, J. Statist. Comput. Simulation 43 (1992) 11–18.
[19] J. Lyhagen, A method to generate multivariate data with the desired moments, Comm. Statist. Simulation Comput. 37 (2008) 2063–2075.
[20] B. Mandelbrot, A. Fisher, L. Calvet, A multifractal model of asset returns, Cowles Foundation Discussion Paper, 1997.
[21] K. Mardia, I. Dryden, The statistical analysis of shape data, Biometrika 76 (2) (1989) 271–281.
[22] K. Mardia, A.N. Walder, Shape analysis of paired landmark data, Biometrika 81 (1) (1994) 185–196.
[23] K.V. Mardia, Measures of multivariate skewness and kurtosis with applications, Biometrika 57 (3) (1970) 519.
[24] A. Meucci, Simulations with exact means and covariances, Risk Mag. 22 (2009) 89–91.
[25] C. Perignon, D. Smith, The level and quality of value-at-risk disclosure by commercial banks, J. Banking Finance 34 (2010) 362–377.
[26] V. Prasolov, Problems and Theorems in Linear Algebra, vol. 134American Mathematical Society, Providence, RI, 1994.